الجمهورية الجزائرية الديموقراطية الشعبية
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE SAAD DAHLEB DE BLIDA

FACULTE DES SCIENCES EXACTES

DEPARTEMENT D'INFORMATIQUE

Mémoire de projet de fin d'études
En vu d'obtention du diplôme
d'ingénieur d'état en informatique

Option : Intelligence Artificielle.

Thème :

Conception et implémentation d'un protocole de routage basé sur la prédiction pour les réseaux mobiles Ad hoc

Proposé et encadré par :

M. DERHAB Abdelouahid.

Présenté par :

TOUMI Redhouane

Devant le jury composé de :
M. D. Bennouar (président).
M. M. Mehieddinne (membre).
M^lle^. N. Boustia (membre).

المكتبة

Promotion: Octobre 2004

MIG-004-25-1

Conception et implémentation d'un protocole de routage basé sur la prédiction pour les réseaux mobiles Ad hoc

5.12 La sauvegarde des routes par les nœuds intermédiaires. 58
5.13 Les routes trouvées sont triées selon le RET. 59
5.14 Le problème en cas d'insérer en premiers les routes plus courtes. 59
5.15 Table intermédiaire. 60
5.16 Les routes stable dans la table de routage. 60
5.17 La sauvegarde de route par les nœuds intermédiaires. 61
5.18 Format d'un paquet de notification "Faible" dans MPRP. 62
5.19 Schéma récapitulatif du processus de maintenance des routes dans MPRP. 63
5.20 Format d'un paquet de notification "Fort" dans MPRP. 63

6.1 Organisation en couche du GloMoSim. 70
6.2 Les trois(3) fonctions d'intégration d'un protocole dans le simulateur. 71
6.3 La fraction de réception de données en fonction de la mobilité. 74
6.4 Le nombre de paquets de contrôles en fonction de la mobilité. 76
6.5 Le délai moyen de transfert en fonction de la mobilité. 77
6.6 L'optimalité des routes en fonction de la mobilité. 78
6.7 La fraction de réception de données en fonction du nombre de session. 79
6.8 Le nombre de paquet de contrôle en fonction du nombre de session. 80
6.9 Le délai moyen de transfert en fonction du nombre de session. 81
6.10 L'optimalité des routes en fonction du nombre de session. 82
6.11 Le nombre de paquet de contrôle en fonction de la connectivité. 83
6.12 Le nombre de paquet de contrôle en fonction de la connectivité. 84
6.13 Le délai moyen de transfert en fonction de la connectivité. 85
6.14 L'optimalité des routes en fonction de la connectivité. 85

A.1 Outil de visualisation du GloMoSim. 96

Liste des tableaux

3.1 Comparaison entre quelques protocoles de routage des réseaux Ad hoc. 35

Introduction générale

Durant les dernières années, le besoin d'avoir plus de mobilité tout en ayant le pouvoir accéder à des données partagées ou à s'échanger de l'information à tout moment, en utilisant des dispositifs mobiles (téléphones portables, PDA, Laptops[1], etc.), a rendu la notion du réseau sans infrastructure (ou Ad hoc) très répandu.

Les réseaux Ad hoc sont des réseaux d'entités mobiles sans fil qui n'utilisent aucune infrastructure fixe préexistante ou administration centralisée. Ces réseaux reposent sur le principe d'auto-organisation puisque toute entité peut disparaître ou apparaître à tout moment laissant le soin au réseau de gérer ces changements de topologie.
Les caractéristiques principales des réseaux Ad hoc sont, l'absence d'infrastructure et le changement fréquent de la topologie qui est la conséquence de la mobilité des nœuds. Ces caractéristiques posent un certain nombre de problèmes non triviaux. En particulier, afin d'assurer la transmission des données d'un bout à l'autre du réseau, les nœuds mobiles doivent avoir la capacité de retransmettre les données, jouant ainsi le rôle des routeurs. Cependant, les spécificités du lien radio ainsi que la mobilité potentielle des nœuds rendent les protocoles de routage utilisés dans les réseaux conventionnels[2] peu performants. C'est pour cela une grande partie des travaux actuels dans les réseaux Ad hoc se sont concentrés sur les problématiques de routage dans les réseaux mobiles Ad hoc. La plus part des solutions existantes proposées pour effectuer le service de routage ne sont encore que des drafts (brouillant) et restent en cours de développement et/ou de spécification.

Les protocoles de routage actuels sont classés en deux principales familles : Famille des protocoles de routages non prédictifs et la famille des protocoles de routage prédictifs. La première famille se divise en trois modèles de fonctionnement : Protocoles de routage proactifs, protocoles de routage réactifs et protocoles de routages hybrides. La deuxième famille se divise également en deux modèles de fonctionnement, protocoles de routage proactifs et protocoles de routages réactifs.
Ce travail entre dans le cadre de l'étude du problème de routage dans les environnements

[1] PC portables
[2] Par exemple les réseaux Ethernet.

mobiles caractérisés par l'absence d'infrastructure.
Notre travail consiste à proposer un protocole de routage pour les réseaux mobiles Ad hoc, s'adapte rapidement aux mobilité des nœuds en utilisant pour ça une technique de prédiction afin d'estimer la durée de vie d'une route qui est la conséquence de mobilité de telle sorte, en réagissant avant l'occurrence d'une défaillance inattendue de ces derniers.

Ce mémoire est composé de six chapitres. Le premier chapitre consiste à décrire le concept de réseaux Ad hoc, ses caractéristiques, ses domaines d'utilisation, etc.
Le deuxième chapitre, entame le concept de routage dans les réseaux filaires et Ad hoc, introduit également quelques approches et notions générales qui vont aider à mieux comprendre la présentation des protocoles de routages existent actuellement.
Le troisième chapitre présente la famille des protocole de routage non prédictifs, tandis que le quatrième chapitre, présente la famille des protocole de routages prédictifs ainsi que les mécanisme de prédiction utilisés par ces derniers.
Le cinquième chapitre se divise en deux parties : Avant de présenter d'une manière détaillée notre protocole proposé dans la deuxième partie, une petite introduction à quelques approches et concepts généraux sont présentés dans la première partie de ce chapitre.
Le dernier chapitre consiste d'abord à présenter l'environnement de simulation dans lesquels les protocoles, y compris notre protocole, vont être simulés. Ce chapitre présente également et interprète les résultats obtenus de la simulation faite dans le simulateur GloMoSim.
Finissons ce mémoire par une conclusion générale récapitulant notre travail, ses objectifs et des éventuelles perspectives.

Table des matières

1 Réseaux mobiles Ad hoc - Généralité 1

1.1 Introduction . . . 1
1.2 Définition d'un réseau Ad hoc . . . 2
1.3 Modélisation d'un réseau Ad hoc . . . 2
1.4 Avantages des réseaux Ad hoc . . . 3
1.5 Caractéristiques des réseaux Ad hoc . . . 4
1.6 Domaine d'application des réseaux mobiles Ad hoc . . . 6
1.7 Conclusion . . . 8

2 Routage unicast dans les réseaux mobiles Ad hoc 9

2.1 Introduction . . . 9
2.2 Définition du routage . . . 10
2.2.1 Problème de routage . . . 10
2.2.2 Protocoles conventionnels . . . 11
2.3 Notions générales . . . 13
2.3.1 Inondation(Flooding) . . . 13
2.3.2 Vecteur de distance (Distance Vector) . . . 14
2.3.3 État de lien(link state) . . . 14
2.3.4 Routage de source (Source Routing) . . . 14
2.3.5 Graphe Acyclique Orienté (DAG : Direct Acyclic Graph) . . . 14
2.3.6 Groupe . . . 14
2.3.7 Hiérarchie . . . 15
2.3.8 Routage saut par saut (hop by hop) . . . 15
2.3.9 Système de positionnement Global (GPS :Global Positioning System) . . . 15
2.3.10 Inversement de lien . . . 15
2.4 Propriétés désirées d'un protocole de routage efficace . . . 16
2.5 Conclusion . . . 17

3 Protocoles de routage non prédictifs 18

3.1 Introduction . . . 18

3.2 Protocoles de routage proactifs 19
3.2.1 Destination Sequenced Distance Vector(DSDV) 19
3.2.2 Wireless Routing Protocol(WRP) 19
3.2.3 Global State Routing(GSR) 20
3.2.4 Fisheye State Routing (FSR) 21
3.2.5 Clusterhead Gateway Switch Routing Protocol (CGSR) 22
3.2.6 Hierarchical State Routing(HSR) 23
3.2.7 Zone-based Hierarchical Link State Routing(ZHLS) 24
3.2.8 Distance Routing Effect Algorithm for Mobility (DREAM) 25
3.3 Inconvénients des protocoles de routage proactifs 26
3.4 Protocoles de routage Réactifs 26
3.4.1 Ad hoc On-demand Distance Vector (AODV) 27
3.4.2 Dynamic Source Routing(DSR) 28
3.4.3 Temporary Ordred routing Algorithm(TORA) 28
3.4.4 Associativity Based Routing (ABR)et Signal Stability routing Adaptive(SSA) . . 29
3.4.5 Location-Aided Routing (LAR) 31
3.5 Inconvénient des protocoles de routage réactifs 32
3.6 Les protocoles de routage hybrides 33
3.6.1 Zone Routing Protocol(ZRP) 33
3.7 Evaluation qualitative des protocoles 34
3.8 Conclusion 35

4 Protocoles de routage prédictifs **37**
4.1 Introduction 37
4.2 Les mécanismes de prédiction 38
4.3 Méthode basée sur les informations de GPS 38
4.3.1 Mécanisme de base 39
4.4 Application 40
4.4.1 Flow Oriented Routing Protocol 40
4.4.2 Distance Vector with Mobility Prediction(DV-MP) 41
4.4.3 Inconvénient de ce protocole : 43
4.5 Méthode d'affinité 44
4.5.1 Route-lifetime Assessment Based Routing protocol (RABR) 44
4.6 Conclusion 45

5 Conception et ipmlémentation du protocole MPRP **46**
5.1 Introduction 46

I Notions générales 47

5.2 Les Routes disjointes 48
5.2.1 Définition 48
5.2.2 Calcul de Link Expiration Time et Route Expiration Time 50
5.2.3 Stabilité d'une route 51
5.2.4 Calcul du Seuil 52

II Présentation de MPRP 54

5.3 Multi-path Prediction-based Routing protocol (MPRP) 55
5.3.1 La découverte des routes 55
5.3.2 La maintenance des routes 61
5.4 Conclusion 65

6 Environnement et résultats de simulation 67

6.1 Introduction 67
6.2 Définitions de la simulation 68
6.3 Le langage de simulation PARSEC 68
6.3.1 Entité-type 68
6.3.2 Message 69
6.3.3 Événement 69
6.3.4 Le simulateur GloMoSim 69
6.4 Les métriques de performances 71
6.5 Paramètres du réseau 72
6.6 Paramètres de comparaison 73
6.7 Métriques de simulation 74
6.7.1 Mobilité : 74
6.7.2 Charge d'application(Nombre de session) : 79
6.7.3 Connectivité : 82
6.8 Conclusion 86

A Installation de GloMoSim 93

A.1 Installation sous Windows (XP , NT ,2000 pro) 93
A.2 Installation sous Linux (Redhat 7.3) 94
A.3 Outil de visualisation du GloMoSim 95

Table des figures

1.1 Classification générale des réseaux mobiles sans fil. 2
1.2 Modélisation d'un réseau mobile Ad hoc. 3
1.3 Connexion multi-saut entre la source et la destination. 3
1.4 Un exemple du changement de la topologie d'un réseau mobile Ad hoc. 5
1.5 Liaisons dans les réseaux Ad hoc. 6

2.1 Le chemin utilisé dans le routage entre la source et la destination. 11

3.1 Classification des protocoles de routage dans les réseaux Ad hoc. 18
3.2 La portée(champ de vision) de Fisheye. 21
3.3 Classification des protocoles de routage dans les réseaux Ad hoc. 22
3.4 Exemple de groupe, Formation de hiérarchie dans le protocole HSR. 24
3.5 La décomposition du réseau en zones. 25
3.6 Les deux approches de LAR. 32

4.1 Paramètres de mouvement. 39
4.2 Le processus d'installation de flux dans FORP. 41
4.3 Le processus de "handoff". 42
4.4 Exemple d'une mise à jour de la table de routage. 43

5.1 routes disjointes 48
5.2 Utilité des routes disjointes 49
5.3 Distribution de la charge dans le réseaux. 50
5.4 Les LETs associés aux liens du réseau 51
5.5 Les états possibles d'un lien dans le réseau. 52
5.6 Calcul du seuil par un nœud intermédiaire. 53
5.7 Format d'un paquet requête de route dans MPRP. 56
5.8 Table de routage. 56
5.9 Table des voisins. 56
5.10 Le processus de découverte des routes dans le protocole MPRP. 57
5.11 Format d'un paquet réponse de route dans MPRP. 58

Chapitre 1

Réseaux mobiles Ad hoc - Généralité

1.1 Introduction

L'essor des technologies sans fil offre aujourd'hui des perspectives intéressantes dans le domaine des télécommunications. L'évolution récente des moyens de communication sans fil a permis la manipulation d'informations au travers d'unités de calcul portables aux caractéristiques bien particulières (faible capacité de stockage, source d'énergie autonomie, puissance limitée, etc.) qui accèdent au réseau par le biais d'une interface de communication sans fil.

Un réseau est dit sans fil lorsque les machines qui le composent ne sont pas reliées entre elles par des câbles, mais utilisent, pour communiquer, le médium radio ou infrarouge. Comme les signaux propagés sur ces médias s'atténuent au fur et à mesure qu'ils s'éloignent de leur émetteur, un nœud ne peut pas communiquer avec un autre s'il est situé trop loin de lui. On définit alors l'ensemble des voisins d'un nœud comme étant l'ensemble des nœuds capables de recevoir et de comprendre les signaux émis par celui-ci. Ces voisins sont appelés des voisins directs ou voisins à un saut. On définit de même un voisin à deux sauts comme étant un voisin à un saut d'un voisin direct, et ainsi de suite. Les *réseaux mobiles Ad hoc* permettent un modèle plus flexible de transmission que les réseaux traditionnels[1] parce que l'utilisateur n'est pas limité à un emplacement physique fixe. À la différence des réseaux sans fil cellulaires, un réseau sans fil Ad hoc n'a aucune infrastructure fixe de transmission. Pour une connexion active, le nœud de fin aussi bien

[1] Réseau Ethernet par exemple.

que les nœuds intermédiaires peuvent être mobiles.
Les réseaux sans fil les plus connus du grand public sont très probablement les réseaux GSM, qui permettent notamment le transport de la voix.
Les réseaux Ad hoc étudiés ici sont, eux aussi, des réseaux sans fil. Cependant, à l'inverse des réseaux GSM qui dépendent des infrastructures fixes, comprenant des stations de base ou points d'accès, les réseaux Ad hoc en sont dépourvus. De même, ils ne comportent ni nœud privilégié, ni infrastructure de contrôle centralisée.
Donc, les réseaux mobiles ou sans fil, peuvent êtres classés en deux classes principales : la classe des réseaux sans fil avec infrastructure, et la classe des réseaux sans fil avec infrastructure(voir figure 1.1).

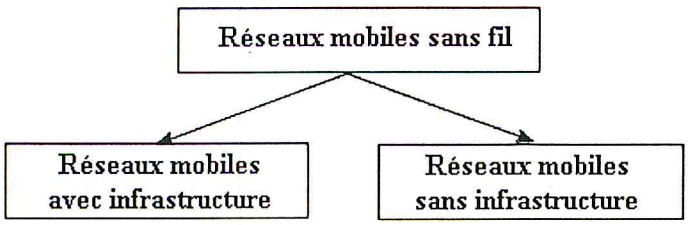


FIG. 1.1 – Classification générale des réseaux mobiles sans fil.

1.2 Définition d'un réseau Ad hoc

Un réseau mobile Ad hoc, appelé généralement MANET(Mobile Ad hoc NETwork), est un ensemble de nœuds[2] (hôtes) mobiles, interconnectés par une technologie sans fil formant un réseau temporaire sans l'aide de toute infrastructure préexistante ou administration centralisée. Chaque nœud agit comme un routeur et se déplace d'une façon arbitraire et imprédictible. Dans la plupart des cas, le nœud destinataire ne se trouve pas obligatoirement dans la portée de communication du nœud source, ce qui implique que l'échange des données entre deux nœuds doit être effectué par des nœuds intermédiaires.

1.3 Modélisation d'un réseau Ad hoc

Un réseau mobile Ad hoc peut être modélisé par un graphe non orienté $G = (V, E)$ (voir la figure 1.2), où : V représente un ensemble de nœuds du réseau, et E est un

[2]Les nœuds dans un réseau Ad hoc peuvent êtres des ordinateurs portables, PDA, handheld, ...etc.

ensemble de liens qui existent entres ces nœuds. Un lien (i, j) est formé quand la distance entre i et j devient inférieure ou égale à R (où R est la *portée de communication* des nœuds). La figure 1.3 montre un réseau Ad hoc constitué de (12) nœuds où les nœuds S

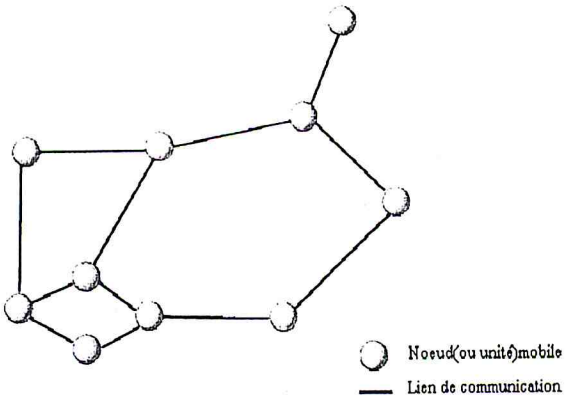


FIG. 1.2 – Modélisation d'un réseau mobile Ad hoc.

et D ne sont pas dans la portée de communication de chacun d'eux, mais S et D peuvent se communiquer entre eux en utilisant les nœuds A, B, C et E comme des routeurs ou des nœuds intermédiaires pour transmettre (faire passer) les paquets de S vers D et vis-versa.

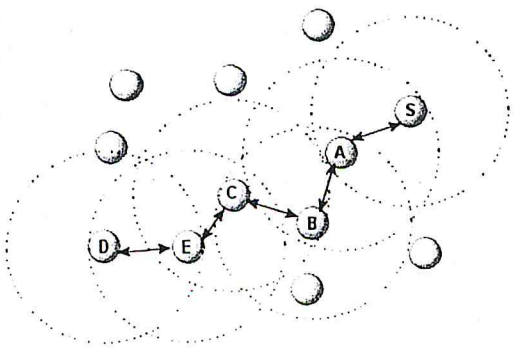


FIG. 1.3 – Connexion multi-saut entre la source et la destination.

1.4 Avantages des réseaux Ad hoc

Les réseaux Ad hoc permettent l'échange de tout type de données numériques en utilisant les ondes radio comme un moyen de communication. De ce fait, ces types de réseaux bénéficient de nombreux avantages par rapport à l'utilisation des réseaux

filaires, qui malgré les performances supérieures(débit, délai de transfert, etc.), ils restent plus onéreux et moins pratique à installer (voir impossible dans certain cas). Parmi ces avantages, nous citons :

- **L'absence d'infrastructure :** Il suffit de mettre en présence plusieurs machines pour que le réseau existe. Ceci rend la construction d'un réseau Ad hoc rapide et peu onéreuse.
- **Mobilité des nœuds :** L'absence de câblages autorise es nœuds à se déplacer librement.
- **le coût d'installation du réseau :** Le réseau Ad hoc peut réduire le coût d'installation d'un réseau puisque il ne requière aucune infrastructure fixe. En plus de ça, une liaison sans fil peut remplacer un câblage manquant.

1.5 Caractéristiques des réseaux Ad hoc

À partir de la définition des réseaux Ad hoc vue précédemment, il est intéressant de présenter les principales caractéristiques qui différencient un réseau Ad hoc d'un autre réseau, que ce soit un réseau filaire ou un réseau sans fil avec infrastructure.

Topologie dynamique : Les nœuds sont libres de se déplacer arbitrairement, ce qui fait que la topologie du réseau – typiquement multi-saut – peut changer aléatoirement et rapidement n'importe quand, et peut être constituée à la fois des liaisons unidirectionnelles et bidirectionnelles. A un moment donné, un mobile peut rejoindre ou quitter le réseau Ad hoc existant, la figure 1.4 ci-dessous illustre cette caractéristique.

Mobilité : La mobilité des nœuds constitue à l'évidence une caractéristique très spécifique des réseaux Ad hoc. Les nœuds dans un réseau Ad hoc sont mobiles. En conséquence, il est impossible de prévoir si un lien existant à une certaine date sera toujours valide à une date ultérieure.

Problème de sécurité : Les réseaux sans fil sont par nature plus sensibles aux problèmes de sécurité. Pour les réseaux Ad hoc, les possibilités de s'insérer dans le réseau sont plus grandes et la protection contre les *intrusions* ou les *dénis de service* sont plus délicates.

Bande passante limitée : Les liaisons sans fil auront toujours une capacité inférieure par rapport aux liaisons filaires. En plus, le débit réel des communications sans

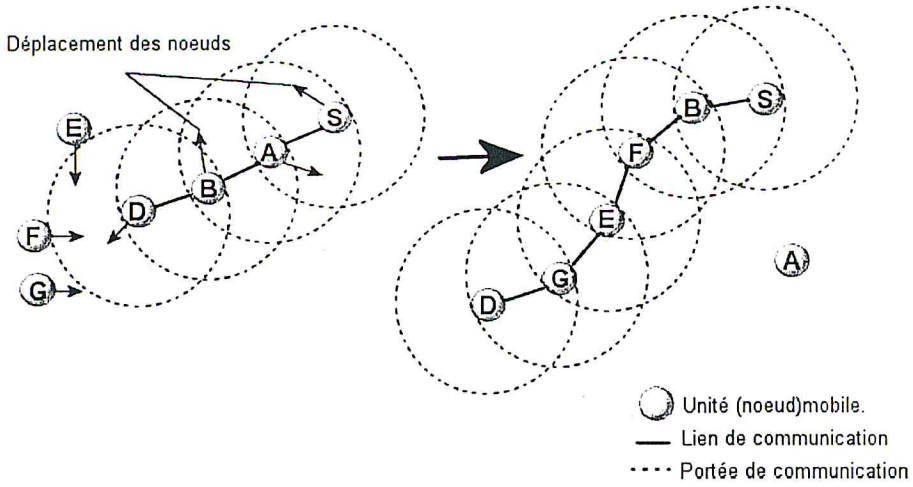


FIG. 1.4 – Un exemple du changement de la topologie d'un réseau mobile Ad hoc.

fil après avoir déduit les effets des accès multiples, du *fading*[3], du bruit, des interférences, etc, est souvent inférieur aux taux de transfert maximum du canal radio. Un des effets de ces débits de liaison relativement faibles est que la congestion est généralement la norme plus que l'exception, i.e. la demande sur les applications distribuées approchera ou dépassera souvent la capacité du réseau. Comme le réseau mobile est souvent une simple extension d'un réseau fixe, les utilisateurs mobiles Ad hoc demanderont les mêmes services. Cette demande ne cessera de croître avec l'augmentation des traitements multimédias et des applications basées sur les réseaux. Comparés aux réseaux filaires Ethernet conventionnels qui fonctionnent à 100 Mbit/sec, les réseaux sans fil n'offrent pas des débits élevés. La technologie 802.11b autorise des taux de transferts pouvant atteindre 11Mbit/sec au maximum.

L'énergie : Les applications relatives aux réseaux Ad hoc ont en général un caractère nomade et tirent leur autonomie de batteries. Emettre ou recevoir des données consomme de l'énergie et l'on peut chercher à l'économiser en optimisant les protocoles de gestion du réseau. La puissance d'émission a un impact important sur la quantité d'énergie utilisée et là encore on essaie si possible de la limiter à ce qui est strictement nécessaire.

Le routage est multi-saut : Cela signifie que des communications entre deux nœuds doivent pouvoir s'effectuer même si ceux-ci sont hors de la portée de communication directe. La connaissance réciproque de leur existence et les échanges d'informations

[3] L'atténuation du signal.

doivent être possibles en traversant d'autres nœuds du réseau.

Liaison unidirectionnelle : Cette liaison peut être intrinsèquement unidirectionnelle par exemple, si les nœuds ont des puissances inégales dans ce cas, l'un peut recevoir l'autre sans que la réciproque ne soit vraie; Ce caractère unidirectionnel d'une liaison, est une caractéristique qui n'apparaît pas pour une liaison filaire (voir la figure 1.5).

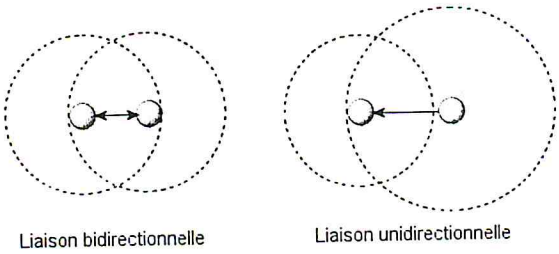


FIG. 1.5 – Liaisons dans les réseaux Ad hoc.

Contraintes physiques limitées : Les unités mobiles des réseaux Ad hoc sont à la fois des terminaux (PC, assistants personnels, téléphones, etc.) et des routeurs pour leurs voisins, puisque le rôle principal des retours est d'exécuter le protocole de routage. En plus, l'espace de stockage augmente en fonction de nombre de nœuds du réseau, ceci exige un espace de stockage très important.

Les caractéristiques précitées forment un ensemble de paramètres prédominants pour la conception des protocoles pour les topologies dynamiques.

1.6 Domaine d'application des réseaux mobiles Ad hoc

Malgré son origine militaire, l'idée a rapidement été jugée intéressante pour des applications civiles, pour les raisons suivantes : La simplicité de déploiement, Le mouvement des usagers, etc (voir les avantages des réseaux Ad hoc section 1.4). La technologie des réseaux Ad hoc peut être utilisée dans beaucoup de domaines[TOH 02], par exemple :

Applications militaires Les champs de bataille digitaux et sans fil moderne nécessitent une communication robuste et fiable dans différentes situations. La majorité des appareils sans fil sont installés dans des véhicules(par exemple : Chars, Camions, jeeps, etc.). En outre, les soldats peuvent porter des appareils de télécommunications permettant la communication sans fil avec une station de base ou même directement avec un autre appareil si les uns se trouvent dans la portée des autres. Cependant,

ces formes de communications sont actuellement considérées primitives. Parfois et quand la station de base sans fil est détruite par l'ennemi, un soldat se trouvera empêché de communiquer avec les autres soldats s'ils ne sont pas dans sa portée de communication. A ce stade, le routage sans fil Ad hoc intervient pour jouer son rôle.

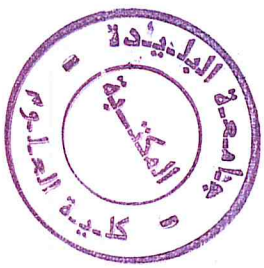
Les réseaux Ad hoc sont aussi auto-reconfigurable, quand les nœuds disparaissent suite à une destruction ou à un déplacement. A travers la communication multi-sauts, un soldat peut communiquer avec les autres soldats s'ils sont accessible via d'autres nœuds (soldats intermédiaires).

Les réseaux de capteurs Un autre scénario est la gestion des réseaux de capteur instantanés dans les champs de batailles, très souvent et avant d'entreprendre une attaque, une opération de surveillance minutieuse du site d'ennemie est essentielle. Cependant, les capteurs filaires ne sont pas pratiques de tout, du fait qu'ils peuvent être détecter aisément par l'ennemie. Ces capteurs sont capables de rassembler rapidement les informations concernant l'emplacement et l'environnement et retransmettre périodiquement les information au centre de commandement et de contrôle(CCC). Ces informations sont donc utilisées pour influencer la stratégie de l'attaque et déterminent la décision adéquate.

Opérations de secours Les opérations de secours se passent dans des régions désastreuses (incendie, inondation, tremblement de terre) où une infrastructure de réseau filaire ou station de base n'existe pas, mais une communication entre les équipes de secours et nécessaire [ROS 00], dans ce cas une installation d'un réseau Ad hoc peut résoudre ce problème.

