N° d'ordre 30/ 2012 – M/ Info

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'électronique et informatique

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTERE

EN INFORMATIQUE

Spécialité : Réseaux Mobiles Ad hoc

Par
DIB Chehrazed

SUJET

Le Protocole de Routage Multicast Ad Hoc MZRP-MPR

Soutenue publiquement le 03 Juillet 2012 Devant le jury composé de :

Dr. S. MOUSSAOUI	Maître de conférence /A à l'USTHB	Présidente
Pr. N . BADACHE	Professeur à l'USTHB	Directeur de mémoire
Dr. N. NOUALI	Maître de Recherche à l'USTHB	Examinatrice
Dr. C. BENZAID	Maître de conférence / B à l'USTHB	Invitée

Dédicace

A l'âme du mon père

A ma très chère mère

A mon mari Abderrahmane

A mon adorable fils Abdeldjalil

A toute ma famille

A toutes mes amies spécialement : Radia, Kahina et Lila.

Je leur dédie ce modeste travail.

Remerciement

Ces quelques lignes ne pourront jamais exprimer la reconnaissance que j'éprouve envers tous ceux qui, de prés ou de loin, ont contribué par leurs conseils, leurs encouragements ou leurs amitiés à l'aboutissement de ce travail.

Mes vifs remerciements accompagnés de toutes mes gratitudes vont tout d'abord à mon directeur de mémoire le Pr. BADACHE Nadjib, Directeur Général du CERIST et Professeur à l'Université des Sciences et de la Technologie Houari Boumediene (USTHB), pour m'avoir proposé cet intéressant sujet et pour les précieux conseils et orientations qu'il m'a prodigué. Je le remercie pour sa disponibilité, son aide, ses critiques constructives et enfin pour ses explications et suggestions pertinentes.

Mes sincères remerciements vont également à mon oncle le Pr. MEHENNI Mohamed, Professeur à l'Ecole Nationale Polytechnique (ENP), pour son aide en me mettant en contact avec le Dr. SAADOUNE Rabah, Professeur à l'ENP, que je le remercie pour son aide, sa méthode de travail, sa disponibilité, ces précieux conseils et sa contribution dans ce travail. Encore merci pour sa gentillesse.

Une mention très spéciale pour Mr LAOUITI Anis , Docteur en Informatique à l'INRIA, et Mr ROMDHANI Imad, Ph.D au Centre de Recherche de Motorola à Paris, qui m'ont beaucoup aidé, de loin, à enrichir mes connaissances dans le domaine des communications sans fil, et des réseaux ad hoc. Merci pour leurs aides, leurs conseils, leurs orientations et leurs remarques pertinentes.

J'aimerai remercier également tous les membres de jury, le président Mm S. MOUSSAOUI et les examinateurs Mm N.NOUALI et M^{elle} C.BENZAÎD.

Un grand merci à ma mère et à mon époux qui ont été toujours derrière moi pour m'encourager et me soutenir durant toutes les années de ce travail; encore merci.
Merci à mes sœurs