Conférences Le réseau Ad hoc est faisable pour un groupe de gens qui veulent créer un réseau temporaire, par exemple un réseau Ad hoc qui relie les différentes unités portables (palmtop, PDA, notebook, laptops) peut être utilisé pour propager et partager les informations entre les participants dans une conférence[FRO 00].

D'une façon générale, les réseaux mobiles Ad hoc sont utilisés dans toute application où le déploiement d'une infrastructure réseau filaire est trop contraignant, soit parce qu'elle est difficile à mettre en place soit, parce que la durée d'installation du réseau ne justifie pas de câblage à demeure (temporairement).



1.7 Conclusion

Les réseaux informatiques basés sur la communication sans fil peuvent être classés en deux classes : les réseaux avec infrastructure fixe préexistante, et les réseaux sans infrastructure. La deuxième classe essaye d'étendre les notions de la mobilité à toutes les composantes de l'environnement, toutes les unités de cette classe(les réseaux Ad hoc), se déplacent librement et aucune administration centralisée n'est disponible. Nous avons essayé de donner les avantages de ce type de réseaux par rapport aux autres réseaux. Nous avons parler des domaines d'utilisation des réseaux Ad hoc qui sont considérés importants dans les situations où l'installation d'une infrastructure est coûteuse, difficile, impossible ou tout simplement inutile. Dans les réseaux mobiles Ad hoc, les mobiles sont à la fois des terminaux (PC, assistants personnels, téléphones, etc.) et des routeurs pour leurs voisins. Ils permettent d'étendre la couverture des réseaux sans fil, sans avoir à déployer de lourdes infrastructures. Dans le chapitre qui suit, nous allons définir le concept de routage ainsi que le problème de routage dans les réseaux Ad hoc suivie par une présentation synthétique des différentes solutions qui existent et qui résolvent un peu le problème de routage dans les réseaux mobiles Ad hoc.

Chapitre 2

Routage unicast dans les réseaux mobiles Ad hoc

2.1 Introduction

L'architecture d'un réseau Ad hoc est caractérisée par une absence d'infrastructure fixe préexistante. Contrairement aux réseaux classiques, un réseau Ad hoc doit s'organiser automatiquement de façon à être déployable rapidement et pouvoir s'adapter aux conditions de propagation, au trafic et aux différents mouvements pouvant intervenir au sein des nœuds.

Dans le but d'assurer la connectivité du réseau, malgré l'absence d'infrastructure fixe et la mobilité des stations, chaque nœud est susceptible d'être mis à contribution pour participer au *routage* et pour retransmettre les paquets d'un nœud qui n'est pas en mesure d'atteindre sa destination ; tout nœud joue ainsi le rôle de station et de routeur. Chaque nœud participe donc à un protocole de routage qui lui permet de découvrir les chemins existants, afin d'atteindre les autres nœuds du réseau. Le fait que la taille d'un réseau Ad hoc peut être énorme, souligne que la gestion de routage de l'environnement doit être complètement différente des approches utilisées dans le routage classique.

Le problème qui se pose dans le contexte des réseaux Ad hoc est l'adaptation de la méthode d'acheminement utilisée avec le grand nombre d'unités existant dans un environnement caractérisé par de modestes capacités de calcul et de sauvegarde.

Si un nœud source S désire communiquer avec un nœud destinataire D, il envoie le paquet à destination de D au nœud privilégié, qui se charge de le renvoyer à D. En l'absence d'un tel nœud, il faut procéder autrement. Ce que l'on souhaite, c'est que tout nœud

recevant de l'information dont il n'est pas le destinataire soit capable de la renvoyer à un autre nœud, plus proche que lui du destinataire en nombre de sauts. En d'autres termes, on souhaite que chaque nœud soit un routeur ou agit comme un routeur.
A cette fin, le groupe MANET de l'IETF[1] est chargé de développer et d'étudier des protocoles de routage, en vue de l'élaboration d'un standard.
Dans le cas où le nœud destination se trouverait dans la portée de communication du nœud source le *routage* devient évident et aucun protocole de routage n'est initié. Malheureusement, ce cas est généralement très rare dans les réseaux mobiles Ad hoc. Un nœud source peut avoir besoin de transférer des données à un autre nœuds qui ne se trouve pas dans sa portée de communication.

2.2 Définition du routage

Le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance. Le problème consiste à trouver l'investissement de moindre coût en capacités nominales et de réserves qui assure le routage du trafic nominal et garantit sa survabilité en cas de n'importe quelle panne d'arc ou de nœud [BAD 02].
Si on suppose que les coûts des liens sont identiques, le chemin indiqué dans la figure 2.1 suivante est le chemin optimal reliant le nœud source et le nœud destination. Une bonne stratégie de routage utilise ce chemin dans le transfert des données entres les deux nœuds[BAD 02].

2.2.1 Problème de routage

La question qui se pose, est avons nous besoin d'un protocole de routage pour les réseaux mobiles Ad hoc ? pourquoi on n'utilise pas les protocoles de routage conventionnels(classiques) comme les protocoles *d'état de lien* ou les protocole de *vecteur de distance* utilisés pour les réseaux filaires ? D'un coté, par ce qu'ils sont bien testés et ils ont montré un bon comportement (l'exemple d'Internet) et d'un autre coté, les administrateurs sont devenus familiarisés avec eux.
Avant de répondre à ces questions, Nous allons expliquer de façon générale les approches

[1] Internet Engineering Task Force un groupe de travail qui travaille sur la spécification des protocoles pour les réseaux sans fil Ad hoc.

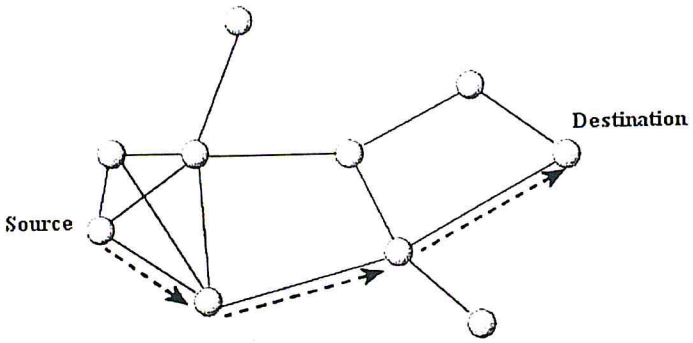


FIG. 2.1 – Le chemin utilisé dans le routage entre la source et la destination.

utilisées dans les protocoles(conventionnels) ainsi que les problèmes qui peuvent générer en cas d'utilisation des ces derniers dans les réseaux Ad hoc.

2.2.2 Protocoles conventionnels

Description

La première idée qui vient à l'esprit et afin de concevoir un protocole de routage pour les réseaux mobiles Ad hoc, est de considérer chaque nœud du réseau comme un routeur et d'utiliser les protocoles des réseaux conventionnels (filaires).
Les deux protocoles principaux les plus utilisés dans le monde des réseaux filaires sont :

1. *Protocoles d'état de lien* (LSP : Link State Protocol)
2. *Protocoles de vecteur de distance* (DVP : Distance Vector Protocol).

Dans le premier protocole, chaque nœud du réseau doit maintenir une vue globale de la topologie du réseau. Pour que cette vue soit à jour, chaque nœud *inonde* le réseau par l'état de lien des ses voisins a tous les nœuds du réseau. Pendant que chaque nœud reçoit cette information, il met à jour sa vue du réseau et applique un algorithme de plus court chemin (celui de Dijkstra) pour choisir le prochain-saut pour chaque destination. Le protocole OSPF (Open Shortest Path First) est le protocole d'état de lien le plus connu. L'inconvénient majeur de ces protocoles est la lourdeur des algorithmes et la taille mémoire nécessaire à l'exécution. C'est pour cela qu'on ne les trouve que sur des matériels haut de gamme avec beaucoup de mémoire centrale et des CPU rapides.
Dans l'approche de *vecteur de distance*, chaque nœud diffuse à ses voisins sa vision des distances qui le séparent de tous les nœuds du réseau. A partir des informations reçues,

chaque nœud de routage fait certain calcul, en utilisant l'algorithme distribué de Bellman-Ford, afin de trouver le chemin le plus court vers toutes les destinations possibles. Le processus de calcul se répète jusqu'à ce que le système *converge* vers un état stable. Le protocole RIP(Routing Information Protocol) est le protocole de vecteur de distance le plus connu. L'inconvénient de ces protocoles est la grande quantité d'information de routage générer sur les canaux de communication pour la diffusion de ces tables (toutes les 30s pour RIP).

Problème

Considérer chaque nœud comme un routeur et utiliser les protocoles de routage des réseaux filaires s'avère une idée mais dans le cas où on l'applique pour les réseaux sans fil génère beaucoup de problèmes[JOH 94], les problèmes sont résumés dans les points suivants :

- L'échange périodique des messages de mises à jour demandées par les protocoles de routage conventionnels, s'avère délicat pour les réseaux Ad hoc. La plupart des noeuds mobiles des réseaux Ad hoc utilisent des ressources d'énergies limitées comme les batteries (la transmission d'un paquet dissipe une partie de l'énergie de la batterie de nœud). La réception quant à lui, exige une énergie mais elle est moins petite que celle d'émission. La nécessité de recevoir ces mises à jour de routage périodiques empêche un nœud d'économiser sa propre énergie de batterie.
- L'échange périodique des messages de mises à jour de routage consomme également la bande passante de transmission : Les protocoles de routages conventionnels exigent, dans des périodes bien déterminées(mises à jour toutes les 30s pour RIP), des mises à jour de routage afin de maintenir la cohérence des informations de routage. Dans certain cas, malgré qu'il n'y a pas de changement au niveau des paquets de mise à jour(les mises à jour précédentes et les mises à jour courantes sont les mêmes), l'échange des messages se continue. Cela a comme conséquence une consommation importante de la bande passante du réseau.
- Les contraintes physiques limitées : Les routeurs utilisés actuellement dans les réseaux câblés sont caractérisés par des CPUs de haute gamme et des unités de calcul très performantes, dont les unités mobiles des réseaux Ad hoc en sont dépourvues. Les nœuds des réseaux Ad hoc peuvent être des PDAs (Personnel Digital Assistant), Laptops, Agendas électroniques, etc, qui se caractérisent par des CPUs moins puis-

santes que celles utilisées dans les routeurs(des réseaux conventionnels). En plus, ces derniers (PDAs, Agendas, ...) ont des mémoires très limités (à peu près 16Mo pour les PDAs) ce qui rendent la sauvegarde des tables de routage impossibles notamment dans les réseaux qui possèdent des milliers de nœuds.

- Les liens bidirectionnels : Les liens dans les réseaux filaires sont bidirectionnels. Si un nœud x a pu transférer des données à y on peut conclure que y, peut à leur tour transférer des données vers x, ce qui est n'est pas le cas dans les réseaux Ad hoc, ceci est dépend de la portée de communication dans les réseaux Ad hoc.

Par conséquent, les protocoles de routage pour les réseaux filaires ne peuvent pas être directement utilisés dans les réseaux sans fil ; et des protocoles de routage spécifiques pour les réseaux Ad hoc sans fil doivent être conçus.
Pour juger les performances d'un protocole de routage dans un réseau Ad hoc, il faut que ce protocole possède certaines critères de performances.
Avant d'exposer ces critères, il faut commencer par la définition de quelques notions qui vont aider à la compréhension des ces critères et aider également à la compréhension des protocoles de routage présentés dans le chapitre qui suit.

2.3 Notions générales

2.3.1 Inondation(Flooding)

L'inondation ou la diffusion pure, consiste à faire propager un paquet (de données ou de contrôle) dans le réseau entier. Un nœud qui initie l'inondation envoie le paquet à tous ses voisins directs. De même, si un nœud quelconque du réseau reçoit le paquet, il le rediffuse à tous ses voisins. Ce processus se répète jusqu'à ce que le paquet atteigne tous les nœuds du réseau. Notons que les nœuds peuvent être amener à appliquer - durant l'inondation - certains traitements de contrôle, dans le but d'éviter certains problèmes, tel que le bouclage et la duplication des messages.
Le mécanisme d'inondation est utilisé généralement dans la première phase du routage plus exactement dans la *procédure de découverte* des routes, et cela dans le cas où le nœud source ne connaît pas une route vers la destination.

2.3.2 Vecteur de distance (Distance Vector)

Dans le routage de vecteur de distance, chaque nœud de réseau diffuse à ses nœuds voisins, sa vision des distances qui le séparent de tous les nœuds du réseau. Quand un nœud reçoit des vecteurs de distance de ses voisins, il calcule de nouvelles routes et met à jour son vecteur de distance. Le processus est poursuivi jusqu'à ce que le réseau atteigne un état stable. Le problème de ces algorithmes est la convergence lente des réseaux utilisant cet algorithme et le risque de formation des boucles dans le réseau. Cette technique est basée sur l'algorithme distribué de Bellman-Ford[STE 95].

2.3.3 État de lien(link state)

Dans le routage d'état de lien, chaque nœud maintien une vue globale de la topologie du réseau avec un coût pour chaque lien. Pour maintenir ces coûts conformés ; chaque nœud diffuse périodiquement les coûts de lien de ses liens sortants à tous les autres nœuds en utilisant l'inondation (flooding). Quand un nœud reçoit cette information, il met à jour sa vue du réseau et applique un algorithme de plus court chemin pour choisir le prochain-saut pour chaque destination[STE 95].

2.3.4 Routage de source (Source Routing)

Le routage de source de l'anglais "Source Routing" consiste à indiquer dans le paquet routé l'intégralité du chemin que devra suivre le paquet pour atteindre sa destination. L'entête du paquet va donc contenir la liste des différents nœuds intermédiaires vers la destination.

2.3.5 Graphe Acyclique Orienté (DAG : Direct Acyclic Graph)

On dit q'un graphe orienté est un DAG pour le sommet (destination) j si, à partir de n'importe quel autre sommet (i, j) il existe au moins un chemin vers j.

2.3.6 Groupe

Une des architectures qui est utilisée dans les réseaux Ad hoc, est la décomposition de ces réseaux en un ensemble de groupes, chaque groupe contient un certain nombre de nœuds, à partir de ces derniers, un nœud doit être élu (représentant) pour représenter les autres nœuds membre de groupe. Les représentants de groupe assurent le routage des données et

contrôle l'utilisation du canal par leurs membres de groupe, ils sont prioritaires d'accéder au canal et donc ils ont plus de chance de transmettre que d'autres nœuds.

2.3.7 Hiérarchie

On peut étendre la notion de groupe vue précédemment pour former des niveaux hiérarchiqu de groupes, les représentant de groupes dans un niveau i, deviennent des membres de groupes dans un niveau $i+1$, ces nouveaux membres s'organisent en un ensemble de groupes de la même manière que le niveau bas, et ainsi de suite jusqu'on arrive à un niveau ou il n'existe q'un seul représentant de groupe.

2.3.8 Routage saut par saut (hop by hop)

Le routage saut par saut de l'anglais "hop by hop" consiste à donner uniquement à un paquet l'adresse du prochain saut vers la destination. Ce type de routage est le routage le plus répandu dans l'Internet, c'est le système de routage par défaut.

2.3.9 Système de positionnement Global (GPS :Global Positioning System)

Le système de positionnement global (GPS) est un système de navigation par satellite composé d'un réseau de 24 satellites placés en orbite[2] par le département de la Défense aux ETATS-UNIS. Le GPS a été à l'origine prévu pour des applications militaires, mais dans les années 80, le gouvernement a rendu le système disponible pour un usage civil. Les satellites de GPS font le tour de la terre deux fois par jour dans une orbite très précise et transmettent l'information de signal à la terre. Les récepteurs de GPS prennent cette information et emploient la triangulation pour calculer l'endroit exact de l'utilisateur. Les GPS sont précis à 15 mètres en moyenne. Le GPS a besoin de 3 satellites pour avoir une position en 2D et 4 satellites pour une vision en 3D (latitude, longitude et altitude).

2.3.10 Inversement de lien

C'est une technique utilisée par les algorithmes de routage basée sur la construction de DAGs, elle est utilisée pour reconstruire un DAG quand un sommet perd son dernier chemin vers la destination à cause d'une défaillance de lien, ceci est réalisé par inversement d'un ensemble de liens.

[2] à environ 20 000 Kms de la terre.

2.4 Propriétés désirées d'un protocole de routage efficace

Actuellement, il existe plusieurs protocoles proposés pour effectuer le routage dans les réseaux Ad hoc(voir le chapitre suivant), certains d'entre eux ont des avantages que les autres n'en possèdent pas. Mais en fin de compte, il y a quelques propriétés désirable que n'importe quel protocole de routage dans les réseaux Ad hoc devrait les posséder [JOH 94][LAR 98], Il semble donc important que toute conception de protocole de routage doive satisfaire les propriétés suivantes :

Traitement distribué Le protocole devrait être distribué. Il ne devrait pas dépendre d'une station de contrôle centralisée comme dans le cas des réseaux filaires. La différence est qu'à un moment donné, un mobile peut rejoindre ou quitter le réseau Ad hoc existant et en raison de la mobilité des nœuds le réseau peut être partitionné.

L'absence de boucle Elle est cependant souhaitable pour éviter certains phénomènes qui arrivent dans le pire des cas, comme par exemple, quelques paquets tournant autour du réseau pendant un temps non défini. Cela a comme conséquence une consommation considérable de la bande passante et du temps CPU.

Réduire la charge du réseau Pour réduire au minimum le nombre de messages de contrôle dans le réseau et de ne pas gaspiller des ressources de réseau plus que nécessaires, le protocole devrait être *réactif*. Ceci signifie que le protocole devrait seulement réagir à la demande et que le protocole ne devrait pas périodiquement diffuser des informations.

Supporter des liaisons unidirectionnelles Les liaisons dans les réseaux Ad hoc peuvent être unidirectionnelles, si les nœuds ont des puissances de transmission inégales. Dans ce cas, l'un peut recevoir l'autre sans que la réciproque ne soit vraie ; la liaison peut aussi être unidirectionnelle à cause du phénomène physique même si les équipements ont les mêmes caractéristiques en particulier dans les réseaux de type sans fil. Ce caractère unidirectionnel d'une liaison est une caractéristique qui n'apparaît pas en général pour une liaison filaire.

Conservation d'énergie Les nœuds du réseau Ad hoc utilisent des batteries ou un autre moyen limité pour puiser leur énergie. Pour ces nœuds, le plus important est sans doute de mettre en place des critères d'optimisation pour la conservation de l'énergie(voire la notion suivante).

Routes multiples Afin de réduire les réactions aux changements topologiques dues aux

mobilités des nœuds et réduire au minimum la congestion (la charge) du réseau, des routes multiples à cet égard peuvent être utilisées : Si une route est devenue *invalide*, il est possible qu'une route enregistrée pour la même destination puisse être utilisées et de cette manière permet au protocole de ne pas initier (déclencher) de nouveau la *procédure de recherche* de route.

Supporter la qualité de service Développer des protocoles de routage capables de transporter divers flux (données, audio, vidéo) en respectant des contraintes comme le délai de transfert de bout en bout, le débit ou le taux de perte des paquets afin d'offrir une bonne qualité.

Fiabilité Le protocole doit garantir la continuité de service de routage malgré la défaillance de nœuds ou de liens de communication tant qu'il y a d'autres routes possibles vers la destination.

2.5 Conclusion

Dans ce chapitre nous avons présenté le concept du réseau Ad hoc et nous avons présenté d'une manière générale les techniques utilisées par les protocoles de routage conventionnels pour l'acheminement des données. Nous avons monté que ces derniers n'assure pas une bonne performance de routage en cas d'utilisation des réseaux Ad hoc, d'où la nécessité de concevoir un protocole spécifique aux réseaux Ad hoc.
Nous avons vu également les critères qui peuvent être vérifier pour qu'un protocole de routage soit plus performant. Quelques notions nécessaires à la compréhension du concept de routage dans les environnements mobiles ont été également présentées.
La plupart des études faites jusqu'à maintenant, avaient comme objectif d'améliorer les stratégies de routage des réseaux filaires pour qu'ils s'adaptent aux réseaux sans fil et proposer éventuellement de nouveaux protocoles de routage qui doivent prendre en considération les limitations des réseaux Ad hoc et offrir une meilleure adaptation à la mobilité de ces environnements.
Dans les deux chapitres qui suivent, nous allons décrire les principales caractéristiques et fonctionnalités des protocoles de routage les plus connus proposés permettant l'acheminement des données entre les différents nœuds mobiles. Le chapitre 3, va présenter les protocoles de routage appartenant à la famille des protocoles de routage non prédictifs, tandis que le quatrième chapitre présente les protocoles de routage prédictifs.

Chapitre 3

Protocoles de routage non prédictifs

3.1 Introduction

L'objectif essentiel d'un protocole de routage dans un réseau Ad hoc est, l'établissemei de la route d'une manière correcte et efficace entre deux nœuds de telle sorte que les paquets de données soient délivrés d'une façon opportune. La construction de la route doit être faite avec un minimum de consommation de bande passante dans le réseau due aux échanges des paquets de contrôles entre nœuds, etc.

Les protocoles de routage non prédictifs dans les réseaux Ad hoc sont divisés en trois modèles de fonctionnement[ZOU 04] : les protocoles *proactifs*, protocoles *réactifs* et protocoles *hybrides*(Voir la figure 3.1). On peut les différencier par la méthode utilisée pour découvrir le chemin entre le nœud source et le nœud destination. Les protocoles proac-

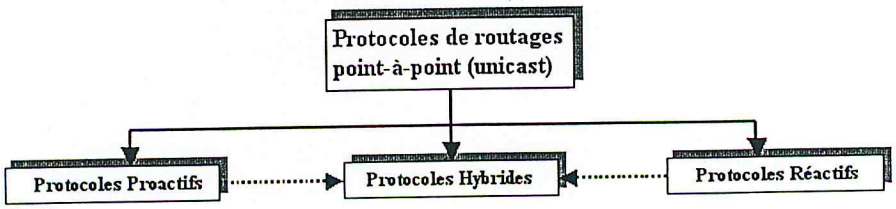


FIG. 3.1 – Classification des protocoles de routage dans les réseaux Ad hoc.

tifs établissent les routes à l'avance en se basant sur l'échange périodique des tables de routage. Quand un paquet doit être transmis, sa route est alors connue à l'avance et peut ainsi être immédiatement utilisée, alors que les protocoles réactifs, commencent la recherche d'une route avant de transmettre un paquet. Les protocoles hybrides ont un comportement proactif si la cible est à moins d'un nombre fixe de sauts, et un compor-

tement réactif au-delà.
Dans ce qui suit nous allons présenter les protocoles les plus connus, proposés pour effectuer le routage dans les réseaux Ad hoc. Nous décrivons leurs principales caractéristiques et fonctionnalités qui permettent d'assurer l'acheminement des données entre les différentes unités mobiles.

3.2 Protocoles de routage proactifs

Ils sont basés sur les protocoles de routage utilisés dans les réseaux filaire. Chaque nœud maintient, localement, une table de routage qui donne, en fonction de l'adresse de la cible, l'adresse du voisin auquel il faudra transmettre la trame, pour que celle-ci atteigne finalement sa destination. Du fait de l'aspect dynamique de la topologie des réseaux Ad hoc, la maintenance des tables de routage nécessite l'envoi périodique par chaque nœud de messages de contrôle indiquant sa présence à tous ses voisins.

3.2.1 Destination Sequenced Distance Vector(DSDV)

Le protocole de routage de *vecteur de distance ordonnancé par destination* est basé sur l'algorithme distribué de Bellman-Ford (algorithme de vecteur de distance). Chaque nœud maintient une table de routage contenant : Les adresses de toutes les destinations possibles, le nombre de saut nécessaire pour atteindre la destination et un numéro de séquence qui correspond à un nœud destinataire. Afin de maintenir la consistance des tables de routage(table mise à jour) dans des topologies qui varient rapidement, chaque nœud du réseau diffuse périodiquement sa table de routage à ses voisins directs. Afin d'éviter le problème de boucle qui peut se présenter dans le réseau, DSDV étiquette chaque entrée de la table de routage avec un numéro de séquence, ce numéro est utilisé également pour faire la distinction entre les anciennes et les nouvelles routes[PER 94].

3.2.2 Wireless Routing Protocol(WRP)

WRP est un autre protocole basé sur l'algorithme de vecteur de distance conçu pour le réseau Ad hoc. WRP modifie et met en valeur le vecteur de distance des trois voies suivantes. ***Premièrement***, quand il n'y a aucun changement de lien, au lieu d'échanger la totalité de table des routes et pour s'assurer que la connectivité avec un voisin existe toujours, un message de mise à jour périodique(message "HELLO") est envoyé aux

voisins sans qu'il y ait un changement au niveau de la table de routage. Quand il y a des changements de topologie, seules les entrées qui ont subit un changement seront envoyées. ***Deuxièmement.*** Afin d'améliorer la fiabilité de transfert, WRP oblige chaque voisin d'envoyer un ACK pour chaque paquet de mise à jour reçu. Des retransmissions sont envoyées si aucun ACK n'est reçu au cours d'un temps bien déterminé(timeout). ***Troisièmement.*** WRP inclus l'ID de l'avant dernier nœud(prédécesseur) pour chaque destination, cette information permet au protocole de calculer périodiquement le chemin de la source à la destination et d'éviter la formation des boucles ce qui réduit le temps de convergence du réseau et évite le problème de comptage à l'infinie[1](count-to-infinity)[ZHU 02][MUR 96].

3.2.3 Global State Routing(GSR)

Le protocole de *Routage à Etat Global*, est similaire au protocole DSDV. GSR utilise l'idées du routage basé sur l'état des liens (LS : link state), et les améliore en évitant le mécanisme inefficace de l'inondation des messages de routage. GSR utilise une vue globale de la topologie du réseau, comme c'est le cas dans les protocoles LS. La différence principale entre GSR et LS traditionnel est la façon dont les informations de routage circulent dans le réseau. Dans LS, si on détecte des changements de la topologie, les paquets des états de lien sont générés et diffusés par inondation dans le réseau entier. Par contre, GSR maintient la table la plus récente d'état des liens reçus à travers les voisins, et l'échange d'une façon périodique uniquement avec ses voisins locaux.
Chaque nœud i dans le réseau maintient une liste des voisins A_i, une table de topologie TT_i, une table des nœuds suivants $NEXT_i$ et une table de distance D_i. Pour chaque destination j, la table $NEXT_i(j)$ contient le nœud vers lequel les paquets destinés à j seront envoyés. $D_i(j)$ contient la plus courte distance pour chaque nœud destination.
Comme les protocoles LSs, les messages de routage sont générés suivant les changements d'états des liens. Lors de la réception d'un message de routage, le nœud met à jour sa table de topologie et cela dans le cas où le numéro de séquence du message reçu serait supérieur à la valeur du numéro de séquence sauvegardée dans la table (comme DSDV). Par la suite, le nœud reconstruit sa table de routage et diffuse les mise à jour à ses voisins. Le calcul des chemins, peut se faire avec n'importe quel algorithme de recherche des plus courts chemins. Par exemple, GSR utilise l'algorithme de Dijkstra modifié de telle sorte

[1]Problème pouvant se produire avec les algorithmes de routage qui convergent lentement. Les routeurs incrémentent continuellement le compte de sauts (hops count) de certains réseaux.

qu'il puisse construire la table des nœuds suivants (NEXT) et la table de distance (D) en parallèle avec la construction de l'arbre des plus courts chemins.
Les inconvénients de GSR sont [ITA 01] la grande taille des messages de mise à jour qui consomme beaucoup de bande passante et le temps d'attente de la propagation de changement d'état de lien. La technique de FISHEYE vient comme solution afin éviter ce problème, en réduisant la taille du message de mise à jour[CHE 98].

3.2.4 Fisheye State Routing (FSR)

Le *routage d'état d'œil de poisson* (FSR) est une amélioration de GSR, il est basé sur l'utilisation de la technique "œil de poisson ". FSR divise le réseau entier en différentes portées basées sur une métrique prédéfini(La portée est définie en termes des nœuds qui peuvent être atteints par un certain nombre de sauts(voir la figure 3.2).
La grande taille des messages de mise à jour dans GSR gaspille une quantité considérable

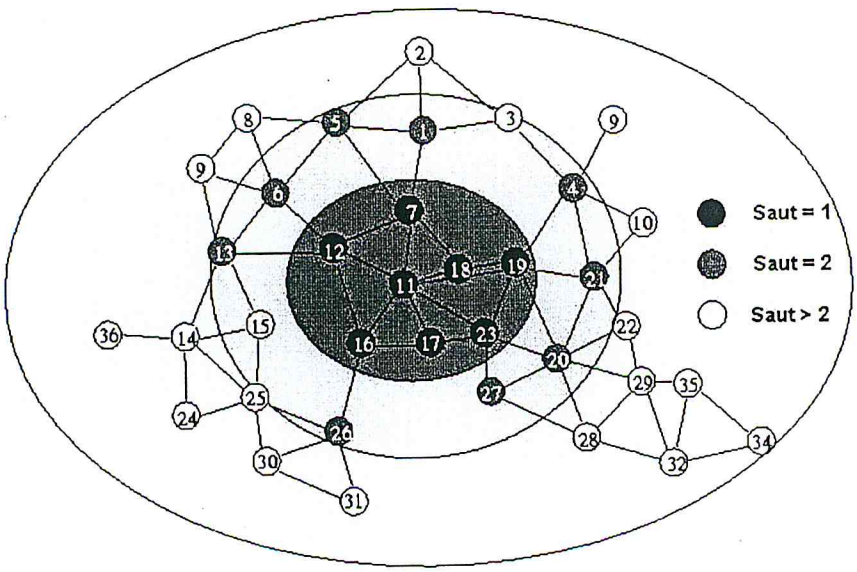


FIG. 3.2 – La portée(champ de vision) de Fisheye.

de bande passante du réseau. FSR améliore l'algorithme d'état de lien de la manière suivante : **Premièrement :** La diffusion par inondation des paquets d'état de lien n'existe pas : l'échange se fait uniquement avec les voisins directs. **Deuxièmement :** Le déclenchement des échanges est basé sur les périodes non aux évènements. **Troisièmement :** Au lieu de transmettre l'information entière d'état de lien à chaque itération, FSR utilise différents intervalles de transmission pour les différentes entrées de table. Plus

précisément, Les entrées qui correspondent aux nœuds les plus proches sont envoyées aux voisins avec une fréquence élevée (donc avec une période d'échange relativement petite). Le reste des entrées, est échangé avec une fréquence moins élevée. De cette manière, un grand nombre de données de routage est évité, ce qui réduit le volume des messages qui circule dans le réseau[KLE 91].

3.2.5 Clusterhead Gateway Switch Routing Protocol (CGSR)

Ce protocole est similaire au DSDV décrit précédemment. Mais au lieu d'avoir un grand réseau plat (pas des nœuds privilégiés), CGSR le décompose en groupes de nœuds. Un représentant pour chaque groupe appelé clusterhead (chef de groupe) est élu. Chaque clusterhead contrôle un plus petit groupe de nœuds aussi bien que des Gateways, (Gateway ou passerelle est un nœud qui appartient à la portée de communication de deux ou plusieurs représentants de groupe[ROY02]). Les nœuds d'un groupe peuvent communiquer avec leur clusterhead et éventuellement entre eux(voir la figure 3.3). Quand

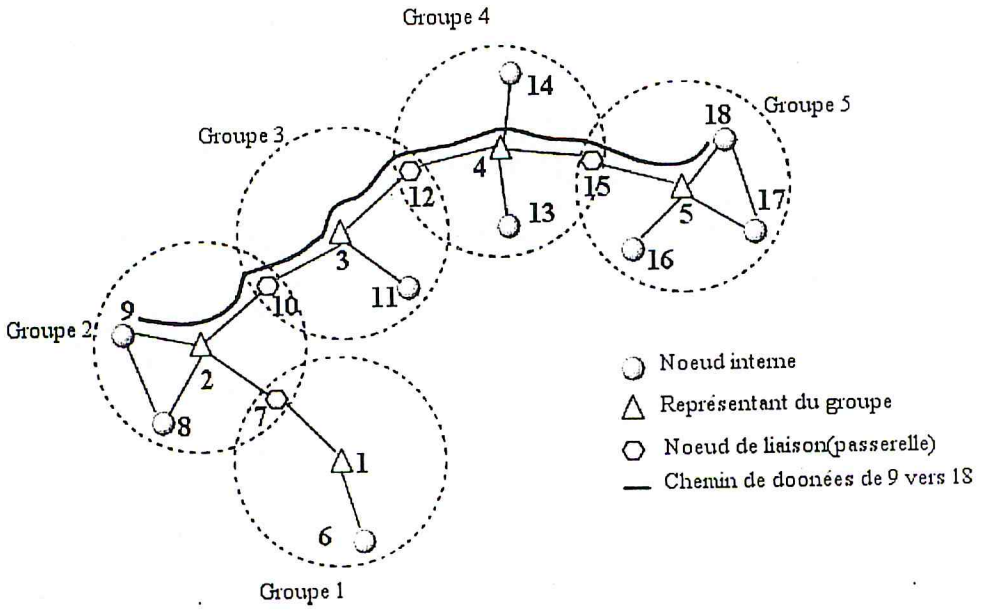


FIG. 3.3 – Classification des protocoles de routage dans les réseaux Ad hoc.

un paquet est envoyé à un nœud, il va d'abord au chef de groupe, si la destination n'est pas dans le même groupe. Le chef de groupe transmet le paquet via la passerelle(Gateway) pour un autre groupe : les paquets sont acheminés alternativement par des clusterheads et des passerelles, la route typique ressemble à $C_1G_1C_2G_2 \ldots C_iG_i$ où C_i est un clusterhead

et G_i est une passerelle. Ce processus se continue jusqu'à ce que la destination soit atteinte.

Pour empêcher les changements fréquents des clusterheads, CGSR utilise l'algorithme LLC(Least Clusterhead Change). Dans cet algorithme un nouveau chef de groupe est choisi chaque fois qu'un nœud devient déconnecté de n'importe quel groupe ou quand deux clusterheads deviennent voisins. L'avantage principal de CGSR est que seulement les routes aux clusterheads sont maintenues. Cependant, le maintien des groupes cause une charge supplémentaire. Spécifiquement, chaque nœud doit périodiquement annoncer sa table de membre de groupe et mettre à jour sa table basée sur les mises à jour reçues[CHI 97].

3.2.6 Hierarchical State Routing(HSR)

Le protocole *Routage à Etat Hiérarchique* combine les notions de groupes dynamiques et de niveaux hiérarchiques avec une gestion efficace de localisation. Dans HSR, l'image de la topologie du réseau est sauvegardée sous forme hiérarchique, le réseau est partitionné en un ensemble de groupes, dont l'union donne le réseau entier. Dans un groupe, un nœud doit être élu pour représenter le reste des membres. Les représentants des groupes dans un niveau l, deviennent des membres dans le niveau $l+1$. Ces nouveaux membres, s'organisent en un ensemble de groupes de la même manière du niveau bas, et ainsi de suite pour le reste des niveaux(voir la figure 3.4). Pour acheminer une information du nœud 5 vers le nœud 10, le nœud 5 envoie l'information au nœud supérieur, qui le suit hiérarchiquement, i.e. le nœud d'ID 1. Le nœud 1 délivre l'information au nœud 3 qui suit le nœud destination dans l'ordre hiérarchique. Un "lien virtuel" existe entre les nœuds 1 et 3, matérialisé par le chemin $[1, 6, 2, 8, 3]$. Par conséquent, l'information suivra ce chemin pour atteindre la destination. Dans la dernière étape, le nœud 3 délivre l'information au nœud 10 en suivant le chemin hiérarchique qui le relie à la destination (dans notre cas, ce chemin se réduit en un seul saut).

Les identificateurs (IDs) des nœuds représentés dans la figure 3.4(niveau 0) sont des adresses physiques : ils sont uniques pour chaque nœud. Une des méthodes qui peut être appliquée afin d'associer des adresses hiérarchiques, ou $HIDs$ (Hierarchical IDs) aux différents nœuds est de prendre les numéros des groupes dans le chemin reliant la racine et le nœud en question. Cela ne pose aucun problème, car le nœud peut être atteint à travers de ces adresses, et ces dernières sont associées à un nœud unique. On peut toujours

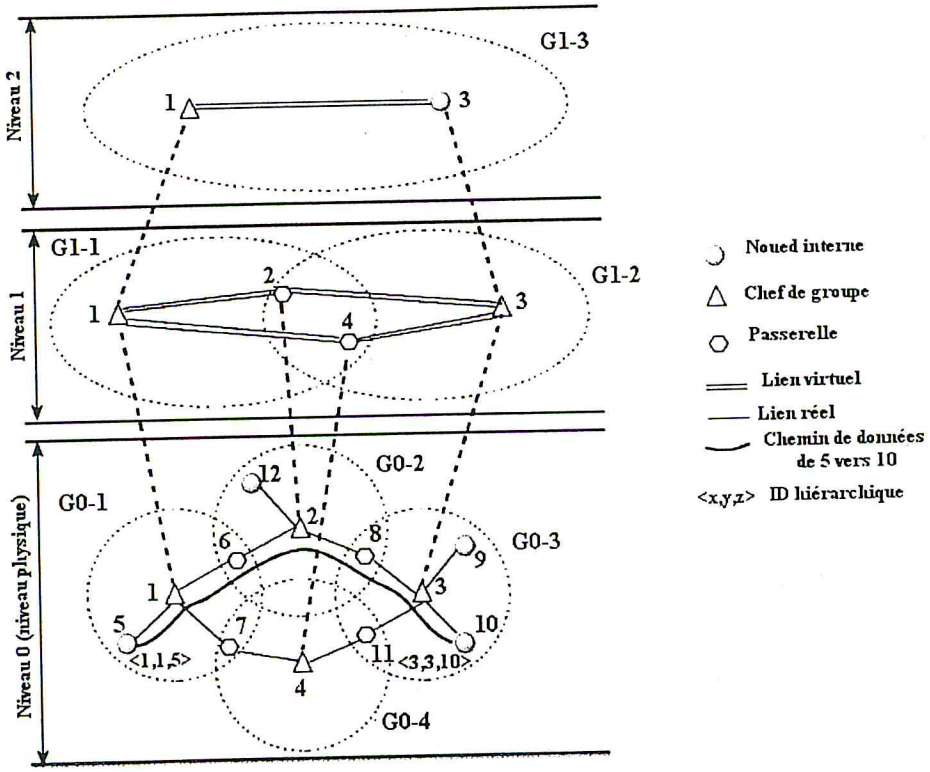


FIG. 3.4 – Exemple de groupe, Formation de hiérarchie dans le protocole HSR.

trouver une manière d'associer une seule adresse à ce genre de nœuds, par exemple en prenant la plus petite valeur des numéros de groupes dans les quels appartient le nœud. Chaque subnet correspond à un groupe particulier d'utilisateurs qui possède un *serveur de gestion de localisation* dit LMS (Location Management Server). Différents ensembles de mobiles peuvent être définis indépendamment dans chaque subnet. Quand la couche de transport délivre au réseau un paquet contenant l'adresse IP privée, le réseau doit trouver, à partir de l'adresse IP, l'adresse hiérarchique basée sur l'adresse physique[PEI 99].

3.2.7 Zone-based Hierarchical Link State Routing(ZHLS)

Le protocole de *Routage à Etat de Liens Hiérarchique basé sur les Zones*, divise le réseau en un ensemble de zones non-recouvertes (voir la figure 3.5). A la différence des autres protocoles hiérarchiques, la notion d'élection des représentants pour chaque zone n'existe pas. Avec cette décomposition, on a deux niveaux de topologies : Le *niveau nœud*, et le *niveau zone*. La topologie basée sur le premier niveau indique que les nœuds d'une zone sont reliés entre eux physiquement. Un lien virtuel entre deux zones existe si

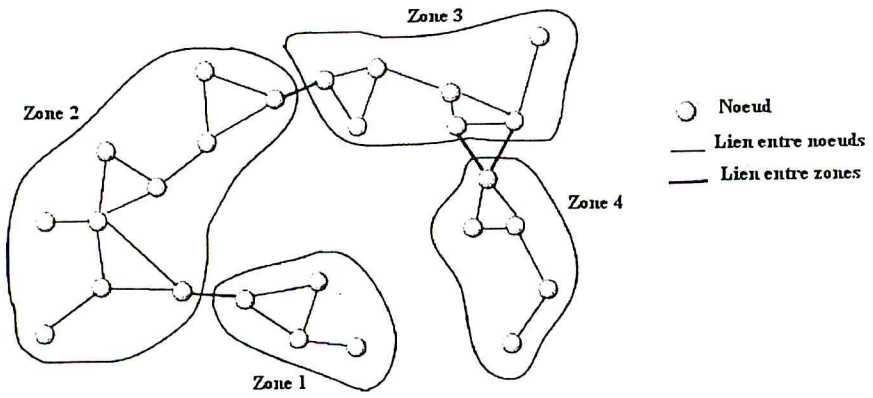


FIG. 3.5 – La décomposition du réseau en zones.

au moins un nœud d'une zone est physiquement relié à un certain nœud de l'autre zone. La topologie de niveau zone indique comment les zones sont reliées ensemble. ZHLS utilise deux types de paquets d'état de lien (LSP :Link State Packet) : *LSPs orienté nœuds* contient l'information d'un nœud voisin, ce type de paquet circule et échangé seulement à l'intérieur de la zone. *LSPs orienté zone* contient l'information de la zone, et il est propagé globalement. De cette façon, chaque nœud du réseau possède une connaissance complète concernant les nœuds de sa propre zone, et seulement une connaissance partielle du reste des nœuds.

Par conséquent, le routage des paquets se fait de deux façons : Le routage inter zone, et le routage intra zone, i.e. pour envoyer un paquet d'une certaine source vers une certaine destination, les paquets de données sont conduits en se basant sur l'identification de zone (routage intra zone), jusqu'à ce que les paquets atteignent la zone finale de destination, ensuite l'algorithme de routage inter zone va déclencher. Les paquets circulent dans cette zone en se basant sur l'identificateur de nœud destination. Donc l'adresse $< IDzone, IDnœud >$ est suffisante pour atteindre n'importe quelle destination même s'il y a un changement de topologie[LOA 99].

3.2.8 Distance Routing Effect Algorithm for Mobility (DREAM)

Le protocole d'*Algorithme d'Effet de Routage basé sur la Distance pour la Mobilité*, est un protocole proactif basé sur les informations de localisation des unités mobiles. Le protocole diffuse les données destinées aux destinations en effectuant une inondation partielle cela afin de minimiser la charge du réseau. Chaque nœud mobile envoie périodiquement des messages de contrôle afin d'informer tous les autres nœuds de sa

localisation. La distance influe dans cet échange, du fait que les messages de contrôle sont envoyés fréquemment aux nœuds les plus proches.
Lors de l'envoi des données, si la source possède des informations récentes sur la localisation du nœud destination, elle choisit un ensemble de nœuds voisins qui sont localisés dans la direction source/destination. Si un tel ensemble n'existe pas, les données sont inondées dans le réseau entier. Dans le cas où de tels nœuds existeraient, une liste qui contient leurs identificateurs est insérée à l'entête du paquet de données avant la transmission, et seuls les nœuds qui sont spécifiés dans la liste de l'entête traitent le paquet. Lors de la réception du paquet, le nœud intermédiaire détermine sa propre liste de nœuds proches, et envoie le paquet avec la nouvelle liste de tête. Si aucun voisin n'est localisé dans la direction de la destination, le paquet reçu est ignoré.
Quand le nœud destination reçoit les données, il envoie des acquittements à la source d'une manière similaire. Cependant, dans le cas de réception par inondation, les acquittements ne sont pas envoyés. Si aucun acquittement n'est reçu avant l'expiration du timeout, les données seront retransmises en utilisant une diffusion ordinaire[BAS 98].

3.3 Inconvénients des protocoles de routage proactifs

Dans la famille des protocoles de routage proactifs, les nœuds du réseau essaient de maintenir(sauvegarder) les meilleurs chemins existants vers toutes les destinations possibles. Les routes sont sauvegardées même si elles ne sont pas utilisées. La sauvegarde permanente des chemins est assurer par un échange contenu des messages de mise à jour, ce qui induit un contrôle excessif notamment dans les grands réseaux. En plus, si la topologie du réseau change fréquemment, alors une partie significative de la largeur de bande peut être occupée. Donc, pour remédier à ces problèmes, la famille des protocoles de routage réactifs est proposée.

3.4 Protocoles de routage Réactifs

Ils ne nécessitent pas le maintien des tables de routage dans chaque nœud, car ils sont basés sur une construction à la demande des routes. Lorsqu'un nœud S désire communiquer avec un nœud D, il commence par la construction d'une route vers D, en envoyant un paquet de requête de route à tous les nœuds qui peuvent l'entendre. Ce paquet, transmis de proche en proche sur tout le réseau jusqu'à son arrivée à D où un

nœud connaissant une route vers D, accumule, au fur et à mesure de son parcours, des informations sur le chemin parcouru. A son arrivée en D, un paquets de réponse de route est envoyé et retourné vers S.
La communication entre S et D peut alors commencer.
Dans l'approche réactive(à la demande), des paquets de requête/réponse sont utilisés pour découvrir une route à une destination donnée. Ces paquets de contrôle sont plus petits que les paquets de contrôle utilisés pour les mises à jour de table de routage dans l'approche proactive, ce qui réduit considérablement le trafic induit durant la mise à jour.
Parmi les protocoles de routage réactifs qui existent on trouve AODV, DSR, ABR, SSA, TORA et LAR.

3.4.1 Ad hoc On-demand Distance Vector (AODV)

Le protocole AODV (Ad hoc On-demand Distance Vector) est un protocole réactif qui est en effet une amélioration du protocole DSDV dans le but de réduire le nombre de messages diffusés pour la maintenance des routes. AODV ne maintient pas une entrée pour tous les nœuds, au lieu de cela chaque nœud stocke que les routes actives.
Un nœud veut envoyer de données vers une certaine destination, diffuse un paquet de *requête de route* dans le réseau si la route vers la destination n'existe pas. Chaque paquet de requête de route est muni d'un *numéro de séquence* de la destination. La requête ensuite est inondée dans le réseau jusqu'à ce qu'elle atteigne la destination ou un nœud qui a une route vers la destination désirée. Les nœuds recevant la requête de route, créent une route inversée vers le nœud émettant la requête.
Une fois la requête atteigne la destination ou un nœud connaissant une route vers la destination, un paquet de *réponse de route* est envoyé vers le nœud source(initiateur de la requête de route). Chaque réponse de route comporte le nombre de sauts nécessaire pour atteindre la destination. Les routes sauvegardées dans les tables de routage des nœuds ne rappellent qu'aux prochains sauts et ne rappellent pas la totalité des routes.
Afin de maintenir la connectivité des voisins, une version d'AODV, exige que chaque nœud doit transmet périodiquement un message *HELLO*. L'absence de réception de trois messages HELLO consécutifs d'un voisin, est prise comme indication que le lien vers ce voisin est devenu défaillant. le nœud détectant la défaillance d'un lien, envoie un paquet d'*erreur de route* au nœud source de la route. Une fois le paquet d'erreur de route est reçu par la source, cette dernière doit établir une nouvelle route à la destination en utilisant la

procédure de la découverte de route décrit ci-dessus. AODV ne peut fonctionner qu'avec les liens bidirectionnels ce qui représente un inconvénient[PER 99].

3.4.2 Dynamic Source Routing(DSR)

Avec ce protocole, quand un nœud veut envoyer un paquet de données vers un nœud destinataire dont il ne connaît pas la route, il envoie un paquet de requête de route, par inondation. Chaque voisin reçoit le paquet, ajoute son adresse dans l'en-tête, et envoie à son tour le message à tous ses voisins. Dès que le destinataire ou un nœud connaissant le chemin reçoit ce message, il envoie un paquet de réponse de route à la source en utilisant la route spécifiée dans l'en-tête. Le nœud source stocke les chemins connus dans son cache et commence à envoyer les données . Les nœuds intermédiaires qui reçoivent les paquets de réponse se servent aussi de l'information pour mettre son cache à jour. De même, quand une route n'existe plus, le dernier nœud valide envoie un paquet d'erreur de route vers la source. Tous les nœuds qui reçoivent ce paquet suppriment la route de leur cache. Le problème de cet algorithme : le surdébit est important car l'en-tête du paquet, notamment dans les grands réseaux, est que beaucoup plus volumineux vu que celui-ci doit contenir toute la route à suivre[DAV 01].

3.4.3 Temporary Ordred routing Algorithm(TORA)

L'*Algorithme de Routage Ordonné Temporairement* ou TORA a été conçu principalement pour minimiser l'effet des changements de la topologie.
L'algorithme s'adapte à la mobilité de l'environnement en stockant plusieurs chemins vers une même destination, ce qui fait que beaucoup de changements de topologie n'auront pas d'effets sur le routage des données, à moins que tous les chemins qui mènent vers la destination soient perdus. La principale caractéristique de TORA est que les messages de contrôle sont limités à un ensemble réduit de nœuds. Cet ensemble représente les nœuds proches du lieu de l'occurrence du changement de la topologie.
Dans ce protocole, la sauvegarde des chemins entre une paire (source, destination) donnée ne s'effectue pas d'une manière permanente : les chemins sont crées et stockés lors du besoin, comme c'est le cas dans tous les protocoles de cette catégorie. L'optimisation des routes (i.e. l'utilisation des meilleurs chemins) a une importance secondaire, les longs chemins peuvent être utilisés afin d'éviter le contrôle induit par le processus de découverte de nouveaux chemins. Ce pouvoir d'initier et de réagir d'une façon non fréquente sert à

minimiser le temps de communication de contrôle utilisé pour découvrir continuellement le meilleur chemin vers la destination.
L'algorithme TORA appartient à la classe des algorithmes appelée la classe "Inversement de Liens" (Link Revesral), très différente des deux classes, état de lien et Vecteur de Distance. TORA est basé sur le principe des algorithmes qui essaient de maintenir la propriété appelée "orientation destination" des graphes acycliques orientés (ou DAG : Directed Acyclic Graph). Un graphe acyclique orienté pour une destination s'il y a toujours un chemin possible vers une destination à partir de n'importe quel nœud. Ainsi, on peut dire que le protocole utilise le concept d'inversement de liens.
Ce concept assure la transformation du graphe précédent en un graphe orienté destination durant un temps fini. Afin de maintenir le DAG orienté destination, l'algorithme TORA utilise la notion de taille de nœud. Chaque nœud possède une taille qu'il échange avec l'ensemble de ses voisins directs. Cette nouvelle notion est utilisée dans l'orientation des liens du réseau. Un lien est toujours orienté du nœud qui a la plus grande taille vers le nœud qui la plus petite taille. Les concepts de taille et d'inversement de liens sont orientés destination, cela veut dire que chaque nœud du réseau exécute une copie logique indépendante de l'algorithme TORA pour chaque nœud destination [PAR 97].

3.4.4 Associativity Based Routing (ABR)et Signal Stability routing Adaptive(SSA)

Le choix des routes dans ABR et SSA se fait en se basant sur les routes qui ont plus longue durée de vie au contraire de certains protocoles qui utilisent le plus court chemin comme métrique de sélection. Dans les deux protocoles chaque nœud du réseau échange périodiquement des *beacons*(message de contrôle) afin de s'identifier et de montrer leur existence aux autres nœuds.
Dans ABR, la stabilité est enregistrée sous forme "état d'associativité", une valeur d'associativité élevée avec un nœud indique un état bas de mobilité du nœud, alors qu'une basse valeur d'associativité peut indiquer un état élevé de mobilité.
Afin d'établir un chemin entre une source et une destination, la source diffuse un paquet BQ muni d'un numéro de séquence. Chaque nœud intermédiaire traite le BQ reçu et vérifier si le BQ a atteint la destination, si ce n'est pas le cas le nœud rediffuse le BQ et rajoute son adresse et ses intervalles d'associativités au paquet de la requête BQ. De cette manière, chaque paquet qui arrive à la destination, va contenir les intervalles d'as-

sociativité des nœuds qui appartiennent au chemin reliant la source et la destination. Le nœud destination peut donc reçoit plus d'un BQ, le choix de meilleur chemin peut de faire en examinant les intervalles d'associativité qui existent dans chaque chemin, parfois plusieurs chemins ont le même degré de stabilité d'association, le chemin ayant le plus petit nombre de nœud (i.e. plus court chemin) est choisi. Ensuite le nœud destination envoie un paquet de réponse (REPLY), au nœud source en utilisant le chemin choisi. Les nœuds qui appartiennent au chemin suivi par le paquet REPLY marquent que leurs routes sont valides, le reste des routes reste inactif.

Dans SSA, la stabilité de lien aux voisins est mesurée en classifiant les voisins comme fortement connecté (SC) ou faiblement connecté (WC). Quand une source veut envoyer un paquet à la destination et n'a aucune route valide, une requête de recherche de route (Route Search) est envoyée par un nœud voisin si et seulement si : (1) Les paquets sont reçus à partir d'un canal fort et (2) si la requête de route n'a pas été propagée précédemment(pour éviter de former de boucle). Ainsi, seulement des routes fiables seront découvertes. Pendant la propagation, l'adresse du nœud est ajoutée à la requête de route RS. Ainsi, quand le message atteint la destination, elle contient les adresses des nœuds intermédiaires composés de liens forts. La destination choisit le premier paquet de requête de route arrivé pour le renvoyer car il est fortement probable que le paquet est arrivé via le chemin le plus court et/ou le moins encombré. Ainsi, quand la source reçoit une réponse de la destination, une route stable est établie. Les paquets de recherche de routes qui arrivent à la destination prennent nécessairement le chemin de forte stabilité de signal car les nœuds intermédiaires n'envoient pas de paquets à travers les liens de faible puissance de signal. Si la source expire son timeout avant la réception d'une réponse, elle relance de nouveau un processus de recherche de routes en indiquant cette fois ci que les canaux de faibles puissances peuvent être utilisés.

ABR utilise un processus de requête de localisation (LQ[h]) pour déterminer si un nœud voisin, de rang h dans le chemin source/destination, peut toujours être atteint ou pas. Si la destination reçoit le paquet LQ, elle sélectionne le meilleur chemin partiel existant et l'envoie dans un paquet de réponse REPLY. Dans le cas contraire, le nœud qui a initié le processus de localisation attend jusqu'à l'expiration de son timeout, et relance par la suite le même processus pour le voisin suivant. Si le processus de localisation LQ échoue pour tous les voisins pour un certain nombre de fois, le lien est considéré comme invalide et la source initie un nouveau processus BQ.

La défaillance d'un lien est détectée si le beacons d'un voisin n'est pas reçu pendent une certaine période. En premier lieu réagisse en essayant de trouver une route plus courte que la précédente sinon une nouvelle découverte de route est initiée.
Dans SSA, le nœud détectant la panne envoie un paquet d'erreur à la source. cette dernière enverra un paquet d'effacement pour informer tous les nœuds du lien déconnecté et initie une nouvelle découverte de route si nécessaire[UNE 02][TOH 97][DUB 96].

3.4.5 Location-Aided Routing (LAR)

Le protocole de *routage aidé par la localisation* est un protocole de routage réactif basé sur l'utilisation des localisations. Ce protocole procède d'une manière très similaire au protocole DSR, la principale différence entre les deux protocoles résidant dans le fait que LAR utilise les informations des localisations fournies par le système de positionnement global (GPS : Global Positioning System) dans le but de limiter l'inondation des paquets de requête. Afin d'assurer cela, deux approches peuvent être utilisées :
Dans la *première approche*, le nœud source définie une région circulaire dans laquelle la destination peut être localisée(voir la figure 3.6(a)). La position et la taille de la région, sont estimées en se basant sur : La position de la destination, telle qu'elle est connue par la source, l'instant qui correspond à cette position et la vitesse moyenne du mouvement de la destination.
Le plus petit rectangle couvrant la région circulaire et le nœud source est appelé la zone de requête(Resquest Zone). L'information calculée est rattachée au paquet de requête de route. Cela est fait uniquement par le nœud source et les nœuds qui appartiennent à la zone de requête. De cette manière chaque nœud qui reçoit ce paquet peut déterminer s'il est dans la zone de requête, dans ce cas expédie le paquet sinon il le rediffuse.
Dans la *deuxième approche*, la zone de requête n'est pas indiquée explicitement par le paquet de requête de route, donc le nœud source calcule la distance qui le sépare de la destination et calcule les coordonnées du mobile, et inclue ces informations dans le paquet de requête de route. Ce dernier est envoyé par la suite aux nœuds voisins(voir la figure 3.6(b)). Quand un nœud reçoit le paquet de requête, 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 envoie le paquet reçu. Lors de l'envoi, le nœud met à jour le champ de distance avec sa propre distance qui le sépare du nœud destination[KO].

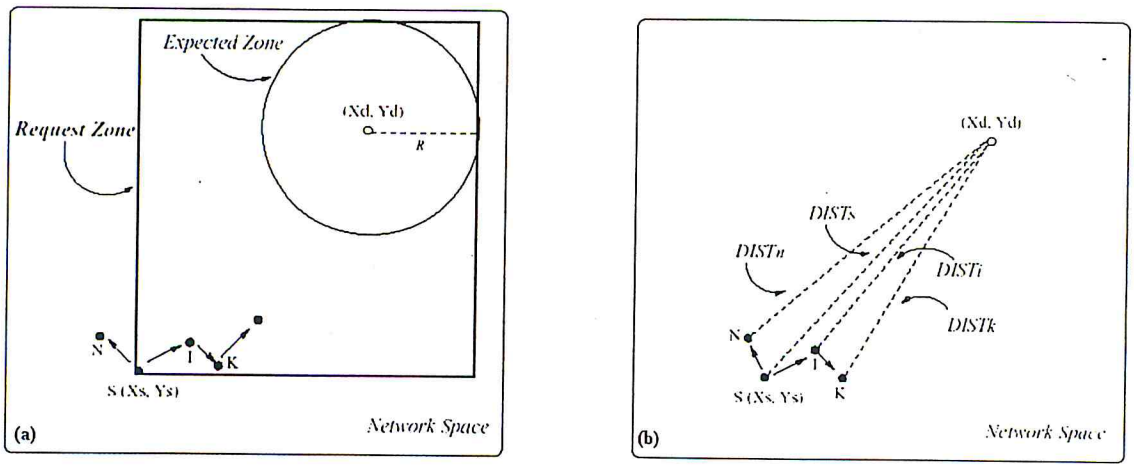


FIG. 3.6 – Les deux approches de LAR.

Dans les deux méthodes, si aucune réponse de route n'est reçue en dépassant une certaine période (timeout), le nœud source rediffuse une nouvelle requête de route en utilisant cette fois une diffusion pure (sans limitation). Autrement dit l'espace de réseau entier est inclus dans le prochain paquet de requête de route.

3.5 Inconvénient des protocoles de routage réactifs

L'inconvénient de ce type de protocole est que le nombre de paquets de contrôle n'est pas borné, car les demandes de routes se propagent dans tout le réseau, même si les nœuds entre lesquels il faut construire une route sont proches en nombre de sauts. Cependant, puisqu'une route doit être entièrement découverte avant la transmission réelle de paquet de données, le délai d'attente initial de recherche de route peut dégrader la performance des applications interactives (par exemple : requêtes de base de données répartie). D'ailleurs, il est impossible de savoir à l'avance la qualité du chemin (par exemple, la largeur de la bande passante, délai de transfert, charge du réseau, etc.) avant le commencement du transfert. Une telle connaissance a priori est très souhaitable dans des applications multimédias, puisqu'elle permet un contrôle plus pertinent d'acceptation d'appel. Si la route est devenue défaillante à cause de la mobilité, un paquet peut avoir besoin des découvertes multiples de route pour une destination. Le cache de routes peut devenir inefficace dans le cas de mobilité élevée. Puisque la diffusion "flooding" est utilisée pour la transmission de requêtes et la maintenance de routes, le routage réactif tend à devenir inefficace quand la charge et la mobilité de la circulation sont hautes et la taille

du réseau devient grand.

3.6 Les protocoles de routage hybrides

Comme nous l'avons expliqué ci-dessus, une approche purement proactive ou purement réactive pour mettre en œuvre (en application) un protocole de routage pour un réseau mobile Ad hoc ont leurs inconvénients. Les protocoles de routage hybrides sont conçus pour combiner les avantages des protocoles proactifs et réactifs. Dans un modèle hybride, en suit, dans la découverte des routes, une approche proactive si la cible est à moins d'un nombre fixe de sauts, et une approche réactive au-delà. Le protocole hybride le plus connu est ZRP.

3.6.1 Zone Routing Protocol(ZRP)

Le *protocole de routage de zone*, est un modèle hybride qui combine l'approche proactive et l'approche réactive. ZRP définit pour chaque nœud une zone de routage qui incluse tous les nœuds dont la distance minimum à ce nœud est R. Les nœuds qui à une distance R sont appelés des nœuds périphériques. Pour trouver une route vers des nœuds qui sont à une distance supérieure à R, ZRP utilise un protocole réactif qui envoie une requête à tous ses nœuds périphériques. ZRP se base sur deux protocoles de routage : IARP (IntrA zone Routing Protocol) et IERP (IntEr zone Routing Protocol) . IARP donne suivant une approche proactive toutes les routes jusqu'à une distance R. IARP est un protocole de type *vecteur de distance*. IERP donne de façon réactive les routes pour des destinations qui seraient à plus de R sauts.
Durant la construction des routes, si le destinataire se trouve dans la zone de source. Le chemin vers la destination est déjà connu et aucun processus d'établissement de connexion n'est nécessaire, sinon la source diffue à l'aide d'IERP une requête de route à tous les nœuds périphériques. Si un nœud périphérique connaît une route vers la destination, une réponse de route est envoyée en retour à la source indiquant le chemin à emprunter pour atteindre la destination, sinon le processus se poursuit récursivement jusqu'à l'obtention d'une route.
IERP dispose également d'un mécanisme de réponse réactive aux erreurs de route. Une erreur de route est générée lorsque le prochain saut est déterminé comme inaccessible (c'est-à-dire qu'il n'apparaît plus dans la table de routage intrazone). A partir d'une

erreur de route, IERP est alerté et un paquet d'erreur de route est généré. Ce paquet se propage en direction de la source comme un paquet de réponse de route. Lorsqu'une notification d'erreur est émise, le lien de communication ayant expirée est retiré de la table de routage interzone[HAA 97].

3.7 Evaluation qualitative des protocoles

Avec un protocole proactif, les routes sont disponibles immédiatement. Cependant, le trafic induit par les messages de contrôle et de mise à jour des tables de routage peut être important et partiellement inutile. De plus, la taille des tables de routage croit linéairement en fonction du nombre de nœuds. A l'opposé, dans le cas d'un protocole réactif, aucun message de contrôle ne charge le réseau pour des routes inutilisées. Mais, pour ces derniers, la mise en place d'une route par inondation peut être coûteuse et provoquer des délais importants avant l'ouverture de la session[SCH 02].
En termes de performances, les protocoles d'état de lien convergent plus rapidement que les protocoles de vecteur de distance. Cependant, dans le cas de réseaux à forte mobilité, le trafic induit par les fréquents messages de contrôle est souvent pénalisant. D'une façon générale, les protocoles de routage proactifs, qu'ils soient de vecteur de distance ou d'état de lien, ne se sont pas adaptés aux réseaux de taille importante et à forte mobilité. Une première solution pour ce type de réseau est l'utilisation de protocoles dits hiérarchiques (par exemple : HSR). Une seconde solution peut être d'utiliser un protocole réactif. Ce type de routage permet de gérer des réseaux de taille importante si la mobilité des nœuds reste faible ; le trafic reste faible s'il est dirigé vers un nombre réduit de destinations. D'autre part, le calcul d'une route à la demande est très pénalisant pour du trafic multimédia demandant des garanties en matière de qualité de service.
Des études comparatives montrent que certains protocoles sont plus performants que d'autres selon les caractéristiques du réseau. Par exemple, AODV, DSR ou GSR sembleraient convenir à des réseaux à forte mobilité, tandis que TORA est mieux adapté à des réseaux plus statiques. Par ailleurs, TORA gère efficacement les grands réseaux à forte densité, AODV semble quant à lui performant dans les réseaux de faible densité, et DSR est bien adapté aux petits réseaux.
A l'heure actuelle, aucun protocole de routage dans les réseaux Ad hoc n'a été standarisé au sein de l'IETF. Les différents protocoles sont présentés dans le tableau ci-dessous ne sont encore que des drafts (brouillant) et restent en cours de développement et/ou

	Sans boucles	Routes multiple	distribué	Support la QoS	sécurité	Conservation d'énergie	Support des Liens unidirectionnels	Fiabilité	Diffusion périodique
DSDV	✓	×	✓	×	×	×	✓	×	✓
WRP	✓	×	✓	×	×	×	✓	×	✓
GSR	✓	×	✓	×	×	×	✓	×	✓
FSR	✓	×	✓	×	×	×	✓	×	✓
CGSR	✓	×	✓	×	×	×	✓	×	✓
HSR	✓	×	✓	×	×	×	✓	×	✓
ZHLS	✓	×	✓	×	×	×	✓	×	✓
DREAM	✓	×	✓	×	×	×	✓	×	×
AODV	✓	×	✓	×	×	×	×	×	✓
DSR	✓	✓	✓	×	×	×	✓	×	×
TORA	✓	✓	✓	×	×	×	✓	×	✓
LMR	✓	✓	✓	×	×	×	✓	×	✓
ABR	✓	×	✓	×	×	✓	✓	×	✓
SSA	✓	×	✓	×	×	✓	✓	×	✓
LAR	✓	×	✓	×	×	×	✓	×	×
ZRP	✓	×	✓	×	×	×	✓	×	✓

Tel que:

✓ : Signifie que le protocole supporte cette propriété.

× : Signifie que le protocole ne supporte pas cette propriété.

TAB. 3.1 – Comparaison entre quelques protocoles de routage des réseaux Ad hoc.

de spécification. Certaines propositions ont été abandonnées et les plus abouties sont AODV, DSR. Ces derniers sont parmi les plus anciens et ils ont fait l'objet de simulations comparatives détaillées.

Jusqu'à présent, les protocoles sont analysés théoriquement(notamment par l'outils de simulation). Le tableau 3.1 récapitule et compare les protocoles présentés auparavant. Comme nous pouvons le voir selon le tableau TAB[2] 3.1, aucun protocole ne conserve l'énergie de sa batterie(les batteries des nœuds auxquels ils opèrent) et la qualité de service. Tous les protocoles sont distribués, donc il n'existe aucun protocole qui dépend d'un nœud centralisé et donc il peut réagi en cas de changement de topologie.

3.8 Conclusion

Dans ce chapitre nous avons présenté les protocoles de routage les plus connus appartenant à la famille des protocoles non prédictible proposés pour les réseaux Ad hoc. Ces protocoles peuvent être catégorisés en trois catégories : proactifs, réactif et hybrides,

[2]Une grande partie de ce tableau a été inespérée de [LAR 98].

selon quand et comment les routes sont découvertes. Dans les protocoles proactifs, l'information de routage vers tous les noeuds est maintenue par chaque nœud alors que pour les protocoles à la demande, les routes ne sont crées que lorsqu'elles sont désirées par un nœuds source et on a vu la troisième catégorie des protocoles, est une hybridation entre l'approche proactive et réactive.

Cependant, à travers notre étude des différents protocoles de routage existants, nous avons vu que ces protocoles utilisent une variété de techniques afin de résoudre le problème de routage dans l'environnement des réseaux Ad hoc. Parmi les techniques exploitées : le principe des numéros de séquence, le concept du groupe (hiérarchie), la technique d'œil de poisson "fisheye", la technique de routage de source, le principe d'inversement de liens, le principe de localisation, etc.

Cette variété de techniques fait que l'utilisation des solutions de routage qui existent dépend de la nature de l'application, par exemple dans un réseau Ad hoc muni d'un système GPS il est préférable d'utiliser un protocole de routage basé sur la localisation comme le protocole LAR ou DREAM et ainsi de suit.

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

Les protocoles de routages étudiés jusqu'ici, ne réagissent aux changements d'état des liens, qu'au moment de la défaillance de ces derniers, ce qui implique une grande perte de données avant qu'une autre route soit établie. Une autre famille de protocoles de routage, à cet égard est proposée, de telle sorte à réagir plus rapidement aux changement d'état des liens ou même avant qu'un lien change son état, en utilisant pour cela les techniques de prédiction, cette famille est la famille des protocoles de routage prédictifs, ce que nous allons le voir dans le chapitre suivant.

Chapitre 4

Protocoles de routage prédictifs

4.1 Introduction

Les réseaux mobiles Ad hoc permettent un modèle plus flexible de transmission que les réseaux traditionnels, parce que l'utilisateur n'est pas limité à un emplacement physique fixe. À la différence des réseaux sans fil cellulaires, un réseau sans fil Ad hoc n'a aucune infrastructure fixe de transmission. Pour une connexion active, l'hôte destinataire aussi bien que les nœuds intermédiaires peuvent être mobiles. Par conséquent les routes sont sujettes à des déconnexions fréquentes. Dans un tel environnement, il est important de réduire au minimum les effets des ruptures provoquées par la topologie changeante pour l'application critique telle que la voix et la vidéo. Ceci présente un défi difficile pour des protocoles de routage, puisque la reconstruction rapide des routes est cruciale en présence des changements de la topologie et elle est coûteuse en termes de temps et de messages de contrôle. En exploitant des comportements non-aléatoires pour les configurations de mobilité, nous pouvant prévoir le futur état de la topologie du réseau, et exécuter la reconstruction des routes d'une manière proactive.

D'ailleurs, en utilisant l'information prévue pour la topologie de réseau, nous pouvons éliminer des transmissions fréquentes des paquets de contrôle autrement requis pour reconstruire la route et pour réduire ainsi la charge du réseau.

Les protocoles de routage prédictifs sont aussi regroupés en des protocoles de routage proactifs et réactifs mais, il y a un autre facteur qui rentre dans le jeu, c'est la prédiction de mobilité.

Dans ce qui suit, nous allons présenter le fonctionnement des protocoles de routage appartement à la famille des protocoles de routage prédictifs les plus connus proposés

جامعة البليدة - المكتبة - كلية العلوم

pour assurer le routage de données dans les réseaux mobiles Ad hoc, Nous allons présenter mécanismes de prédiction également utilisés pour chaque protocole.

4.2 Les mécanismes de prédiction

Des algorithmes de prédiction de mobilité ont été précédemment utilisés dans deux protocoles de routage, LAR et DREAM(vus précédemment). Mais le mécanisme utilisé est très faible car, LAR et DREAM ne prennent pas en considération la direction des unités mobiles, ils essayent de prévoir l'emplacement futur à un instant bien déterminé. Dans le protocole DREAM, les intervalles de mise à jour dépendent seulement de la vitesse du mouvement du nœud. Dans les méthodes que nous allons les présenter prennent en compte les deux paramètres : la vitesse et la direction du mouvement des unités mobiles pour prédire le temps qui reste entre deux nœuds i.e. Link Expiration Time(LET). Le LET entre deux nœuds mobiles est basé sur le principe que deux voisins dans le mouvement pourront prévoir le futur temps de déconnexion. Il y a deux méthodes qui peuvent faire une telle prédiction.

Dans la *première méthode*, nous supposons que la puissance du signal reçu d'un nœud mobile dépend seulement de la distance qui le sépare de l'émetteur. En utilisant les paramètres de mouvement de deux voisins, nous pouvons déterminer la durée que deux nœuds mobiles resteront connectés. Ces paramètres peuvent être obtenus en utilisant le système de positionnement global (GPS).

Dans la *deuxième méthode*, on utilise les mesures des puissances de signaux. Ces derniers sont mesurés périodiquement à partir des paquets reçus des voisins d'un nœud mobile. En utilisant cette information, le nœud peut calculer le taux du changement de la puissance du signal d'un voisin particulier. Par conséquent, la durée de vie d'un lien peut être calculée.

4.3 Méthode basée sur les informations de GPS

Dans cette section on va expliquer la méthode de prédiction utilisant les informations de localisation ainsi que les informations de mobilité fournies par le système de positionnement global (Global Positioning System) pour prédire le temps restant(Route Expiration Time) entre deux nœuds (source/destination) avant qu'ils deviennent déconnectés.

4.3.1 Mécanisme de base

La méthode de prédiction de la disponibilité de lien est proposée dans[SU 99][SU 01]. Elle suppose que si la distance entre deux nœuds est moins que le rayon de transmission, ils peuvent se communiquer directement.

On suppose un modèle de propagation des signaux dans un espace libre(pas d'obstacle), où le signal reçu dépend seulement de sa distance de l'émetteur. On suppose également que tous les nœuds dans le réseau se référencent au même horloge (par exemple : Network Time Protocol ou l'horloge de GPS elle-même). Par conséquent, si on connaît les paramètres de mouvement, i.e. la vitesse, la direction, l'intervalle de transmission radio entre deux nœuds, nous pourrons estimer la durée de temps pour laquelle ces deux nœuds resteront connectés.

Supposons deux nœuds I et J l'un se trouve dans la portée de communication de l'autre. Le nœud I a comme coordonnées (x_I, y_I) et déplace avec une vitesse v_I suivant la direction de déplacement θ_I. de même, les coordonnées de nœud J sont (x_J, y_J) qui déplace avec une vitesse v_J suivant la direction θ_J $(0 \leq \theta_I, \theta_J < 2\pi)$. La vitesse, la direction et l'emplacement des nœuds sont fournis par le GPS. Par conséquent, la durée de temps

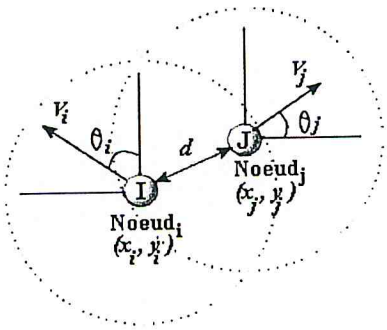


FIG. 4.1 – Paramètres de mouvement.

pour que ces deux nœuds restent connectés est donnée par :

$$T = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2 - (ad-bc)^2}}{a^2+c^2} \tag{4.1}$$

avec :

$$a = v_I \cos\theta_I - v_J \cos\theta_J \ , b = x_I - x_J, \ c = v_I \sin\theta_I - v_J \sin\theta_J$$

4.4 Application

Dans ce qui suit, nous décrivons les deux protocoles de routage unicast qui utilisent le mécanisme de prédiction de mobilité expliqué dans la section précédente. Le premier protocole (FORP) utilise une approche réactive tandis que l'autre est un protocole de routage proactif basé sur l'algorithme de vecteur de distance.

4.4.1 Flow Oriented Routing Protocol

FORP est un autre protocole réactif utilise la méthode de prédiction expliquée précédemment. Seulement les routes actives sont maintenues.
Quand une source a un flux de données à transmettre elle construit ré-activement une route vers la destination et injecte le flux. La destination prévoit le changement de la topologie à l'avance via les informations de mobilité contenues dans les paquets de données. On suppose qu'un nœud mobile peut prévoir le temps de déconnexion de lien de tous ses voisins on utilisant la méthode de prédiction décrite précédemment. Le concept de base d'amélioration est que, si nous pouvons prévoir le temps d'expiration de lien, LET(Link Expiration Time), au long de chaque saut sur la route, nous pouvons prévoir le temps d'expiration de route RET (Route Expiration Time). Le RET est le minimum des LET_i composant une route, la durée de vie de la route la plus stable, RET, est égale au maximum des RET_i.
Quand une source a un flux de données a envoyé vers une destination, elle va d'abord vérifier son cache de route et voir si elle a une route non expirée à la destination, si c'est le cas elle enverra le paquet de données en utilisant la route, sinon, elle diffuse un paquet de requête FLOW-REQ, La figure 4.2(a) illustre ce principe. La source (nœud A) diffuse un message Flow-REQ dans le réseau, les nœuds intermédiaires B, C, D et E retransmettent le Flow-REQ en ajoutant dans le message Flow-REQ leur ID et le LET du lien à partir duquel la requête est reçue. Finalement (selon cet exemple), la destination (nœud F) reçoit deux messages Flow-REQ l'un contient la route $[A, B, C, E, F]$ avec des LETs=$(4, 4, 3, 6)$ et l'autre contient la route $[A, B, D, E, F]$ avec des LETs=$(4, 5, 4, 6)$, comme le RET(Route Expiration Time) est le minimum entre les LETs de chaque route et le maximum entre les RET_i, la destination choisie donc la première route qui a $RET = 5$ car, elle est plus stable que la deuxième route. La destination utilise la route choisie pour le Flow-SETUP (réponse de route), la figure 4.2(b) illustre ce processus. Durant le transfert des données, les nœuds intermédiaires ajoutent les LETs prévus à chaque

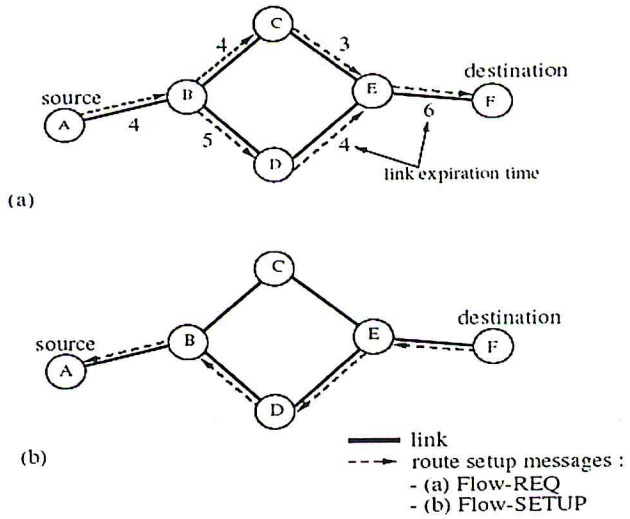


FIG. 4.2 – Le processus d'installation de flux :(a) la propagation des messages Flow-REQ sur leur chemin au nœud destinataire F ; (b) les nœuds intermédiaires installent leurs états de flux quand ils reçoivent le message Flow-SETUP.

paquet de données. La destination continue de recevoir les RETs prévus contenus dans chaque paquet de données. Quand la destination détermine que la route utilisée est sur le point d'être d'expirée un message Flow-HANDOFF est généré par la destination et diffusé de la même manière que le message Flow-REQ. Après la réception de ce message par la source, cette dernière peut déterminer la meilleure route en se basant sur les LETs contenus dans le message Flow-HANDOFF, donc la source envoie un message Flow-SETUP suivant la nouvelle route. La figure 4.3(a) montre le processus de "Handoff", l'ancienne route est selon les nœuds $[A, B, C, E, F]$ est sur le point d'être expirée. Dans la figure 4.3(b) juste avant l'expiration de lien et après la fin de processus de HandOff, la nouvelle route maintenant sera $[A, B, D, E, F]$.

4.4.2 Distance Vector with Mobility Prediction(DV-MP)

Ces protocoles utilisent le RET (Route Expiration Time) dans la table de routage comme métrique pour la sélection des routes où sur lesquelles les paquets de données seront envoyés. la plus grande valeur de RET représente la route la plus stable, l'utilisation des RETs permet d'éliminer les mises à jours événementiels utilisées par le protocole DSDV, puisque les routes utilisées sont établies en se basant sur la stabilité, cela signifie que l'intervalle de mise à jour peut être étendue et les mises à jour fréquentes ne sont

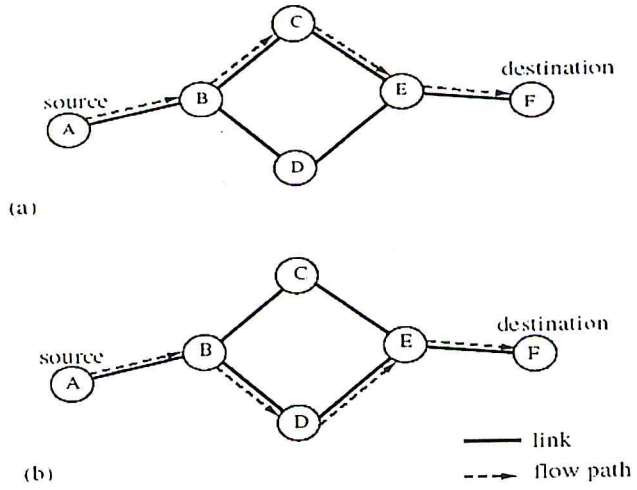


FIG. 4.3 – Le processus de handoff : (a) la route initiale avant le processus de handoff; (b) la nouvelle route après la terminaison de processus de handoff.

plus exigées. En outre, l'utilisation des routes les plus stables réduit considérablement les déconnexions provoquées par la mobilité.

Ce protocole exige un échange périodique des tables de routage pour chaque nœud, à la différence de l'approche réactive, aucun temps d'attente de route n'est nécessaire pour le transfert puisque les routes sont construites à l'avance et la charge du réseau reste constant indépendamment du nombre de transmetteurs.

Dans cette amélioration, pour utiliser l'information de prédiction (c à d. LETs et RETs), un champ supplémentaire doit être ajouté au niveau de la table de routage. Par conséquent, la métrique RET est ajoutée à la table pour chaque entrée. Chaque nœud diffuse périodiquemc la table de routage et associer a chaque table un numéro de séquence, ce dernier est incrémenté après chaque annonce d'une table de routage.

D'après la figure 4.4(a), Le nœud A reçoit une table de routage à partir de son voisin B, le vecteur de mobilité contenu dans la table de routage est utilisé par A pour calculer le LET entre le nœud A et le nœud B. Le LET et la table de routage reçus du nœud B est alors utilisés pour mettre à jour la table de routage de nœud A. un nœud mettre à jour sa table si et seulement :

① Si l'entrée reçue pour la destination D a un meilleur RET, et le numéro de séquence reçu est supérieur ou égal à l'ancien numéro associé à l'entrée D, l'entrée du nœud A pour la destination D est mis à jour ;

② Si l'entrée pour la destination D a un numéro de séquence plus élevé, l'entrée du nœud A pour la destination D est mis à jour.

Le processus est expliqué dans la figure 4.4(b). En résumé, toutes les méthodes décrites

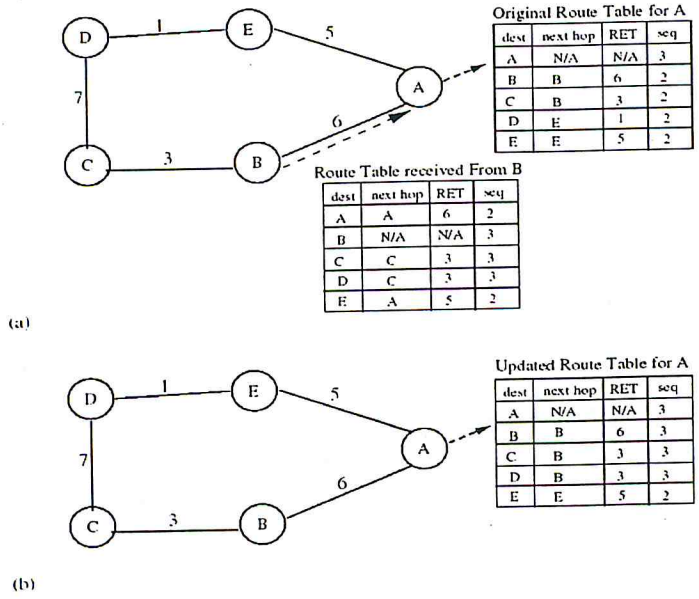


FIG. 4.4 – Exemple d'une mise à jour de la table de routage.

précédemment supposent que les nœuds se déplacent avec des vitesses constantes et des directions constantes pendant la prédiction. Quand la distance des nœuds est obtenue en mesurant les puissances de signaux reçus, la puissance de transmission est également supposée constante.

4.4.3 Inconvénient de ce protocole :

En raison des limitations de GPS(le coût, la panne, le non-fonctionnement dans certaines situations, exactitudes des informations fournies, etc.), une autre méthode a été proposée. Elle se base sur le calcul de la puissance du signal pour prédire le temps de connexion restant entre les deux nœuds.

Plusieurs modèles de propagation radio ont été proposés pour calculer la puissance du signal reçu, le modèle le plus populaire est le modèle de propagation dans un espace où les obstacles n'existant pas(free space propagation model). Fondamentalement, la puissance de la transmission du nœud est constante. Des échantillons des puissances de signaux

reçus sont mesurés à l'aide des paquets reçus à partir des des nœuds voisins. Ainsi, il est possible de calculer le taux du changement pour le niveau de puissance du signal d'un voisin particulier et nous pouvons prévoir quand le niveau de puissance du signal reçu descend au-dessous d'un certain seuil.

4.5 Méthode d'affinité

L'affinité A_{nm} liée à un lien l_{nm}, est l'estimation de la durée de vie de ce lien, pour simplifier les choses nous supposons que l'affinité A_{nm} égale à A_{mn} et les portées des communications pour tous les nœuds sont égales à **R**. Pour calculer l'affinité A_{nm}, le nœud **n** envoie périodiquement des beacons(messages de contrôle) et le nœud **m** mesure périodiquement la puissance des signaux reçus du nœud **n**. Le taux du changement de puissance du signal est donné par :

$$\Delta S_{nm} = \frac{S_{nm(courante)} - S_{nm(precedente)}}{\Delta(t)} \tag{4.2}$$

Où :
$\Delta(t)$ est le temps écoulé entre $S_{nm(courante)}$ et $S_{nm(precedente)}$, ceci nous donne la moyenne sur les derniers échantillons pour obtenir $\Delta S_{nm(avg)}$. L'affinité est maintenant calculée comme :

$$A_{nm} = \begin{cases} elevee & \text{si } \Delta S_{nm(Avg)} > 0 \\ \frac{S_{thresh} - S_{nm(courante)}}{\Delta S_{nm(Avg)}} & \text{sinon} \end{cases} \tag{4.3}$$

Où :
le S_{thresh} est le seuil de puissance. Si la puissance du signal reçu descend au dessous de ce seuil, on considère que qu'il y a une déconnexion entre les deux nœuds **n** et **m**.

4.5.1 Route-lifetime Assessment Based Routing protocol (RABR)

Le protocole de routage basé sur l'estimation de la duré de vie d'une route, est un protocole de routage réactif utilise la technique d'affinité pour prédire le temps d'expiration de lien.
Durant la transmission de la requête, les nœuds intermédiaires ajoutent des valeurs successives d'affinité dans le paquet de requête, la destination va examiner l'affinité minimum, A_{min}, de tous les paquets qui sont atteints la destination. La route qui a la plus grande valeur d'affinité A_{min} est finalement choisie par la destination.

Quand une source a un certain nombre de données à transmettre, elle va initier une *requête de route* et la diffuse dans tout le réseau. Chaque paquet *requête de route* a une <source, destination, numéro de séquence>, chaque nœud intermédiaire rediffuse la requête de route au voisin après avoir ajouter l'affinité de lien de nœud duquel il a reçu le paquet de requête de route. Afin d'éviter le bouclage, chaque nœud ne traite jamais un paquet qui a la même < source, numéro de séquence > au plus d'une fois, ce processus se répète jusqu'à ce que le paquet atteigne sa destination. Après avoir reçu le paquet de requête de route par la destination, cette dernière attend durant un intervalle de temps. Après ce temps la destination aura plusieurs routes ayant des affinités déférentes.
La destination détermine l'affinité de lien la plus faible A_{min} de chaque route. A_{min} définie (désigne) la durée de vie de la route. La route qui a la plus grande valeur de A_{min} est finalement choisie pour la transmission de données. Après la sélection appropriée de la route par la destination, cette dernière envoie une réponse de route vers la source[SIN 00].

4.6 Conclusion

Ce chapitre a présenté la deuxième famille des protocoles de routage, les protocoles de routage prédictif. Nous avons vu que cette famille est divisée en deux modèles de fonctionnement : les protocoles de routage prédictifs réactifs, et les protocoles de routage prédictifs proactifs.
Dans ce chapitre nous avons présenté également, les différentes techniques de prédiction utilisées pour les protocoles de routage des réseaux Ad hoc. Comme nous avons vu, deux techniques peuvent être utilisées pour estimer la durée de vie d'un lien, l'une se base sur les informations de localisation fournies par le système GPS, tandis que l'autre utilisent les puissances des signaux émis par les nœuds voisins.

Chapitre 5

Conception et ipmlémentation du protocole MPRP

5.1 Introduction

Ce chapitre est divisé en deux parties principales, la première partie consiste à mettre en évidence certaines notions utilisées dans notre protocole : dans cette partie nous allons expliquer le principe des *routes multiples*, la technique de *routes disjointes*, les avantages qu'elle fournisse, expliquer également la notion de LET et RET.

La deuxième partie va expliquer d'une manière détaillée le principe de fonctionnement de notre protocole de routage proposé pour effectuer le service de routage dans les réseaux mobiles Ad hoc.

Première partie

Notions générales

5.2 Les Routes disjointes

Afin de réduire les réactions aux changements topologiques dues aux mobilités des nœuds et réduire au minimum la congestion (la charge) du réseau, des routes *disjointes multiples* à cet égard peuvent être utilisées : Dans la table de routageou le cache d'un nœud, quand une route devient invalide cela peut induire l'invalidité d'autres routes qui partagent les mêmes liens/nœuds que la route défaillante.

5.2.1 Définition

Étant données deux routes **R1** et **R2** sans circuit, on dit que **R1** et **R2** sont disjointes [LEU 01] si :

① **R1** et **R2** partagent le même nœud source et le même nœud destination et ;

② Il n'existe plus un nœud intermédiaire ou un lien en commun entre **R1** et **R2**.

Les routes disjointes dans un réseau sont en trois catégories :

Routes totalement disjointes (Nodes disjoints routes) Sont des routes qui ne partagent pas des nœuds ou des liens avec d'autres routes ;

Routes disjointes au termes de liens (Links disjoints routes) sont des routes qui ne partagent pas au moins un lien avec d'autres routes mais peuvent contenir des nœuds partagés avec d'autres routes.

Routes non disjointes (Non-disjoints routes) Sont des routes pouvant contenir des nœuds et des liens en commun avec d'autres routes.

Les trois catégories sont bien représentées dans la figure 5.1. Les routes multiples dis-

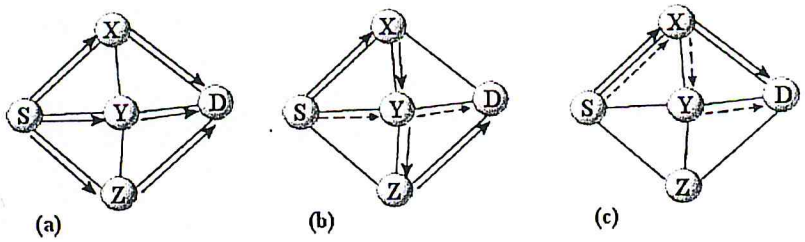


FIG. 5.1 – Routes [SXD], [SYD], et [SZD] dans(a) sont des nodes disjoints routes car elles ne possèdent ni un nœud ni lien en commun. Routes [SXYZD] et [SYD] dans (b) ont le nœud Y en commun donc sont des links disjoints routes. Routes [SXD] et [SXYD] dans (c) ont le nœud X et le lien [SX] en commun donc sont des non-disjoints routes.

jointes offrent certains avantages par rapport aux routes non disjointes, par exemple :

- **Tolérance aux fautes (fault tolerance)** Un système est dit tolérant aux fautes s'il continue de fonctionner même il y a des pannes dans une ou plusieurs parties du système.
  La figure 5.2 montre un réseau de cinq nœuds où le nœud source S a trouvé deux routes vers D, $[S, B, D]$ et $[S, A, C, D]$. Supposons que le lien entre le nœud S et B est devenu invalide à cause d'une mobilité ou d'une défaillance de lein inattendue de nœud B, les deux routes vers la destination seront perdues, ceci dans le cas où le principe des routes disjointes n'est pas utilisé. Or, la défaillance

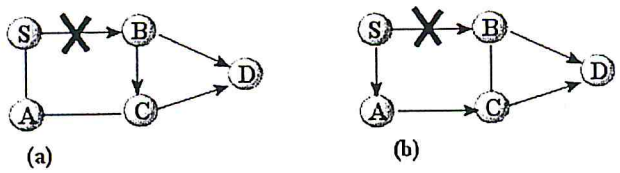


FIG. 5.2 – La possibilité d'utiliser d'autres routes alternatives quand le principe des routes disjointes est utilisé.

de lien en utilisant le principe des routes disjointes, la source a la possibilité de continuer la transmission des données sur la deuxième route. L'invalidité (la panne) d'un lien et/ou nœud dans le cas ou les routes disjointes sont utilisées, n'influe que sur une et une seule route à la fois. Or, ceci n'est pas vrai dans le cas où les routes utilisées dans le routage ne sont pas disjointes.

- **Équilibrage de la charge (Load Balancing)** L'une des méthodes pour éviter la surcharge du réseau est, l'utilisation des routes disjointes multiples, il se pourrait qu'une route soit trop chargée du aux utilisations multiples de ses nœuds par plusieurs nœuds source en même temps. Quand une route est devient trop chargée, la source peut utiliser une autre route alternative, de cette manière on réduit considérablement la charge du réseau et la probabilité que le réseau tombe dans un état de *goulot d'étranglement* est très petite. La figure 5.3 explique ce mécanisme, la source utilise deux routes en même temps pour transférer le flux de données du nœud A vers respectivement le nœud E et le nœud G. il vaut mieux d'utiliser une autre route pour distribuer la charge sur le réseau et d'éviter de charger le lien reliant A et B.

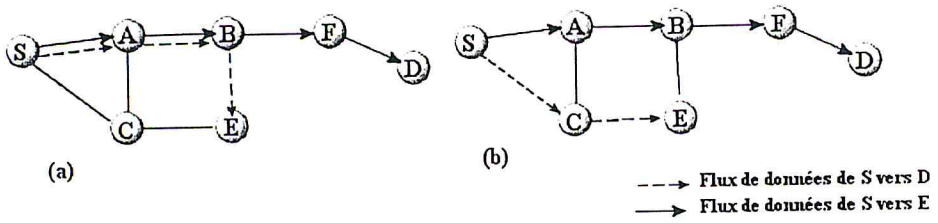


FIG. 5.3 – Distribution de la charge dans le réseaux.

5.2.2 Calcul de Link Expiration Time et Route Expiration Time

L'estimation de la durée de vie d'un lien, LET(Link Expiration Time), entre deux nœuds mobiles est basée sur le principe que deux voisins en mouvement pourront prévoir le futur temps de déconnexion. Comme nous l'avons déjà mentionné, deux méthodes peuvent être utiliser pour faire une telle estimation :

La méthode du GPS Dans cette méthode, la puissance du signal reçue d'un nœud mobile dépend seulement de sa distance à l'émetteur, et via l'utilisation des paramètres de mouvement de deux nœuds voisins (par exemple : la vitesse de déplacement, la distance entre les deux nœuds, etc.), nous pouvons estimer la durée que deux nœuds mobiles resteront connectés, ces paramètres sont obtenus en utilisant le GPS (Global Positioning System.).

Il y a énormément de protocole[1] qui se basent sur les informations fournies par le GPS. Puisque le GPS peut ne pas fonctionner correctement dans certaines situations [SU 99](par exemple, à l'intérieur des bâtiments, affaiblissement des signaux, etc.) nous pouvons toujours ne pas pouvoir prévoir exactement le LET.

La méthode d'affinité Dans cette méthode, on utilise des mesures de puissance : Des échantillons de puissance reçus sont mesurés périodiquement à partir des beacons reçus des nœuds voisins d'un nœud mobile. À partir de ces informations, on peut calculer(voir plus loin) le taux de changement pour le niveau de la puissance d'un voisin particulier. Par conséquent, le temps où le niveau de puissance descend au-dessous de la valeur acceptable(valeur seuil) peut alors être calculé.

L'utilisation de la méthode d'affinité offre certaines avantages par rapport à celle du GPS :

- L'utilisation de cette méthode coûte rien tandis que l'utilisation de GPS est payante.
- Le fonctionnement dans les situations où le GPS ne fonctionne pas.

[1] Par exemple, le protocole LAR.

Cependant, à partir des inconvénients du GPS cités précédemment et les avantages que fournie la méthode d'affinité, nous avons choisi et implémenter cette dernière pour estimer la durée de vie d'un lien entre un nœud et son voisin.

La durée de vie d'un lien $l_{n,m}$, reliant deux nœuds n et m, est calculée en se basant sur la méthode d'affinité présentée dans le chapitre 4 (l'équation 4.2 et 4.3). Si on peut estimer la durée de vie d'un lien LET_i reliant un nœud et son voisin, on peut estimer la durée de vie d'une route RET_i qui est égale au minimum entre les LET_i tout au long de la route, on peut formuler ça comme suit :
$RETi = Min\{LETi\}$.
Donc, la route la plus stable entre les routes reçues par le nœud S est déduit comme suit : $RET = Max\{RET_i\}$.
Expliquons ceci par l'exemple suivant : Dans la figure 5.4, le nœud S possède deux routes vers la destination D, la première route $[S, A, B, D]$ et l'autre $[S, C, F, G, D]$. Les durées de vie des liens sont représentées par les étiquettes des arcs du graphe. $RET_1 =$

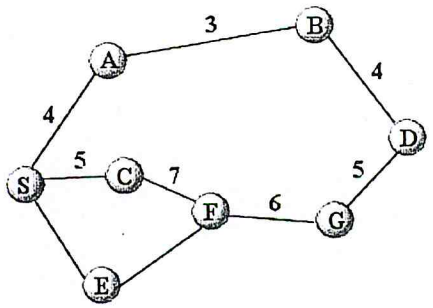


FIG. 5.4 –

$Min\{4, 3, 4\} = 3$ selon la première route.
$RET_2 = Min\{5, 7, 6, 5\} = 5$ selon la deuxième route.
Donc, la route la plus stable peut être déduite comme suit :
$RET = Max\{RET_1, RET_2\} = RET_2 = 5$.

5.2.3 Stabilité d'une route

Étant donnée une route **R**={i, j, k,..., y, z}, on note par $l_{n,m}$ chaque lien reliant deux nœuds voisins dans la route R.

Liens forts

On dit qu'un lien est en état fort si sa durée de vie, LET, est supérieure ou égale à une certaine valeur *seuil*(voir plus loin).

Liens faibles

On dit qu'un lien est en état faible si sa durée de vie, LET, est inférieure a une certaine valeur *seuil*.

Une route est dite stable si tous ses liens $l_{n,m}$ sont forts, elle dite instable si l'un de ses liens $l_{n,m}$ est en état faible.

L'automate suivant(figure 5.5) résume les trois états qu'un lien peut les prendre. L'état *défaillant* d'un lien, signifie que la communication entre un nœud et son voisin n'est plus existait.

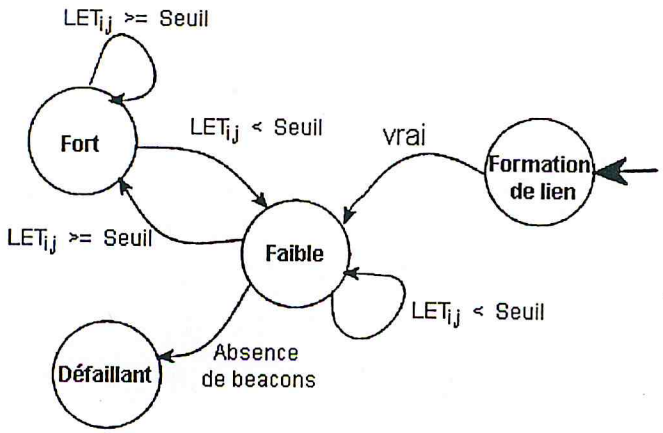


FIG. 5.5 – Les états possibles d'un lien dans le réseau.

5.2.4 Calcul du Seuil

Supposons qu'on a une route **R** entre le nœud source S et le nœud destination D, voir la figure 5.6) dont le nombre de saut de S vers D est égal à k sauts et le nombre de sauts de S vers le nœud I est égal à m sauts. Chaque nœud participant[2] dans le routage fait périodiquement certain test, ce test sert à vérifier si la durée de vie, $LET_{I,J}$, du lien reliant un nœud I et un autre nœud voisin J est suffisante pour transférer un paquet de données de S vers D. Si la durée est insuffisante, un paquet de notification est envoyé à

[2]I.e. il existe au moins un nœud source qu'il l'utilise pour le routage des paquets vers la destination.

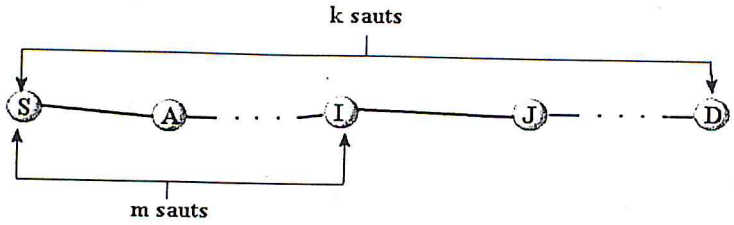


FIG. 5.6 – Calcul du seuil par un nœud intermédiaire.

la source par le nœud détectant le changement d'état.

Supposons qu'il est temps au nœud I d'effectuer la vérification, supposons également que le temps de transfert d'un paquet de donnée entre un nœud et son voisin égale à $LINK_TIME_i$.

Le nœud I doit notifier la source avant que cette dernière envoie un paquet de données et si la source à déjà envoyé un paquet il faut que le paquet ait le temps suffisant pour atteindre la destination. Le temps de notification doit être supérieur ou égal à un certain $seuil_i$. Le seuil est donné comme suit :

$Seuil_i = k * LINK_TIME_i + m * LINK_TIME_i + (m + k) * MPRP_BROADCAST_JITTER.$

Tel que :

k : Le nombre de saut(nœud)entre la source et la destination.

m : Le nombre de saut entre le nœud en question et la source.

MPRP_BROADCAST_JITTER : C'est le temps pris par un paquet pour qu'il se déplace entre deux nœuds.

Remarque : Le seuil est dynamique pour chaque nœud participant dans le routage. Si $LET_{i,j} < seuil_i$, le nœud I envoie une notification vers la source, sinon le temps est suffisant donc pas de notification (continuité d'utilisation de la route active). De cette manière, quand la source a reçoit un paquet de notification, elle met la route contenant le voisin qui a invoqué le changement d'état en attente et choisit une autre route alternative (voir plus loin les détails dans la présentation du prtocole).

Deuxième partie

Présentation du Multi-path Prediction-based Routing protocol (MPRP)

Cette partie va décrire le principe de fonctionnement détaillé de notre protocole proposé pour effecteur le service de routage dans les réseaux Ad hoc.

5.3 Multi-path Prediction-based Routing protocol (MPRP)

Le *protocole de routage multi-chemin basé sur la prédiction*, MPRP, est un protocole qui suit une approche réactive où les routes sont construites à la demande. MPRP est une combinaison d'idées entre les protocoles DSR, ABR et SSA : Il combine le mécanisme de *découverte de routes* et l'utilisation de la technique "routage de source" de DSR, l'utilisation des beacons (message de contrôles) de ABR, la distinction entre un lien fort et faible de SSA. En plus de ça MPRP utilise le principe de prédiction (vus précédemment) permet de maintenir les routes d'une manière proactive (prédire la déconnexion avant qu'elle est lieu).
Afin de router les paquets de données à leurs destinations et maintenir les routes, MPRP utilise deux mécanismes de base : la *découverte* des routes et la *maintenance des routes.*

5.3.1 La découverte des routes

Le protocole MPRP est basé sur l'utilisation de la technique "routage de source". Dans cette technique, la source des données détermine la séquence complète des nœuds à travers lesquels les paquets de données seront envoyés (voir les notions générales du capitre 2 section Routage de source).

Quand un nœud veut envoyer des données à un autre nœud dont la route vers ce dernier est connue, la source envoie directement les données en utilisant la route connue(sauvegardeé dans son cache).

En Principe, la source connaît la route vers la destination en consultant son cache des routes récemment reçues, si une telle route vers la destination désirée n'est pas connue, la source déclenche une procédure de découverte de routes en diffusant un paquet de *requête de route* RREQ(Route REQuest) dans tout le réseau. Chaque paquet RREQ contient les champs suivants[3] :

src : L'adresse du nœud initiateur du RREQ ;

dst : L'adresse du nœud destinataire vers lequel le paquet RREQ est destiné ;

[3]Le protocole manipule des paquets, chaque paquet doit contenir un champ *type_paquet* pour que le protocole puisse différencier entre les paquets. Juste pour simplifier les structures des paquets le champ *type_paquet*, et dans tous les structures qui viennent, ne sera pas représenté.

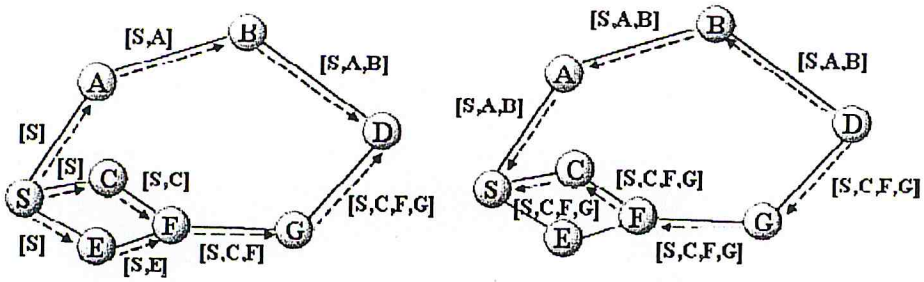
LET : La durée de vie du lien.

Parfois et à cause de l'indisponibilité des routes, la source ne reçoit pas une réponse. Si la réponse n'est pas reçue pendant une certaine période (MPRP_REQUEST_PERIOD), la source va rediffuser une nouvelle requête RREQ en incrémentant le numéro de séquence. Lors de la réception d'un paquet requête de route par un nœud P du réseau, le traitement suivant est effectué :

① Le nœud P ajoute son adresse dans le champ enregistrement de la route(champ *chemin* dans le paquet de requête de route RREQ) du paquet reçu et rediffuse le paquet ;

② Marquer qu'il a déjà reçu un paquet(src, NS).

De cette manière, la requête de route est propagée (figure 5.10(a)) dans le réseau jusqu'à ce qu'elle atteigne la destination qui va répondre la source par un paquet de *réponse de route* RREP, figure 5.10(b). Dans le but de retourner les paquets de réponse de



(a) La propagation de la requête de route (RREQ)

(b) La destination répond la source par deux Réponses de route(RREPs)

FIG. 5.10 – Le processus de découverte des routes dans le protocole MPRP.

route RREPs à la source (initiateur de la procédure de découverte), la destination doit connaître la route vers la source. Dans le cas où la destination n'a pas déjà gardé une telle route, la route spécifiée dans le champ *chemin* contenue dans le paquet requête de route, RREQ, peut être inversée et utilisée directement.

Pour chaque paquet de requête de route, RREQ, qui atteint la destination, cette dernière va répondre à la source par un paquet de réponse de route RREP. Chaque paquet réponse de route, contient les champs suivants : Où :

src : L'adresse de la source de la route ;

dest :L'adresse de la destination de la route ;

src	dest	RET	chemin

FIG. 5.11 – Format d'un paquet réponse de route dans MPRP.

RET : La durée de vie de la route ;

chemin : La séquence de nœuds prise par RREQ pour atteindre la destination.

Les nœuds intermédiaires et pendent la réception des réponses de route (RREP), extraient les routes vers la destination et les insèrent dans ses tables de routage si elles sont disjointes. La figure 5.12 suivante illustre ce mécanisme : Le nœud A peut extraire les

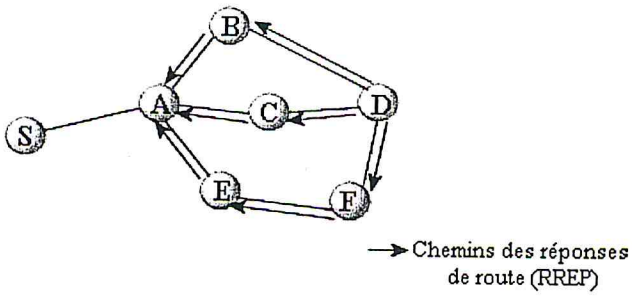


FIG. 5.12 – Le nœud A peut sauvegarder les trois routes vers la destination [B,D] ,[C,D] et [E,F,D].

routes vers la destination et les insère dans sa table de routage.
L'intérêt de ce comportement il se résume dans le point suivant :

- La continuité de transfert de données en cas d'en panne inattendue : Supposons que le lien entre A et B(figure 5.12) a tombé en panne, les paquets de données arrivant au nœud A de S et ne peuvent être transférer selon la route active(supposons $[S, A, B, D]$), A va choisir pour acheminer le paquet de données, la route $[C, D]$ ou $[E, F, D]$. Les routes qui ont une durée de vie plus grande sont favorisées par rapport à celles qui ont des petites durées de vie.

Quand le nœud source reçoit la première réponse de route RREP, il effectue le traitement suivant :

① Insère le chemin dans sa table de routage après avoir l'extrait du paquet reçu ;

② Envoie un paquet de données sur le chemin extrait.

Ces opérations se font tout en attendant l'arrivé d'autres réponses de routes.
Durant l'intervalle de temps entre la première réception du paquet réponse de route et la fin du délai de garde(timeout) déjà déclenché au moment d'initialisation de la procédure

de recherche de route, la source fait le suivant :

Elle applique l'algorithme de sélection des routes disjointes sur les paquets de réponse reçus. Si la nouvelle route reçue respect les conditions des routes disjointes la route est insérée dans la table de routage du nœud source selon la valeur RET incluse dans le paquet RREP(voir la figure 5.13) et envoie également un paquet de données sur la route nouvellement trouvées, sinon la route est ignorée.

Comme mentionné déjà, les routes reçues sont insérées directement dans la table de

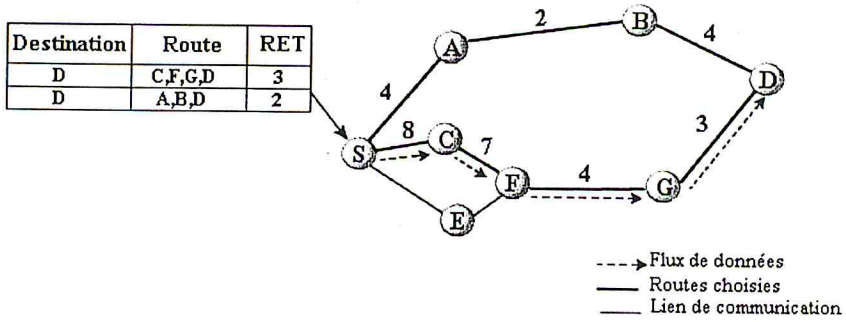


FIG. 5.13 – Les routes disjointes sélectionnées sont tries selon la valeur RET.

routage si elles sont disjointes, mais durant l'implémentation du MPRP, l'insertion directe a posée un grand problème tel que, les routes les moins stables seront insérées les premières, donc elles ne laissent pas la place (au sens route disjointes) aux routes les plus stables : La figure 5.14 montre un réseau Ad hoc comportant 5 nœuds.

Supposons que le nœud S a déjà initié une procédure de recherche de route et entrain

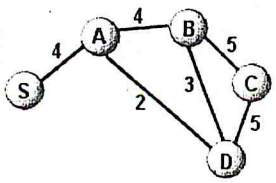


FIG. 5.14 –

maintenant de recevoir les réponses de routes(routes vers la destination), les routes reçues par S sont comme suit : $[A, D]$, $[A, B, D]$ et $[A, B, C, D]$ avec les durées des routes sont respectivement 2, 3 et 5.

La première route à insérer dans la table de routage est celle qui vient la première et qui a moins de (i.e. [A,D]), tandis que les deux autres routes ne seront pas insérées et seront

Destination	Route	RET
D	A,B,C,D	4
D	A,B,D	3
D	A,D	2

FIG. 5.15 – Toutes les routes reçues sont insérées dans la table intermédiaire.

Destination	Route	RET
D	A,B,C,D	4

FIG. 5.16 – Table de routage ne contient que les routes stables disjointes.

ignorées par ce qu'elles possèdent un nœud en commun avec la route déjà insérée donc, elles ne respectent pas les règles des routes disjointes.

A fin de remédier à ce problème et d'éviter l'insertion des routes les moins stables avant les routes les plus stables, nous avons utilisé pour cela une autre table temporaire, ROUTE_STORE, sert à stocker, selon la valeur RET, toutes les routes reçues même si elles ne sont pas disjointes (figure 5.15), après un certain temps t bien défini, les routes seront insérées depuis la table intermédiaire vers la table de routage, bien entendue si les routes sont disjointes, de cette manière on favorise que les routes stables(figure 5.16). La table intermédiaire, ROUTE_STORE, est libérée une fois les routes sont insérées dans la table de routage.

Remarque importante

Le fait d'envoyer pour chaque nouvelle route disjointe trouvée, un paquet de données, garantit que les nœuds intermédiaires par lesquels le paquet de donnée est passé informent la source quand il y a un changement d'état de lien. Autrement dit, quand les nœuds intermédiaires reçoivent le premier paquet de données correspond au couple $< source, destination >$ ils extraient à partir du paquet de données, la route partielle vers la source et l'insèrent si elle est disjointe(voir Figure 5.17). Par conséquent, la réception des paquets de données par les nœuds intermédiaires signifie que ces nœuds sont actifs et concernés par le routage, donc quand il y a un problème entre ces nœuds et ses voisins directs ils doivent informer la source par ce changement d 'état pour que la source puisse choisir d'autre route(éviter le perte de données). Le processus de collection des routes(la réception des RREPs par le nœud source) continue jusqu'à l'expiration de timeout correspond au nœud source.

Une fois la collection des routes est terminée et afin de router les paquets de données vers la destination souhaitée, la source a le choix d'utiliser les routes soit parallèlement soit alternativement :

Parallèlement : Dans cette approche, la source utilise toutes les routes trouvées à la

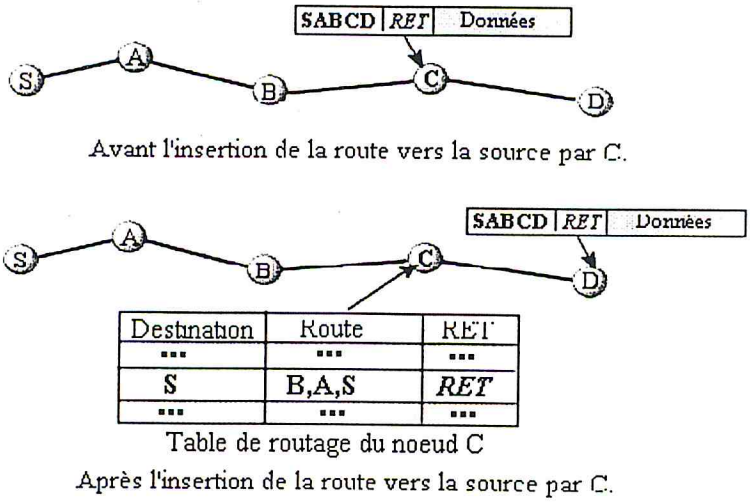


FIG. 5.17 – La sauvegarde de route par les nœuds intermédiaires.

fois pour transférer les paquets de données. Quand il y a une route qui est sur le point d'expiration, la source met cette dernière dans la structure d'attente des routes et choisie une autre route si : Une telle route existe et elle n'est pas en cours d'utilisation (nouvellement établie).

Cette approche est très pratique notamment dans le cas où un nœud intermédiaire est utilisé par plusieurs nœuds source ce qui réduit considérablement l'encombrement des routes ;

Alternativement : Dans cette approche, la source utilise une seule route durant le transfert des données, quand cette route est sur le point d'être expirer, la source met cette route en attente et choisie, suivant l'ordre des routes.

5.3.2 La maintenance des routes

Afin d'assurer la validité des routes utilisées, MPRP exécute une procédure de maintenance de routes. Le processus de maintenance de routes est requis quand il y a un changement d'état des liens formant une route utilisée par un nœud source, le changement d'état des liens aura en lieu soit à cause de la mobilité des nœuds soit à cause de la défaillance inattendue de ces derniers.

Quand un lien relayant un nœud **n** à un nœud qui le suit dans le chemin de routage a

changé son état, le nœud **n** envoie[4] un message de notification au nœud source.
Les paquets (messages) de notifications utilisés par MPRP sont en deux types :
NTF_LOW ce paquet informe la source qu'un lien entre un nœud et son voisin est devenu faible. Or, le paquet **NFT_STR**, informe la source qu'un lien reliant un nœud et son voisin est rétabli et devenu fort.
Le paquet d'erreur de route RERR(Route ERRor) n'est envoyé que si un nœud ne reçoit plus de beacon de son voisin (voir plus loin).
Quand la source a reçu le message de notification faible,NTF_LOW(voir la figure 5.19), la source procède comme suit :
Elle met la route contenant le nœud détectant le changement d'état dans une nouvelle structure appelée "Structure d'attente de routes", incrémente le compteur associé et choisie une autre route, selon l'ordre, si une telle route existe (voir figure 5.19).

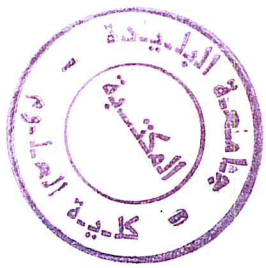
① Il se pourrait que la deuxième route s'expire avant la première route. Mais ceci est un exemple et juste pour clarifier les choses et montrer le principe de fonctionnement ;

② Le nœud source et pour chaque paquet de notification faible, vérifie si le nombre de routes vers la destination est inférieur ou égale à deux, si c'est le cas la source va déclencher une procédure de recherche de routes pour la construction proactive de route vers la destination ;

Chaque message de notification faible (NTF_LOW) a le format suivant :

src	dest	Notificateur	nouveau RET	chemin

FIG. 5.18 – Format d'un paquet de notification "Faible" dans MPRP.



Où :

src : L'adresse du nœud source qui doit être informé par le changement d'état de lien ;

dest : Seules les routes qui ont cette adresse sont concernées par la notification.

Notificateur : L'adresse du nœud détectant le changement d'état ;

nouveau RET : La nouvelle durée de vie d'un lien (donc la nouvelle durée de vie de la route) ;

chemin : La séquence des nœuds à parcourue par le message de notification NTF_LOW pour atteindre la source.

[4]Pas tous les nœuds détectant un changement d'état de lien envoient un message de notification au nœud source, seuls les nœuds concernés par le routage, de cette manière on réduit beaucoup l'envoi de ces messages.

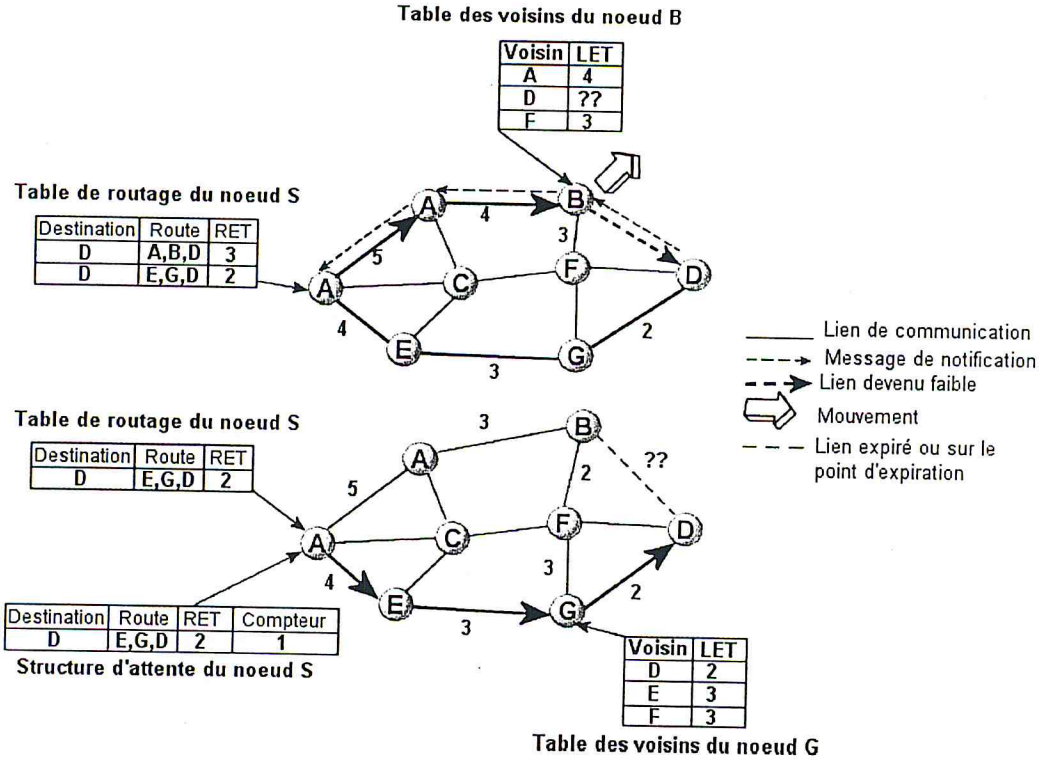


FIG. 5.19 – Schéma récapitulatif du processus de maintenance des routes dans MPRP.

Si la source a reçu un message NTF_FORT, elle décrémente le compteur (dans la structure d'attente de routes) correspond à la route contenant le nœud qui a envoyé le message NTF_FORT(Notificateur) et met à jour la valeur du RET (toujours dans la structure d'attente de routes) par la valeur LET envoyée dans le message NTF_FORT. Si la valeur du compteur a atteint la valeur 0, c à d. que la route en question ne contient que les liens forts (la route est devenu stable) donc, la source peut réinsérer à nouveau cette route dans sa table de routage selon la nouvelle valeur RET, si la valeur du compteur reste supérieure à 0 c à d. la route en question contient d'autre(s) lien(s) faible(s) donc, elle doit attendre d'autre(s) message(s) NTF_FORT concernant cette route.
Chaque message de notification NTF_FORT a le même format que le message de notification Faible NTF_LOW sauf, le type de paquet qui est un paquet de notification fort :
Où :

src	dest	Notificateur	nouveau RET	chemin

FIG. 5.20 – Format d'un paquet de notification "Fort" dans MPRP.

src :L'adresse du nœud source qui doit être informé par le changement d'état de lien ;

dest : Seules les routes qui ont cette adresse sont concernées par la notification.

Notificateur :L'adresse du nœud détectant le changement d'état ;

nouveau RET :La nouvelle durée de vie d'un lien (donc la nouvelle durée de vie de la route) ;

chemin :La séquence des nœuds à parcourue par le message de notification NTF_STR pour atteindre la source.

Pour chaque paquet de notification faible reçu par un nœud source, une vérification de nombre de route vers la destination concernée par le routage est faite : Tant qu'il y a des données à transmettre et le nombre de routes est inférieure à deux(2), une requête de route, RREQ, est initiée afin de construire *proactivement* les routes.
Il se pourait que la dernière route active est suffisante pour transférer le reste des paquets de données, c'est pour cette raison on n'initier pas une RREQ à chaque fois le nombre de routes est inférieur à deux, un nouveau processus de découvert de routes n'est initie que si la durée de vie de la dernière route est inférieure à OPERATION_TIME plus certaine une valeur VAL où : OPERATION_TIME est le temps suffisant pour trouver(découvrir) d'autres routes vers la destination.
Les nœuds du réseau s'échangent périodiquement des beacons (message de contrôle) afin de montrer leur existence aux autres nœuds. Pour chaque beacon reçu, le nœud recevant le beacon met à jour la valeur LET associée au nœud voisin envoyant le beacons et vérifier si le LET est supérieur ou égale a une certaine valeur seuil(décrit précédemment) de cette manière le nœud informe la source quand il y a un changement d'état de lien.
Si un beacon n'est pas reçu après deux seconds à partir d'un nœud voisin, le lien en question est considéré comme défaillant. Le nœud détectant la défaillance fait le suivant :

① Supprime toutes les routes qui ont le prochain saut le voisin perdu ;

② Envoie un paquet d'erreur de route RERR(Route ERRor) au nœud source ;

③ S'il a des paquets données à transmettre pendant la défaillance inattendue de lien, il va essayer de trouver une route partielle vers la destination pour sauver les données, si une telle route n'existe plus le paquet est ignoré.

Une fois le paquet d'erreur de route RERR a atteint la source, cette dernière va supprimer la route active utilisée et choisie une autre route alternative si elle a des données à transmettre et supprime définitivement la route concernée par ce nœud voisin de la table

de routage ou de la structure d'attente.

Remarque : Normalement la source ne supprime pas directement les routes depuis la table de routage, mais elle les en supprime de la structure d'attente, car avant de passer à l'état défaillant du lien(voir la figure 5.5), en passe avant à l'état faible de lien donc les routes contenant des liens faibles seront mises en attente. La suppression directe n'est effectuer que dans un seul cas : la défaillance(panne inattendue) du lien et/ou nœud.
De même pour les paquets d'erreur de route, RRER. Quand ils sont reçus par un nœud source, tant qu'il y a des paquets de données à transmettre et le nombre de route est inférieur à deux(2)[5], une RREQ est initié pour construire des routes vers la destination concernée par le routage.
Il n'est pas logique d'attendre jusqu'à la prochaine "deuxième seconds" pour savoir si le voisin avec lequel en lui transmet les paquets de donnée reste connecté ou non notamment pendent le transfert des données.
Par conséquent, afin d'éviter ce genre de problème, deux approches sont utilisées dans notre protocole :

① **Échange périodique des beacons :**
Cette approche a été bien expliquée précédemment ;

② **Intervention de la couche MAC :**
Étant donnés un nœud A et son voisin B, pour chaque paquet envoyé par la couche MAC du nœud A, un accusé de réception (ACK) du nœud B doit être reçu par A, si un tel accusé n'est reçu, la couche MAC du nœud A envoie un message d'erreur à la couche IP et le nœud B est considéré comme perdu(cette approche est utilisée pour tous les types de paquets : beacon, donnée, erreur, etc.).

5.4 Conclusion

Dans ce chapitre nous avons présenté MPRP pour assurer le service de routage dans les réseaux mobiles Ad hoc, MPRP est un protocole de routage suit une approche réactive où les routes ne sont construites qu'à la demande. Comme nous avons vu, MPRP est un protocole prédictif, utilise comme moyen de prédiction les puissances des signaux reçus des nœuds voisins. En plus de ça, les nœuds dans MPRP stockent plusieurs routes disjointes pour même destination.
Les variétés de techniques utilisées par notre protocole, ont pour but de donner de

[5] Le processus de découvert de route n'est initier que si $RET \leq OPERATION_TIME + VAL$.

meilleures performances de routage (exemple, la fraction de réception de données) par rapport aux protocoles de routage existant qui ont montré de bonnes performances. L'interprétation des résultats obtenus par la simulation est présentée dans le chapitre suivant.

Chapitre 6

Environnement et résultats de simulation

6.1 Introduction

Les protocoles de routage doivent être évalués afin de mesurer les performances de la stratégie utilisée et de tester sa fiabilité. Pour faire une telle évaluation on a souvent recours à la simulation. En effet, il serait trop coûteux, voire impossible, de mettre en place un réseau à des fins de tests pour certains critères. Par exemple, tester des applications sur des réseaux de grande envergure n'est possible en réalité que si l'on dispose de moyens matériels importants. Or, dans le cadre de simulation, les moyens matériels que nous en parlons sont totalement absents(sauf la machine exécutant le simulateur est nécessaire). Donc, il suffit de changer les paramètres de simulation correspondant à la taille du réseau pour d'autres essais ou même pour d'autres statistiques importantes. Actuellement, déférents logiciels de simulation sont disponibles permettant de simuler toutes les couches réseau d'une manière déterministe (identiques).

Ce chapitre et avant d'entreprendre l'interprétation des résultats obtenus par la simulation, il va d'abord présenter l'environnement de simulation, de tel sort à présenter et définir tout ce qui est nécessaire pour l'étape de la simulation.

6.2 Définitions de la simulation

Le terme *simuler* signifie "représenter la réalité par un moyen quelconque". Autrement dit, la simulation d'un système est l'opération d'un simulateur qui est une représentation du système. Le simulateur se prête à des manipulations qui seraient impossibles, trop coûteuses ou non pratiques à effectuer sur le système.
La simulation utilise souvent les programmes informatiques pour imiter les événements causals et les actions conséquentes dans le système. Les statistiques sont accumulées et sauvegardées durant la période de simulation pour évaluer les mesures d'intérêts qui sont résumées et rapportées à la fin de l'exécution du programme de simulation.

6.3 Le langage de simulation PARSEC

PARSEC(**PAR**allel **S**imulation **E**nvironment for **C**omplex systems)[BAG 98], est un langage de simulation développé par le PCL (Parallel Computing Laboratory) de l'université UCLA[1], a pour but l'exécution séquentielle ou parallèle d'un modèle de simulation à évènements discrets. PARSEC se base sur le langage C-ANSI avec quelques possibilités supplémentaires héritées du C++. PARSEC introduit les notions d'*entités* (blocs autonomes qui peuvent s'exécuter en parallèle) et de *messages* entre ces entités ce qui est très utile pour la simulation de nœuds s'échangeant des informations.

6.3.1 Entité-type

Pour réaliser une simulation, GloMoSim utilise une approche simple. En effet, si nous supposons que chaque nœud dans la simulation est une entité à part, alors nous allons remarquer non seulement une dégradation mais aussi une limitation des performances de la simulation car l'initialisation d'un millier d'entités peut influer sur le temps d'exécution et sur l'utilisation de l'espace mémoire. Pour palier à ce problème, GloMoSim fait recours à une approche d'agrégation de nœuds où une seule entité peut simuler divers nœuds réseaux à la fois.

- Chaque nœud a sa propre structure de données et son code doit s'exécuter sans interférence ni violation d'accès aux structures de données des autres nœuds ;
- Chaque nœud à son propre état et l'ensemble des états complets de tous les nœuds est maintenu par l'entité.

[1] University of California Los Angeles.

Cette dernière approche permet d'augmenter le nombre de nœuds dans une simulation sans augmenter le nombre d'entité.
Une entité n'est autre qu'une zone géographique où chaque nœud est déterminé par ses coordonnées dans cette zone.
Un programme PARSEC est une collection d'entité-type et de fonctions écrites en C. Chaque entité-type décrit une classe d'objets. Une instance d'entité-type, appelée "**entité**", représente un objet dans le système physique. L'instanciation des entités peut se faire par l'utilisation de l'instruction **new**.
Chaque programme PARSEC doit contenir une entité ***driver***, cette entité a le même rôle que la fonction main du C (i.e. le point de départ d'un programme), dans notre cas l'entité driver est responsable du lancement de la simulation.

6.3.2 Message

La communication entre les entités est matérialisé par les échanges de messages entre ces dernières. La définition d'un message en PARSEC est similaire à la définition d'une structure en C. chaque entité doit avoir un buffer et deux primitives : l'une, **receive**(), pour la réception des messages à partir de son buffer, et l'autre, **send**(), pour l'envoie des messages vers un *buffer* d'autres entités.

6.3.3 Événement

Dans le modèle de simulation, on associe à chaque événement une estampille qui indique à quel moment l'événement apparaît dans le système.
Puisqu'un événement dans PARSEC est modélisé par un message, ce message doit porter l'estampille. Par défaut, l'estampille indique le temps(de simulation) d'envoi de ce message.

6.3.4 Le simulateur GloMoSim

GloMoSim(**GLO**bal **MO**bile information system **SIM**ulator)[BAJ 00], est un environnement de simulation et d'évaluation de performance pour les réseaux sans fil de grande taille, Il fournit un environnement assez détaillé permettant entre autre de réaliser des simulations d'IP, TCP, du routage et des protocoles multicast aussi bien sur des liens filaires que sans fil.

Couche	Option
Application	Telnet, FTP, **CBR** and HTTP
Transport	TCP, **UDP**
Réseau	**IP**
Routage	Bellman-Ford, Fisheye, AODV, DSR, LAR1, WRP.
MAC	CSMA, FAMA, MACA, **IEEE 802.11b**.
Radio	**Radio with various noise parameters**
Propagation	Free space, **Two-Ray**, Generic
Placement des nœuds	Uniform, Grid, **Random**, Group Random, File
Mobilité	None, **Random Waypoint**, File

FIG. 6.1 – Organisation en couche du GloMoSim.

GloMoSim a été développé en PARSEC, la version utilisée dans ce mémoire est la version 2.03.
Après la conception et l'implémentation du protocole de routage il faut le compiler pour qu'il puisse l'exécuter sur le simulateur GloMoSim.

Organisation en couche du GloMoSim

GloMoSim a une structure de couches proche de celle du modèle OSI(voir la figure 6.1). Les options soulignés et mets en police gras dans le figure 6.1, sont utilisés dans les paramètres du réseau (voir plus loin).
En effet, chaque entité intègre les diverses couches réseaux qui peuvent être simulées sous forme de fonction. Au début de la simulation, le simulateur crée une entité **'Driver'** qui va lire les informations à partir d'un fichier de configuration et une fonction d'initialisation est appelée pour chaque couche de chaque nœud. La simulation se déroule ensuite sous forme d'échange de messages. L'entité reçoit un message d'une couche d'un nœud et l'envoie à la couche d'un autre nœud.
A la fin de la simulation un fichier est produit appelé **"GLOMO.stat"** contient toutes les statistiques demandées dans le fichier de configuration.

L'ajout d'un nouveau protocole dans GloMoSim

Chaque protocole ajouté a besoin de trois fonctions principales(voir figure 6.2 :

Fonction d'initialisation : Cette fonction permet d'initialiser à zéro toutes variables statistiques utilisées par la couche où elle est appelée (exemple : Nombre de beacons

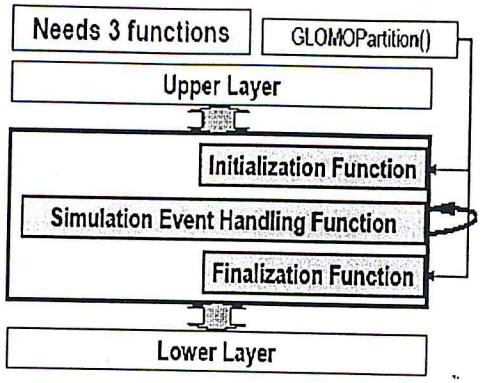


FIG. 6.2 – Les trois(3) fonctions d'intégration d'un protocole dans le simulateur.

reçu/envoyés, nombre de route utilisées pour chaque nœud, nombre de paquets perdus, etc.). La fonction d'initialisation permet ainsi d'initialiser toutes les structures de données utilisées par le protocole de routage (exemple : table de routage, la table des voisins, la structure d'attente, etc.)

Fonction de gestion(traitement) d'événements : Cette fonction traite les différents évènements qui peut survenir dans le réseau comme : l'envoi d'un message périodique, la vérification(donc détection)de défaillances des liens, etc.

Fonction de terminaison(finalisation) : Dans cet endroit où les statistiques de la couche où le protocole s'exécute seront collectées.

6.4 Les métriques de performances

Pour comparer (évaluer) notre protocole routage par rapport aux autres protocoles existants, nous avons utilisé quatre paramètres de comparaison pour juger les performances des protocoles de routage. Ces paramètres sont donnés comme suit :

Fraction de réception de données(*Packet delivery ratio*) C'est le nombre de paquets reçus par rapport au nombre de paquets envoyés au niveau de la couche application ;

La charge du réseau (*Routing overhead*) : C'est le nombre total des paquets de contrôle envoyés pendant la période de simulation ;

L'optimalité des routes(*Path optimality*) : C'est la durée de vie moyenne des routes actives utilisées par les nœuds sources pendent toutes la période de simulation.

Le délai moyen de transfert d'un paquet de données (*end-to-end delay*) : C'est le temps en secondes mis par un paquet pour traverser le réseau d'une extrémité (couche application cliente) à une autre (couche application serveur).

6.5 Paramètres du réseau

- **Temps de simulation :** 900s(15 Minutes) le temps choisi pour garanti que la topologie change totalement ainsi les routes !
- **Modèle de mobilité :** Dans ce modèle un nœud choisit aléatoirement une destination dans le terrain de la simulation et il se déplace à cet endroit en utilisant une vitesse choisie arbitrairement entre une vitesse maximale et une vitesse minimale spécifiées auparavant par l'utilisateur dans fichier de configuration (config.in). Lorsqu'il atteint sa destination, un nœud fait une pause pendant un temps aussi défini dans le fichier configuration (pause time).
- **Terrain de simulation :** Nous avons choisi un terrain carré de 1000 x 1000 mètres.
- **Nombre de nœud :** Pendent toute la période de simulation le nombre de nœud est fixé à 50 nœuds.
- **Bande passante :** La bande passante a été fixée à 2Mbps dans toute la période de simulation.
- **Couche MAC :** Le simulateur possède quatre protocoles pour la couche MAC, nous avons utilisé comme protocole 802.11b.
- **Modèles de propagation :** GloMoSim permet également de choisir parmi les modèles de propagation suivants, qui influeront en particulier sur la manière dont seront atténués les signaux en fonction de la distance :

  Free space model(modèle de propagation en espace libre) : Ce modèle considèr le cas idéal où il y a un seul chemin de propagation entre l'émetteur et le récepteur et qu'il est en vue directe.

  Two-ray ground reflection model (modèle à deux rayons) : En environnement réel, il est en fait peu probable que le seul chemin de propagation soit le chemin direct. Le modèle two-ray ground considère donc à la fois le chemin direct et une réflexion sur le sol, ce modèle donne des résultats plus justes que le modèle de propagation en espace libre quand la distance est assez grande.

Nous avons choisi comme modèle de propagation le deuxième modèle(modèle à deux rayons) pendant toute la période de simulation.

- **L'application utilisée :** Généralement, pour juger les performances des protocoles de routage l'application CBR est utilisée. CBR est un acronyme de Constant Bit Rate(Débit Constant) une catégorie de trafic utilisée pour le transport d'information numériques comme la vidéo ou la voix, qui peut être représenté par un flux continu de bits. Le trafic CBR nécessite une bande passante et un niveau de service garanti. Chaque session CBR et pendent un intervalle de temps bien déterminé, envoie pour chaque seconds 1000 paquets de données dont la taille de chaque paquet est égale à 1024 octets (1Ko).

6.6 Paramètres de comparaison

1. **Mobilité :** Dans cette section, nous varions la mobilité et on fixe les autres paramètres (le nombre de session et la connectivité) et on observe l'effet du mobilité aux métriques.

La vitesse de mobilité des nœuds	0, 2, 5, 10, 15 et 20 m/s.
Nombre de session CBR	fixé à 13 sessions
Portée de communication des nœuds	fixée à 250m

2. **Charge du réseau :** Dans cette étape, nous varions le nombre de session CBR et on fixe les autres paramètres et on observe l'effet de la charge sur les autres métriques.

Nombre de session CBR	2, 4, 6, 8 et 10.
La vitesse de mobilité des nœuds	fixée à 10m/s
Portée de communication	fixée à 250m

3. **Connectivité du réseau :** La connectivité sert à augmenter le nombre de connexion entre les nœuds du réseau. Pour faire ceci nous varions la portée de communication des nœuds. De cette manière plus la portée de communication augmente plus le nombre de connexion entre les nœuds augmente(la taille du terrain de simulation reste inchangée).

Portée de communication des nœuds	111, 198,250, 352 et 627 mètres.
La vitesse de mobilité des nœuds	fixée à 10 m/s
Le nombre de sessions CBR	fixé à 6 sessions

6.7 Métriques de simulation

6.7.1 Mobilité :

Fraction de réception de données(Packet delivery ratio) :

L'effet de mobilité sur la fraction de réception de données est représenté par la figure 6.3. Comme on peut voir sur la figure que, plus la mobilité augmente plus l'efficacité de routage de ABR et DSR se dégrade. Les deux autres protocoles AODV et MPRP ne sont pas beaucoup affectés par l'augmentation de la mobilité. La détection

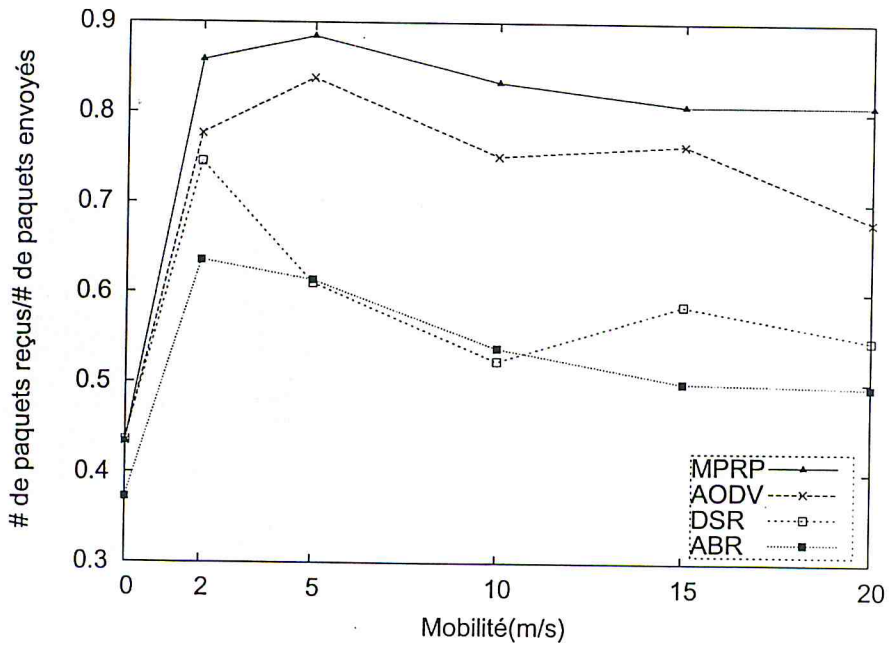


FIG. 6.3 – La fraction de réception de données en fonction de la mobilité.

des routes défaillantes et les changements des routes actives dans AODV, ABR et DSR ne sont faits qu'au moment de l'envoi d'un paquet de données à travers le lien défaillant, ce paquet sera perdu si le nœud détectant la défaillance ne possède pas une route vers la destination du paquet. Or ce qui n'est pas le cas pour MPRP, les routes qui sont sur le point d'être défaillant seront changées par d'autres routes plus stables, ce qui explique la bonne performance du MPRP par rapport aux autres protocoles.

Un nœud dans MPRP doit attendre périodiquement deux second pour savoir l'état-de lien qui le relie avec son voisin, donc la détection de changement d'état de lien (faiblesse ou défaillance) n'est effectué que dans les deux prochaines secondes suivantes, il se pourrait, notamment dans la grande mobilité(comme la figure 6.3 le montre), que les nœuds sources dans MPRP commencent le transfert des paquets de données avant qu'elles sachent qu'il y a une défaillance de(s) route(s) ce qui explique la dégradation de l'efficacité de routage de MPRP dans la grande mobilité des nœuds.
MPRP s'adapte à la mobilité de l'environement en stockant plusieurs routes vers même destination, ce qui fait beaucoup de changements de topologie n'auront pas d'effet sur le routage des données. à moins que tous les chemins qui mènent vers la destination soient perdus.
Dans le cas d'une topologie statique (Mobilité = 0), la fraction de réception de données est très petite, ceci est du au phénomène de partitionnement du réseau (pas de communication entre certains nœuds du réseau). Ce qui fait qu'un nombre très petit de nœuds source va trouver des routes vers leurs destinations alors que les autres nœuds n'auront pas la possibilité d'accéder à ces destinations, ce qui explique la mauvaise performance.

La charge de routage (Routing overhead) :

Le nombre de paquets de contrôles transmit en fonction de la mobilité est présenté dans la figure 6.4, plus la mobilité augmente plus le nombre de paquets de contrôle générés par les protocoles augmente notamment dans MPRP et ABR. La figure 6.4, montre également que malgré qu'il n'y a pas de mobilité, ABR et MPRP génèrent beaucoup de paquets de contrôle par rapport à DSR et MPRP, dans cette période le réseau est partitionné, les nœuds sources du réseau continuent d'envoyer les paquets requêtes de routes tant qu'ils n'ont pas encore reçu des routes vers leurs destinations, la raison que le nombre de paquets de contrôle dans DSR et MPRP est moins grand que AODV et ABR est du à l'intervalle de transmission des requêtes de routes des protocole, tel que pour AODV et ABR la transmission de paquet requête de route s'effectue respectivement pour chaque 50 et 80 milli-secondes, tandis que pour DSR et MPRP, elle s'effectue pour chaque 90 milli-secondes.
En outre, qui dit mobilité dit (généralement) un changement de topologie, donc le changement des routes actives sauvegardées dans les tables de routage des nœuds sont augmentés . La probabilité pour que ces routes restent valident est très petite. Par

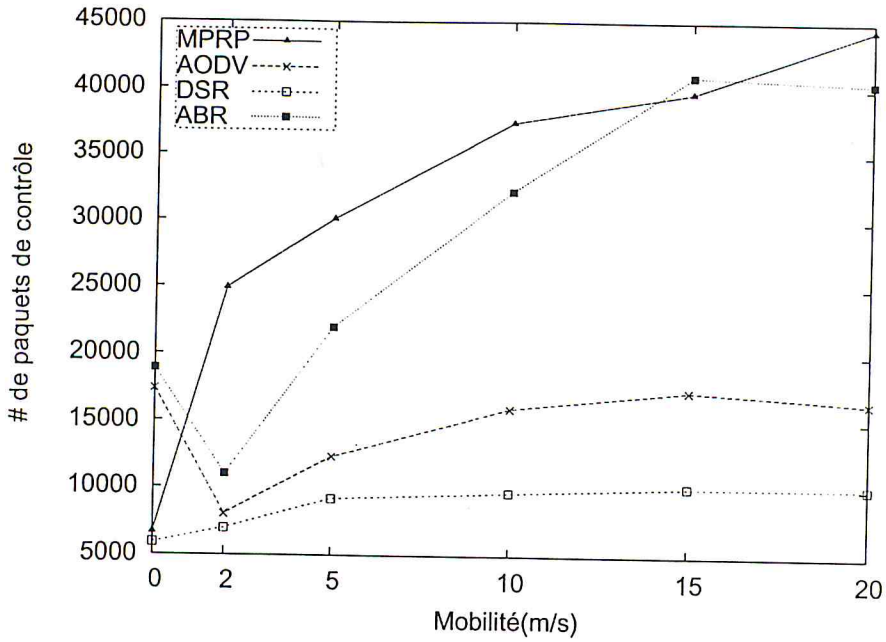


FIG. 6.4 – Le nombre de paquets de contrôles en fonction de la mobilité.

conséquent, plus les routes deviennent instables ou invalides plus le nombre de paquets de contrôle (Erreur ou notification) augmente. De plus, plus la mobilité augmente, plus le nombre de routes diminue, ce qui implique, et afin de trouver d'autres routes, plus de transmissions des RREQ qui va augmenter sans doute le nombre de paquets de contrôle. On remarque que DSR génère un nombre de paquet légèrement inférieur par rapport à AODV, car DSR utilise son cache de route et dans le cas où il trouve un chemin vers la destination il n'aura pas besoin de rechercher de nouveau des routes vers la même destination, de cette manière, un nombre important de paquets de requête de routes est réduit, ce qui implique une réduction importante de nombre de paquets de contrôles. Dans AODV, un timer est associé à chaque route, la route non utilisée après un certain temps t sera considérée défaillante. Donc un paquet d'erreur de route est envoyé, c'est pour cette raison que AODV génère plus de paquets de contrôle que DSR.

Le délai moyen de transfert de données (end-to-end delay) :

Le retard pris par un paquet de données d'aller d'un bout (couches applications des nœuds sources) à l'autre bout(couches applications des nœuds destinations) est montré dans la figure 6.5. On remarque que le temps de bout en bout de AODV est relativement

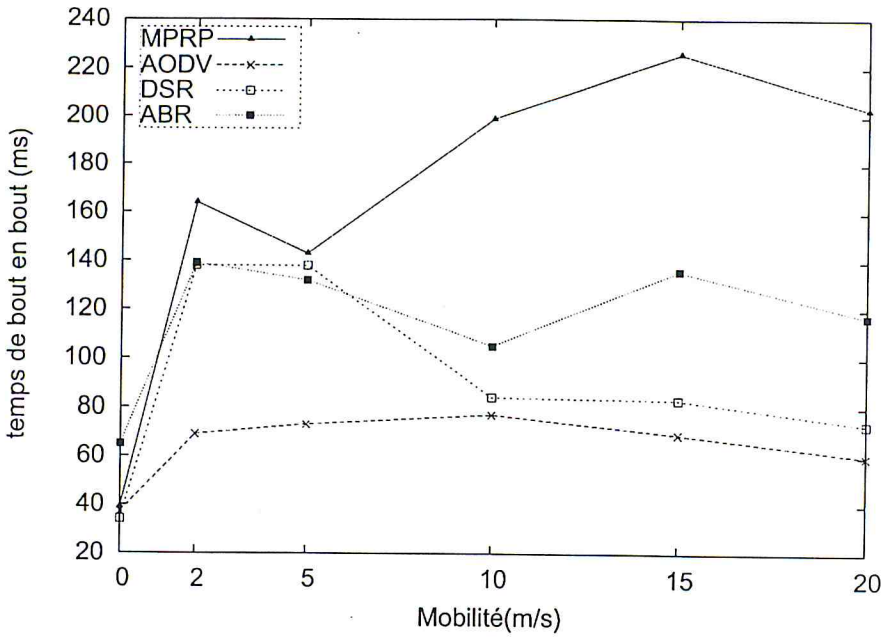


FIG. 6.5 – Le délai moyen de transfert en fonction de la mobilité.

faible par rapport à DSR et ABR. Dans AODV, les nœuds destination ne répondent qu'aux premières requêtes reçues. Donc ils répondent aux requêtes venues suivants les routes les plus courtes et les moins encombrées, ce qui est n'est pas le cas dans ABR et MPRP, tel que les destinations dans ABR attendent plus d'une requêtes et dans MPRP les sources doivent attendre également plus d'une réponse pour choisir les meilleures routes qui ont une longue durée de vie. Ce qui justifie le grand temps de bout en bout dans ABR et MPRP. Ceci est d'un côté, dans l'autre côté, les réponses de routes dans MPRP quand elles sont reçues par les nœuds intermédiaires, ces derniers vont accéder aux entrées correspondantes dans les tables des voisins pour avoir les durées de vie des liens et mettre à jour les champs RETs des réponses de route ce qui induit un temps de bout en bout plus grand que les autres protocoles.

Plus la mobilité augmente, plus le nombre de contrôle augmente (voir la charge en fonction de la mobilité) plus le nombre de collision augmente, Ces collisions doivent être détectées, pour que la couche MAC puisse retransmettre le paquet sans avoir à repasser par les couches supérieures, ce qui engendrerait des délais significatifs.

L'optimalité des routes(Path optimality) :

MPRP et ABR sont conçus principalement pour favoriser les routes qui ont une longue durée de vie. On peut observer ça dans la figure 6.6.
La raison d'être AODV montre une mauvaise durée ceci est du à deux raisons principales :

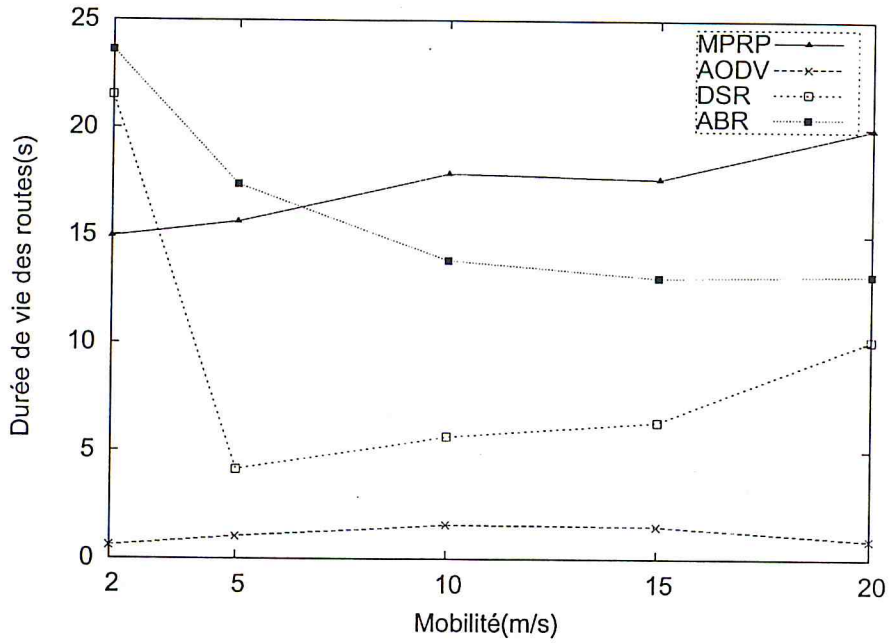


FIG. 6.6 – L'optimalité des routes en fonction de la mobilité.

La première raison est la possibilité aux nœuds intermédiaires de répondent à la source par une réponse de route quand ils reçoivent une RREQ et qui connaissent des routes vers la destination, il se pourrait que les routes partielles connues vers les destinations aient des durées de vie plus petites, c'est pour cette raison quand elles sont utilisées, elles durent peu de temps et s'expirent. La deuxième raison dès dès que les noeuds destinations reçoivent la première RREQ, ils vont répondre automatiquement sans attendre d'autres RREQs qui viennent, donc AODV répond au RREQ venues des routes plus courtes et les moins encombrées ce qui n'est pas le cas dans le protocole ABR et MPRP car ils utilisent comme un paramètre de de privilège, la puissance du signal. donc, ils sont capables de distinguer entre les routes qui ont de grandes durée de vie et les routes qui ont de petites durée de vie.
Les techniques utilisées par DSR et MPRP dans la construction des routes sont presque identiques, i.e. la possibilité de trouver plusieurs routes pour la même destination : il se

pourrait qu'il existe une ou plusieurs routes parmi ces routes trouvées qui ont des durées de vie plus longue c'est pour cette raison DSR donne une durée plus grande que AODV et plus petite que ABR et MPRP.

6.7.2 Charge d'application(Nombre de session) :

Fraction de réception de données(Packet delivery ratio) :

La mobilité dans ce scénario est fixée à 10m/s. malgré cette grande vitesse le protocole MPRP donne des meilleures fractions de réception de données par rapport à AODV dans tous les cas sauf pour le cas de deux sessions (voir la figure 6.7). La

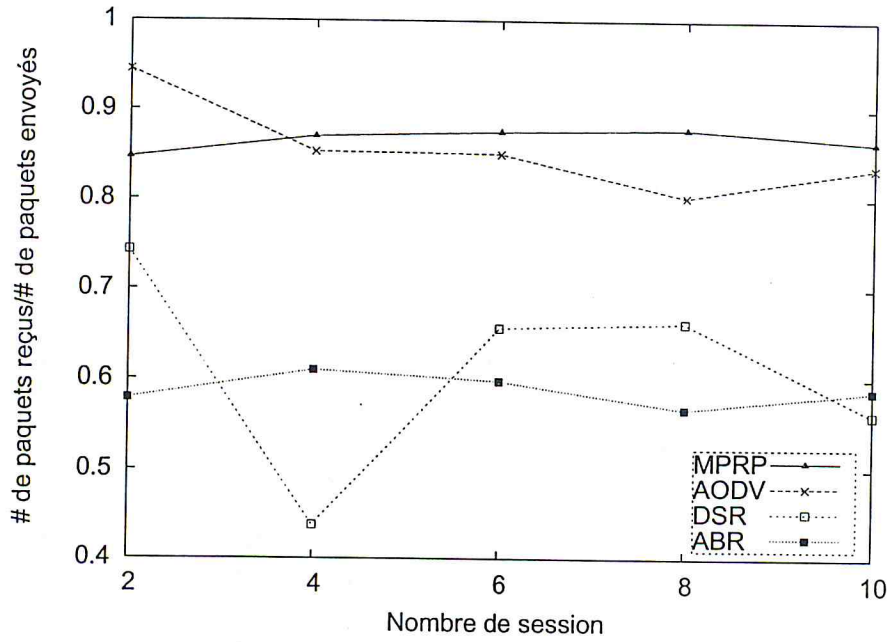


FIG. 6.7 – La fraction de réception de données en fonction du nombre de session.

dégradation de la fraction de MPRP dans deux sessions peut être expliqué comme suit : Le routes favorisées par MRPR sont des routes qui ont une longue durée de vie, il se pourrait que pour deux nœuds source différents pour deux nœuds destinations différents les routes choisies par MPRP partagent les mêmes nœuds et/ou liens, donc la transmission de données en même temps peut surcharger les routes(congestion) qui va influer sans doute sur la fraction de réception de données.

En ce qui concerne le protocole ABR et DSR, la forte mobilité a causé une grande perte de données. En plus de ça, une grande quantité de paquets de contrôles qui a beaucoup

influer sur la fraction de réception de données.

La charge du réseau (Routing overhead) :

On remarque de la figure 6.8, que le nombre de paquets de contrôle de ABR, MPRP et AODV croit linéairement avec le nombre de session tandis qu'il est relativement faible pour AODV. Plus le nombre de session augmente plus la charge de routage augmente.

DSR quant à lui montre un comportement non monotone.

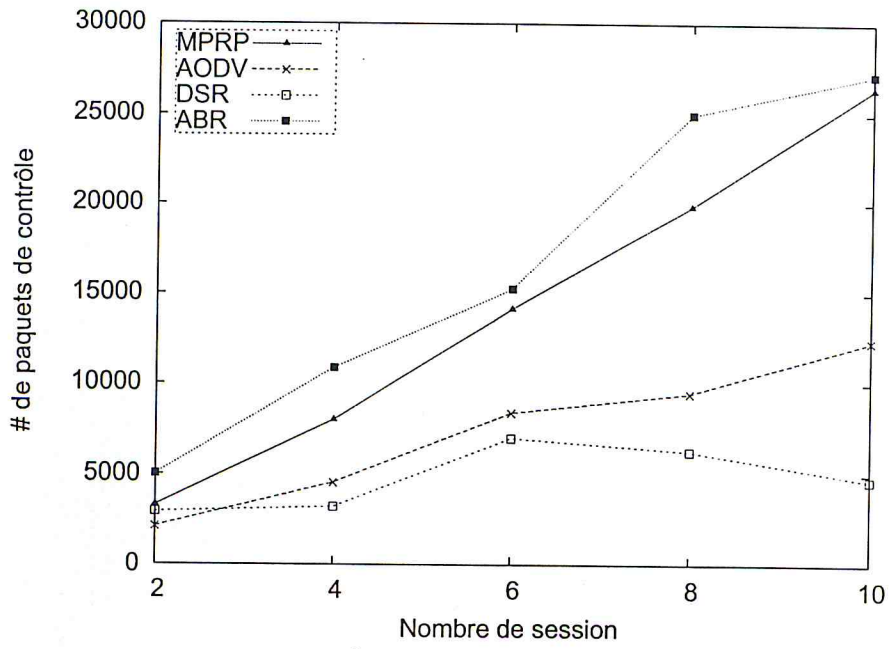


FIG. 6.8 – Le nombre de paquet de contrôle en fonction du nombre de session.

L'augmentation de nombre de session nécessite plus de paquets de contrôle (paquets RREQs et RREPs pour construite des routes et paquets RERRs pour les maintenir). Le nombre de paquets de contrôle dans ABR et MPRP est très grand car ces deux derniers utilisent d'autres paquets en plus des paquets de contrôle cités précédemment, des paquets de notification pour MPRP et des paquets de localisation des chemins partiels pour ABR.

DSR génère un petit nombre de paquets de contrôle avec l'augmentation du nombre de session. Plus le nombre de sessions augmente plus le nombre de routes stockées dans les caches de routes des nœuds du réseau augmente ce qui diminue considérablement le

déclenchement fréquent du processus de recherche de route qui va sans doute diminuer considérablement l'envoi des paquets de requête de routes (RREQ).

Le délai moyen de transfert de données (end-to-end delay) :

On remarque dans la figure 6.9 que presque tous protocoles régissent de la même manière. Plus le nombre de session augmente plus le temps de bout en bout augmente, car, plus le nombre de session augmente plus le réseau est chargé et plus le nombre de collision augmente ce qui oblige les couches MAC d'attendre plus de temps pour retransmettre de nouveau les paquets (de données et/ou de contrôles) afin d'éviter les collisions. Le temps de MPRP est plus grand en raison de l'approche utilisée par ce

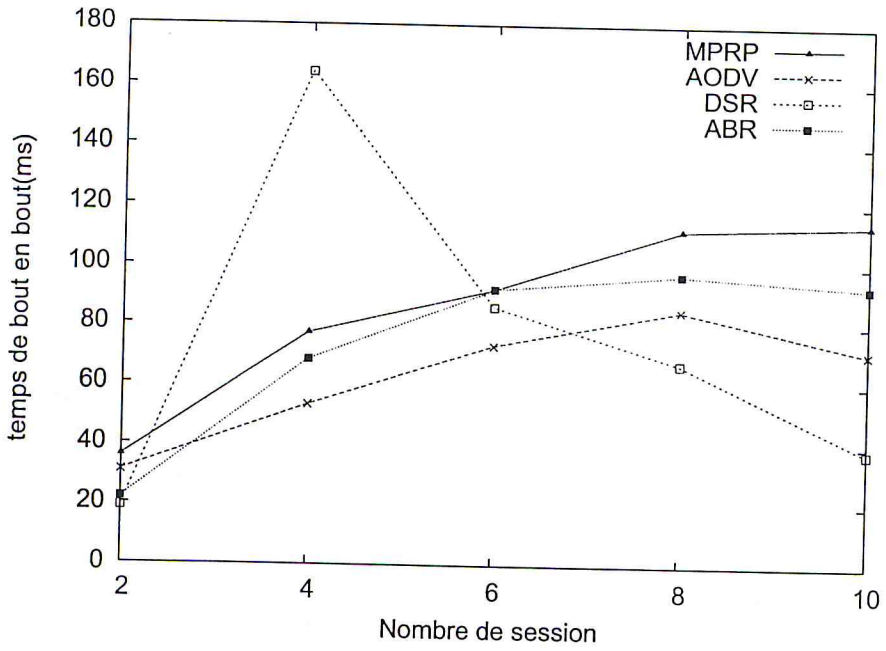


FIG. 6.9 – Le délai moyen de transfert en fonction du nombre de session.

dernier quand les réponses de routes sont transmises d'un nœud à un autre et quand les réponses sont reçues par les nœuds sources (voir Le délai moyen de transfert de données en fonction de la mobilité).

Quant à DSR, le comprtement est aléatoire.

L'optimalité des routes(Path optimality) :

Il est possible qu'il existe de routes plus longues que celles qui existent dans les caches des routes, les nœuds quand ils veulent transmettre des données consultent d'abord leurs caches si des telles routes existent, il les utilise directement sans avoir le privilège d'utiliser les routes les plus longues, ce qui explique le comportement aléatoire de DSR et MPRP(voir la figure 6.10). L'optimalité moyenne des routes utilisées dans

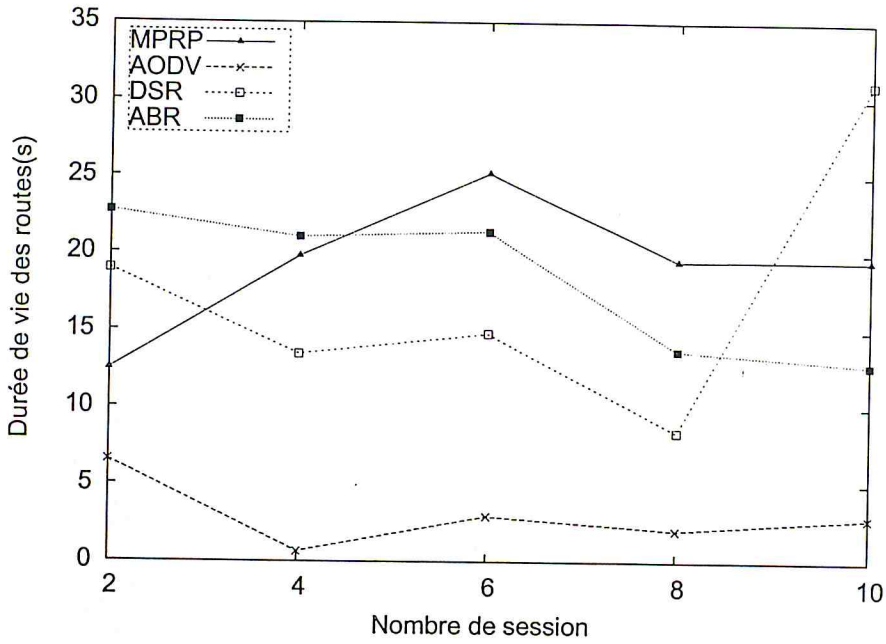


FIG. 6.10 – L'optimalité des routes en fonction du nombre de session.

AODV restent relativement fiables car ce protocole favorise les routes les plus courtes et les moins encombrées mais pas les routes qui ont plus longue durée de vie.

6.7.3 Connectivité :

la connectivité sert à augmenter le nombre de liens entres les nœuds du réseau. Pour faire ceci, nous avons varié le rayon de la portée de communication des nœuds de tel sort le rayon prend respectivement les valeurs suivantes : 111m, 198m, 250m, 352m et 627m.

Fraction de réception de données(Packet delivery ratio) :

On remarque dans la figure 6.11 que plus la connectivité augmente plus la fraction de réception de données augmente, car quand la connectivité augmente le nombre de voisins de chaque nœud augmente et les nœuds deviennent de plus en plus liés(liaison de la topologie) alors la probabilité que le paquet de données arrive à sa destination augmente.

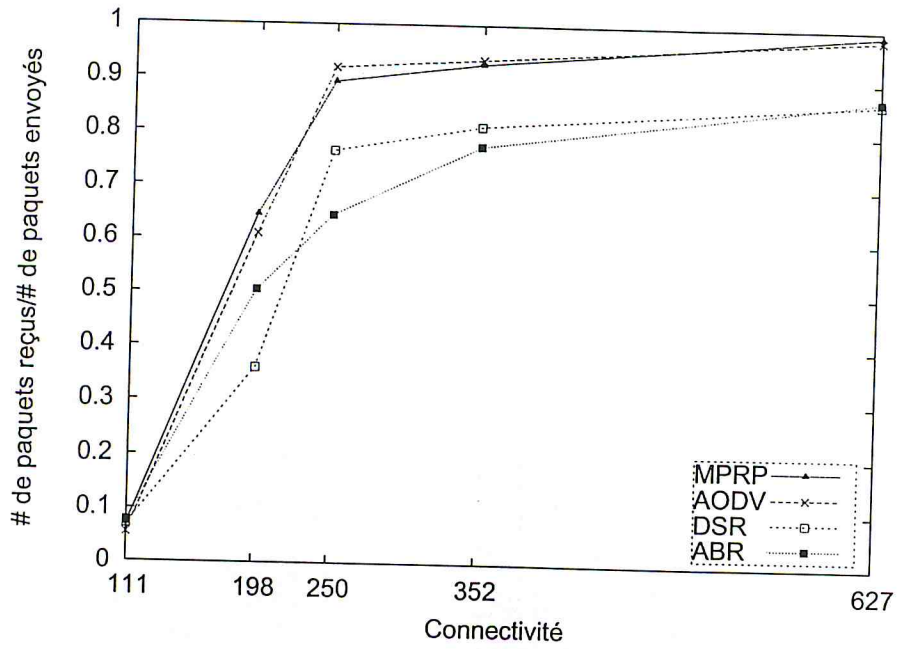


FIG. 6.11 – Le nombre de paquet de contrôle en fonction de la connectivité.

La charge du réseau (Routing overhead) :

La figure 6.12 montre que plus la connectivité augmente plus le nombre de paquets de contrôle est plus petit, Plus la connectivité augmente plus le nombre de routes augmente ce qui permet aux protocoles de routage de trouver rapidement les routes dès les premières requêtes de routes sans avoir les retransmettre plusieurs fois. Cela a comme conséquence ; un petit nombre des requêtes de route (i.e. moins de paquets de contrôle). Le grand nombre de paquet de contrôle de ABR et MPRP est du aux paquets de contrôle en plus utilisés par ces deux derniers(voir également la figure 6.12).

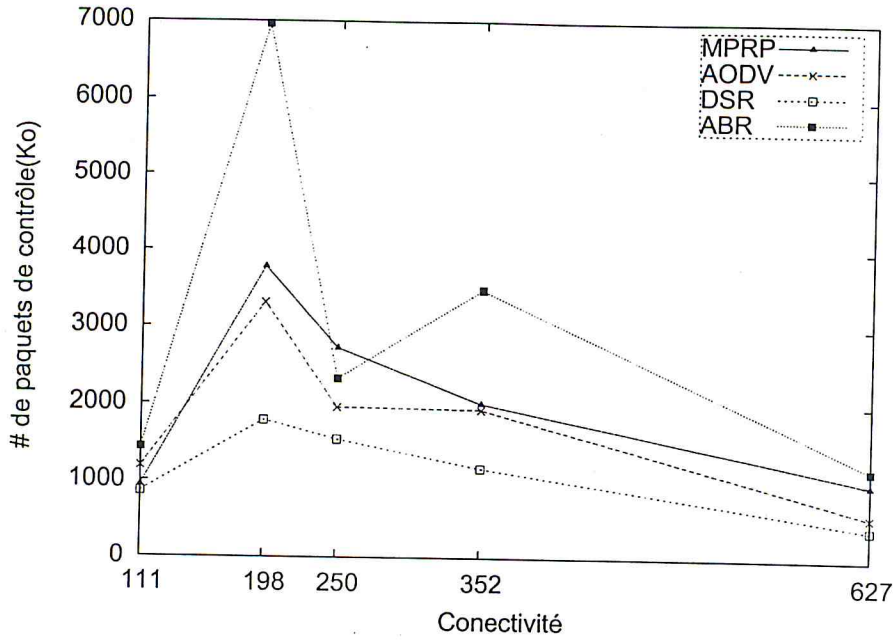


FIG. 6.12 – Le nombre de paquet de contrôle en fonction de la connectivité.

Le délai moyen de transfert de données (end-to-end delay) :

Plus la connectivité augmente plus la probabilité de trouver des routes vers la destination souhaitée est grande donc les paquets émis par les couches restent moins de temps dans leur buffer avant d'être envoyé, ce qui va donner un temps de bout en bout très élevé c'est ce que montre la figure 6.13. La grande mobilité dans ce scénario (10m/s), influe beaucoup sur la fraction de réception de données de ABR et DSR pour les raisons citées dans la section "Fraction de données en fonction de la mobilité".

L'optimalité des routes(Path optimality) :

On remarque de la figure que plus la connectivité augmente plus les durées de vie des routes augmente. L'algorithme s'adapte à la mobilité de l'environnement en stockant plusieurs chemins vers une même destination, ce qui fait que beaucoup de changements de topologie n'auront pas d'effets sur le routage des données, à moins que tous les chemins qui mènent vers la destination soient perdus.

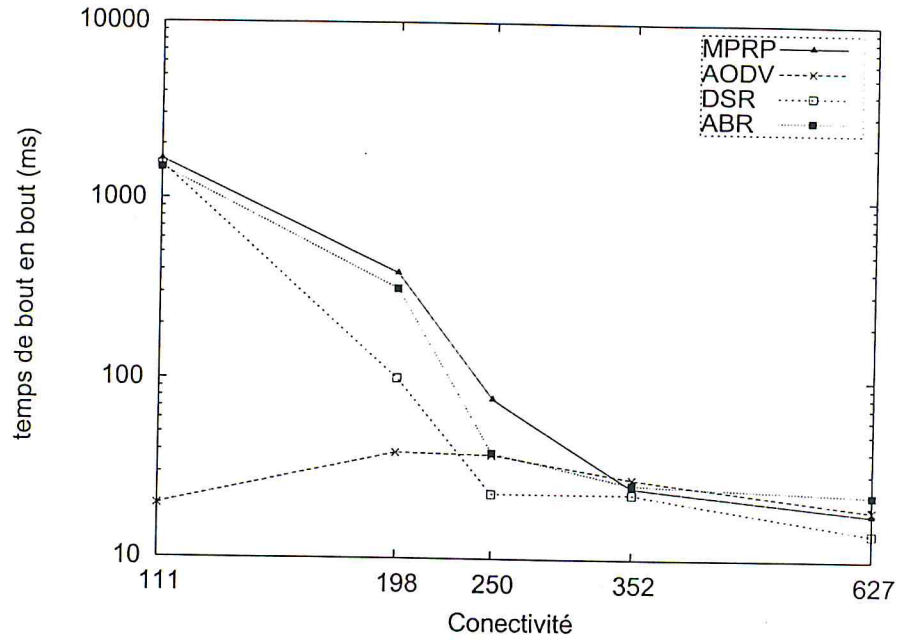


FIG. 6.13 – Le délai moyen de transfert en fonction de la connectivité.

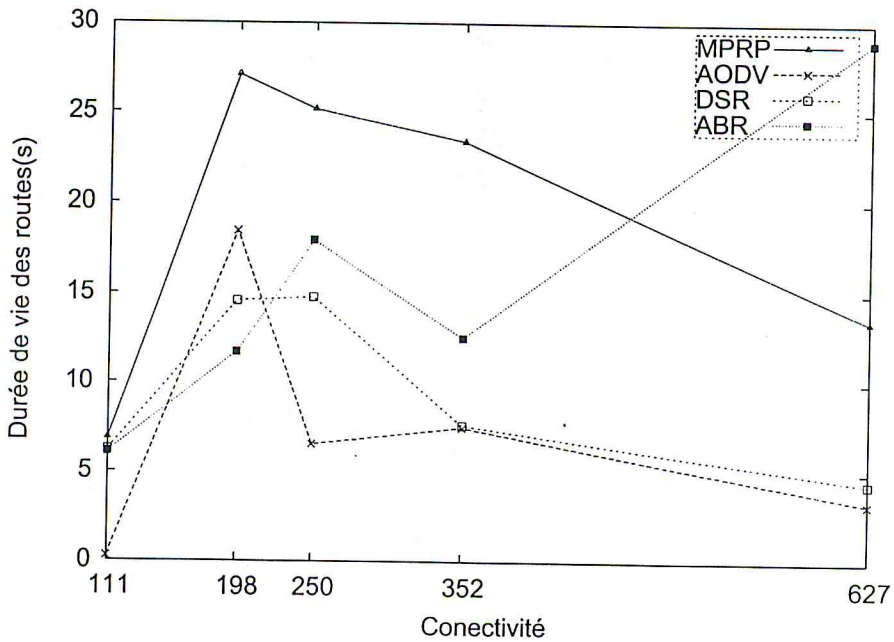


FIG. 6.14 – L'optimalité des routes en fonction de la connectivité.

6.8 Conclusion

Dans ce chapitre, nous avons présenté en premier lieu, l'environnement de simulation dans lequel le protocole est simulé. Puis nous avons présenté et interprété les résultats de simulation de notre protocole avec les autres solutions existantes.
Les résultats de simulations faits dans le simulateur GloMoSim, ont montré que notre protocole a donné de meilleures performances que les autres protocoles. Notamment, le taux de réception de données très élevé et l'utilisation des routes optimales(au sens durée de vie) dans le cas des topologies fortement mobiles ou fortement chargées.
Nous avons vu également que le protocole MPRP génère plus de paquets de contrôle que certains protocoles, parce que le protocole nécessite cette quantité de paquets de contrôle pour son fonctionnement. MPRP montre également que le délai moyen de transfert est relativement (pas dans tous les cas) faible par rapport d'autres protocoles, qui revient aux approches utilisées par MPRP.

Conclusion générale

Le problème de routage dans les réseaux mobiles Ad hoc, est loin d'être évident, Au cours de ces dernières années, beaucoup de travaux de recherche ont été consacrés pour la résolution de ce problème. La plus part des solutions existantes proposées pour remédier aux problèmes de routage ne sont encore que des drafts, et restent en cours de développement et de spécification.

Les protocoles de routage existants peuvent se classer en deux grandes familles de protocoles : Famille de protocoles de routage non prédictifs et famille de protocoles de routage prédictifs. Les études[SU 01][SIN 00]de comparaison via l'outil de simulation, montrent que les protocoles de routage prédictifs sont plus performants que les protocoles de routage non prédictifs.

L'objectif de cette thèse était d'étudier les différent techniques de prédiction et proposer, pour remédier aux problèmes liés de routage, un protocole de routage appartenant à la famille des protocoles de routage prédictifs pour les réseaux mobiles Ad hoc, de telle sorte à répondre à deux besoins principaux : Améliorer la fraction de réception de données et l'utilisation des routes optimales (en termes de durée de vie) dans le transfert de données.
D'après les résultats obtenus de la simulation, notre protocole proposés offre de bonnes performances dans presque tous les scénarios de simulation, notamment les résultats concernant la fraction de réception de données et l'utilisation optimale des routes(notre objectifs).

Généralement, ces deux métriques sont les plus importantes pour un protocole pour qu'il soit performant. Le protocole montre un délai important de transfert que nous en avons référai aux approches utilisées, ce qu'il faut l'optimiser dans des futurs travaux, pour que le protocole soit utilisé dans les applications temps réel qui demandent de temps de bout en bout relativement faible.

En effet, le protocole ainsi développé, offre d'autres avantages que certains protocoles n'en offrent pas. Par exemple, et grâce à la technique des routes disjointes utilisée,

un nombre important des routes est réduit de telle sorte à sauvegarder que les routes les plus optimaux. Les appareils portatifs qui se caractérisent par, des capacités de calcul et de stockage limitées, s'adaptent mieux à notre protocole proposé. D'un côté, parce que le nombre de routes est relativement petit. Donc, mois d'espace mémoire pour le stockage des routes, et d'un notre côté, le petit nombre de routes sauvegardées, implique moins de temps CPU pour les mettre à jour.

Parmi les avantages du protocole MPRP, le plus significatif est le fait que les nœuds intermédiaires n'aient pas besoin de maintenir les informations de mise à jour pour envoyer les paquets de données, puisque ces derniers contiennent toutes les décisions de routage. En outre, dans ce protocole, il y a une absence totale de boucle de routage, car le chemin source-destination fait partie des paquets de données envoyés.
La principale faiblesse de notre protocole est, la grande taille de l'entête du paquet de données : Le protocole doit mettre la séquence complète des nœuds que doit un paquet de données les visiter durant son transfert pour atteindre son nœud destination, ce qui va influer sans doute, sur la fraction de réception de données notamment dans les réseaux à grande échelle.
On peut penser que cette solution est suffisante dans déférentes situations : dans les réseaux à grande mobilité, dans les réseaux trop chargés, etc.

Dans la continuité du travail présenté. Il faudrait par exemple étudier l'influence exacte de l'utilisation des obstacles dans les scénarios de simulation, la version actuelle ne permet pas de réaliser ce genre de scénarios, la version supportant les scénarios des obstacles sera fonctionnelle dans les prochains mois de cette année.
Il serait également intéressant d'utiliser les techniques utilisées par notre protocole de routage unicast dans les protocoles de routage multicast, notamment pour prédire le phénomène de partitionnement des réseaux qui est un problème posé jusqu'à présent et beaucoup de recherche sont faites dans cet axe.

Liste des acronyms

ABR	Associativity Based Routing.
AODV	Ad Hoc On-demand Distance Vector Routing.
BQ	Broadcast Query.
CBRP	Cluster based Routing Protocol.
CGSR	Clusterhead Gateway Switch Routing Protocol.
DREAM	Distance Routing Effect Algorithm for Mobility.
DSDV	Dynamic Destination-Sequenced Distance-Vector routing protocol.
DSR	Dynamic Source Routing.
DV-MP	Distance Vector with Mobility Prediction.
FORP	Flow Oriented Routing Protocol.
FSR	Fisheye State Routing.
GloMoSim	GLObal MObile information system SIMulator.
GPS	Global Positioning System.
GSR	Global State Routing.
HSR	Hierarchical State Routing.
IARP	IntrA zone Routing Protocol.
IERP	IntEr zone Routing Protocol.
IETF	Internet Engineering Task Force.
IP	Internet Protocol.
LAR	Location-Aided Routing.
LCC	Least Clusterhead Change algorithm.
LET	Link Expiration Time.
MANET	Mobile Ad hoc NETwork.
MPRP	Multi Path prediction-based Routing Protocol.
NTF_LOW	NoTiFication LOW.
NTF_STR	NoTiFication STRong.
PARSEC	PARallel Simulation Environment for Complex systems.
PDA	Personal Digital Assistants.
PCL	Parallel Computing Laboratory.
RABR	Route-lifetime Assessment Based Routing protocol.
RERR	Route ERRor.
RET	Route Expiration Time.
RREP	Route REPly.
RREQ	Route REQuest.
SSA	Signal Stability routing Adaptive.
TCP	Transmission Control Protocol.
TORA	Temporally Ordered Routing Algorithm.
UCLA	University of California Los Angeles.
VT	Visualization Tool.
WRP	Wireless Routing Protocol.
ZHLS	Zone-based Hierarchical Link State Routing Protocol.

Bibliographie

[BAD 02] Badache N.,Djenouri D., Derhab A., Lemlouma "Revue d'information scientifique et technique "Les protocoles de routage dans les réseaux mobile ad hoc" 2002.

[BAG 98] R. Bagrodia, R. Meyer et al., "PARSEC : A Parallel Simulation Environment for Complex Systems," Computer, Vol. 31(10), October 1998, pp. 77-85.

[BAJ 00] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, M. Gerla, "GloMoSim : A Scalable Network Simulation Environment", Proceedings of the 12th Workshop on Parallel and Distributed Simulations - PADS '98, May 26-29, 1998 in Banff, Alberta, Canada.

[BAS 98] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B, A.. Woodward. "A distance routing effect algorithm for mobility(DREAM) ." In processing of ACM/IEEE MOBICOM'98, Dallas, Texas, pp 76-84 October 1998.

[BAS 03] Claudio Basile, Marc-Olivier Killijian, David Powell "A Survey of Dependability Issues in Mobile Wireless Networks" February 21, 2003 Technical Report, LAAS CNRS Toulouse, France 2003.

[CHE 03] Robin Chellappa, Andrew Jennings, Nirmala Shenoy, "Route Discovery and Reconstruction in Mobile Ad hoc Networks", 11th IEEE International Conference on Networks, Sydney Australia, Sept 28 to 1st Oct. 2003.

[CHE 98] GER T.-W.Chen and M.Gerla, "Global State Routing : A New Routing Scheme for Ad hoc Wireless Networks," In Proceedings of IEEE ICC'98, Atlanta, GA, pages 171-175, Jun. 1998.

[CHI 97] C.C.Chiang, H-K Wu, Winston Liu, and Mario Gerla." Routing in cluster multihop, mobile wireless networks". The IEEE Singapore International Conference on Networks, pp 197-211, 1997.

[DAV 01] David B. Johnson David A. Maltz Josh Broch "DSR : The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks". in Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.

[DUB 96] Rohit Bube Cynthia D. Rais Kuang-Yeh Wang Satish K. Tripathi "Signal Stability based Adaptive Routing(SSA) for Ad-Hoc Mobile Networks". December 4, 1996.

[FRO 00] Magnus Frodigh, Per Johansson and Peter Larsson "Wireless ad hoc networking, The art of networking without a network". Ericsson Review No. 4, 2000, pp 248-263

[HAA 97] Zygmunt J. Haas, Cornell University and Marc R. Pearlman, Cornell University INTERNET-DRAFT "The Zone Routing Protocol (ZRP) for Ad Hoc Networks" ¡draft-zone-routing-protocol-00.txt¿, November 1997

[ITA 01] atsushi Itawa, Ching-Chuan Chiang, Guangyu Pei, Mario Gerla and Tsu-wei Chen "A Mobility Framework for Ad Hoc Wireless Networks". Mobile Data Management 2001 : 185-196 Computer science department University of California, Los Angeles

[JOH 94] David B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts", Proceedings of the Workshop on Mobile Computing Systems and Applications, IEEE Computer Society, Santa Cruz, CA, pages 158-163,December 1994.

[KLE 91] L. Kleinrock and K. Stevens, "Fisheye : A Lenslike Computer Display Transformation," Technical report, UCLA, Computer Science Department, 1991.

[KO] Young-Bae Ko and Nitin H. Vaidya "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks" Department of Computer Science Texas A&M University College Station, TX 77843-3112.

[LAR 98] Tony Larsson, Nicklas Hedman MASTER'S THESIS "Routing protocols in Wireless Ad-hoc Networks - A Simulation study "Stockholm, 1998 Lulea University of technology.

[LEU 01] Leung. R, Jilei Liu, Poon. E, Chan A. L. C, Baochun Li, "MP-DSR : a QoS-aware multi-path dynamic source routing protocol for wireless ad-hoc networks" , 26th Annual IEEE Conference on Local Computer Networks, 2001.

[LOA 99] M. Loa-Ng and I-T. Lu. "A peer to peer zone based two-level link state routing for mobile ad hoc networks". IEEE journal on selected Areas in Communications. Special issue on Ad-hoc networks, pp 1415-1425, August 1999.

[MUE 04] Stephen Mueller, Rose P. Tsang, and Dipak Ghosall "Multipath Routing in Mobile Ad Hoc Networks : Issues and Challenges" Invited paper in Lecture Notes in Computer Science, Edited by Maria Carla Calzarossa and Erol Gelenbe, 2004.

[MUR 96] Shree Murthy & J.J. Garcia-Luna-Aceves "An efficient routing protocol for wireless networks" ACM Mobile Networks and Applications 1996. pp 183-197.

[PAR 97] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in IEEE Infocom,1997.

[PEI 99] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang, "A Wireless Hierarchical Routing Protocol with Group Mobility," In Proceedings of IEEE WCNC'99, New Orleans, LA, Sep. 1999.

[PER 99] Charles E. Perkins, Elizabeth M. Royer and Samir R. Das "Ad Hoc On-Demand Distance Vector (AODV) Routing" INTERNET DRAFT draft-ietf-manet-aodv-04.txt 22 October 1999 Mobile Ad Hoc Networking Working Group

[PER 94] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," ACM SIGCOMM, October 1994.

[ROS 00] Gail Rosen, Amin Atrash "Ad Hoc Networking Extended Research Project",7 june 2000

[ROY 02] E. Royer and C.-K. Toh "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks" vol. 6, no. 2, page. 48 51 date 2002/12/12. source : IEEE Personal Communication April 1999.

[SCH 02] Jan Schaumann "Analysis of the Zone Routing Protocol" http ://www.netmeister.org/misc/zrp/zrp.pdf, December 8, 2002.

[SIN 00] Jatinder Pal Singh, Ashish Ahuja and Sulabh Agarwal, "Link Connectivity Assessment Based Applications for Mobile Ad-hoc Networks", 42nd Annual Technical Convention of The Institution of Electronics and Telecommunication Engineers, September 2000, New Delhi, India.

[STE 95] M. Steenstrup, editor. Routing in Communications Networks. Prentice Hall, Inc, Englewood Cliffs, NJ, 1995.

[SU 99] W. Su, "Motion Prediction in Mobile/Wireless Networks" PhD Dissertation, UCLA Computer Science Department, Los Angeles, CA, Dec. 1999.

[SU 01] William Su, Sung-Ju Lee, and Mario Gerla "Mobility prediction and routing in ad hoc wireless networks" INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT, Int. J. Network Mgmt 2001 ; 11 :3-30.

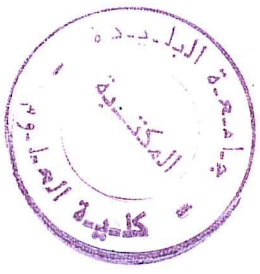
[TOH 97] C.-K. Toh, "Associativity-Based Routing For Ad Hoc Mobile Networks," Wireless Personal Communication Journal, Special Issue On Mobile Networking and Computing Systems, Kluwer Academic Publishers, vol 4, no 2, Mar. 1997, pp. 103-109

[TOH 02] C.K. TOH "Ad Hoc Mobile Wireless Networks : Protocols and systems" Edition Prentice Hall PTR NEWJERSY

[UNE 02] Nil Unerdem. "Dynamic Source Routing and Associativity-Based Routing" E6768, Professor Maxemchuk May 2, 2002 http ://comet.ctr.columbia.edu/ jaekwon/E6768/prschedule.htm.

[ZHU 02] Yufang Zhu "PRO-ACTIVE CONNECTION MAINTENANCE IN AODV AND MAODV " A thesis submitted to the Faculty of Graduate Studies in partial fulfillment of the requirement for the degree of Master of Science in Information and Systems Science Carleton University, Ontario, Canada, August 2002.

[ZOU 04] Xukai Zou, Byrav Ramamurthy and Spyros Magliveras "Routing Techniques in Wireless Ad Hoc Networks Classification and Comparison" International Conference on Computing, Communications and Control Technologies : CCCT'04.



Annexe A

Installation de GloMoSim

A.1 Installation sous Windows (XP , NT ,2000 pro)

Pour un bon fonctionnement GloMoSim exige que les logiciels suivants soient installer dans le PC dont lequel en va installer le simulateur :

1. Microsoft VC++ version 6.0 (**Essentiel**) ;
2. JAVA JRE version 1.2 ou plus (**Optionnel** : pour l'outil de visualisation) ;
3. JAVA SDK version 1.2 ou plus (**Optionnel** : pour l'outil de visualisation).

En plus des logiciels précités précédemment, l'utilisateur veut installer le simulateur doit posséder le droit de définir (ajouter) d'autres variables d'environnement.

1. Après avoir téléchargé[1] le simulateur GloMoSim 2.03, décompresser et sauver le dans un répertoire (Par exemple "C :/glomosim/glomosim")
2. Aller dans le sous répertoire "glomosim/Parsec", ouvrer le dossier "windows-nt-vc6.0", et copier tous ses fichiers dans le répertoire Parsec, puis Supprimer tous les autres fichiers (car ils sont pour d'autres systèmes d'exploitation)
3. ajouter une autre variable d'environnement comme suit :
   (a) Post de travail -> Propriétés -> Avancé -> Variables d'environnements.
   (b) Nouveau "Nom de la variable= PCC_DIRECTORY", Valeur de la variable="C :/glomosim/parsec".
   (c) Ajouter dans la variable PATH la phrase suivante "C :/glomosim/parsec/bin".
4. Mettre le chemin de Visual C++ 6 dans les variables PATH, INCLUDE et LIB.

[1] Vous pouvez télécharger ce logiciel à partir du site `http://pcl.cs.ucla.edu/projects/glomosim/`

(a) Set include : "C :/Program Files/Microsoft Visual Studio/VC98/MFC/Include ; C :/Program Files/Microsoft Visual Studio/VC98/Include".

(b) Set lib : "C :/Program Files/Microsoft Visual Studio/VC98/MFC/Lib ; C :/Program Files/Microsoft Visual Studio/VC98/Lib".

(c) Set path : "C :/Program Files/Microsoft Visual Studio/Common/MSDev98/Bin".

5. En principe PARSEC est correctement installé, pour vérifier ça, taper sous DOS la commande pcc, s'il y a un message qui vous dit " No input files " c'est bon. Il reste l'installation du GloMoSim.

6. Installation du GloMoSim :

   (a) Aller dans le sous répertoire "glomosim/glomosim/main", exécuter le fichier batch "makent.dat"

   (b) Normalement un fichier exécutable appelé "glomosim.exe" est crée dans le sous répertoire "glomosim/glomosim/bin", exécuter le à partir de DOS en tapant la commande "glomosim config.in".

   (c) Un fichier "glomo.stat" est généré à la fin de la simulation, comparer le avec "glomo.stat.sample".

7. Compilation de l'outil de visualisation de GloMoSim (Visualisation tool). Aller dans le répertoire "C :/glomosim/glomosim/JAVA_GUI", taper sous DOS "javac *.java", entrer.

8. GloMoSim avec ses accessoires sont mentant prêts à utiliser.

A.2 Installation sous Linux (Redhat 7.3)

De même pour les utilisateurs Linux, ils doivent posséder le droit de définir (ajouter) d'autres variables d'environnement. Pour que GloMoSim fonctionne correctement sous Linux, il faut que les logiciels j2sdk 1.2 et JRE 1.2 (ou plus) soient installés dans le PC dans lequel l'utilisateur va installer le simulateur.

Remarque : Le compilateur C++ est installé par défaut dans cette version du Linux, il ne vous est pas nécessaire de le réinstaller.

Une fois le téléchargemment du fichier "glomosim-2.03.tar.gz" est effectué, sous un terminal Linux, décompressé le fichier téléchargé en tapant la commande suivante :
"tar xvf glomosim-2.03.tar.gz", cette commande va crée le répertoire"glomosim-2.03"

dans le dossier où il se trouve le fichier compressé.
Le simulateur GloMoSim exige que le langage PARSEC soit correctement installer, pour faire ceci, on suit les étapes suivantes :
L'utilisateur doit créer un sous répertoire dans "/usr/local/parsec" et copier tous les fichiers se trouvant dans "./parsec/redhat-7.2/" dans le sous répertoire crée précédemment. Une fois cela est fait, l'utilisateur doit ajouter le chemin du PARSEC à la variable d'environnement en tapant la commande suivante dans le Shell :
"setenv PCC_DIRECTORY /usr/local/parsec". Puis mettre le chemin des fichiers binaires dans la variable PATH comme suit :
"set /usr/local/parsec/bin".
Une fois l'installation du PARSEC est bien effectuer, l'étape d'installation du GloMoSim est lieu est peut être fait comme suit :
En utilisant un terminal Linux, Aller dans le sous répertoire "./glomosim/main" et taper la commande "make", Quand la compilation est réussie, un fichier exécutable "glomosim.exe" est crée dans le sous répertoire "./glomosim/bin". pour vérifier si le simulateur est bien installer taper la commande :"./glomosim config.in". Normalement vous allez trouverez un fichier de sortie appelé "glomo.stat." dans le répertoire bin. Comparer ce fichier avec celle qu'existe par défaut dans GloMoSim "glomo.stat.sample".

A.3 Outil de visualisation du GloMoSim

Le simulateur peut fonctionner avec ou sans l'outil de visualisation(VT : Visualization Tool). Il suffit d'utiliser la ligne de commande en cas ou l'utilisateur veut exécuter le simulateur sans le VT, si l'utilisateur veut que l'exécution soit effectuée avec le VT, il faut utiliser le GUI fourni par GloMoSim(voit la figure A.1), Via cet outil de visualisation l'utilisateur peut arrêter, reprendre et exécuter pas à pas la simulation, il peut également augmenter, diminuer et suivi les déplacements de tous les nœuds pendent toute la période de simulation, Pour plus d'informations sur cet outil consulter l'aide fournie par le simulateur GloMoSim concernant le VT.

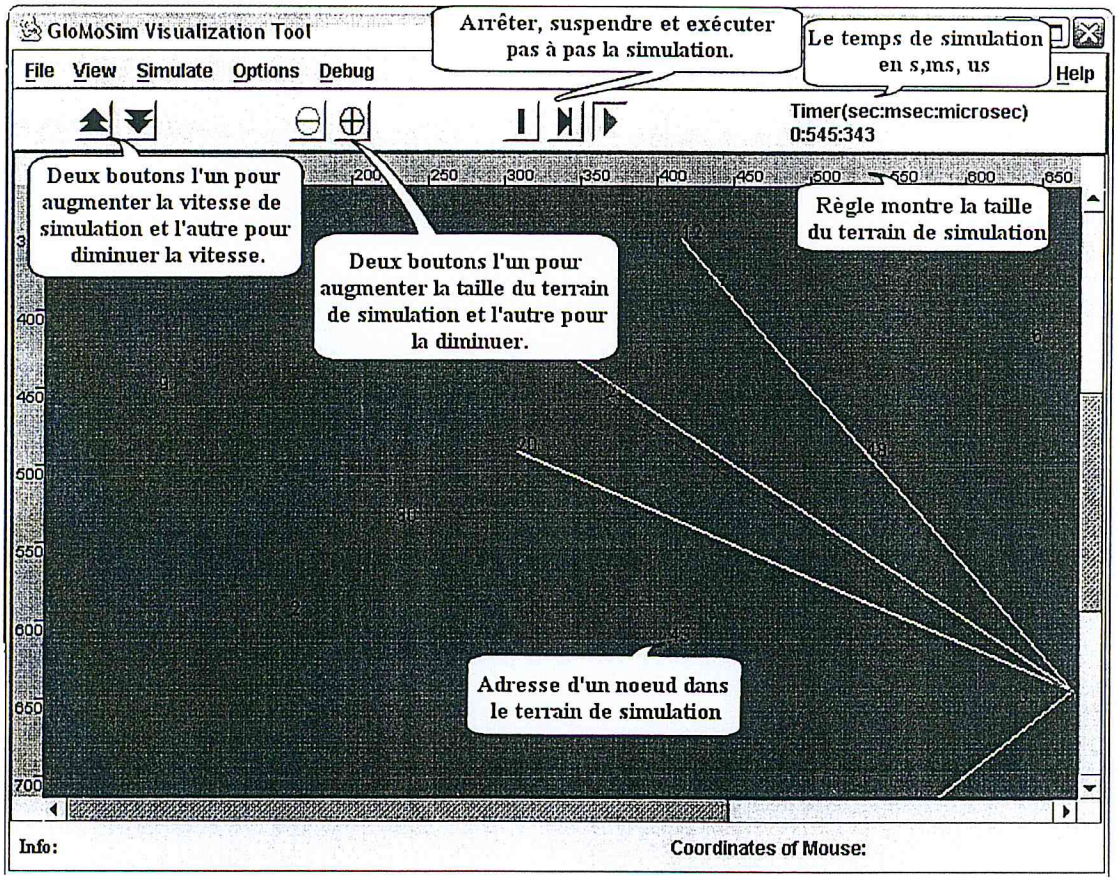


FIG. A.1 – Outil de visualisation du GloMoSim.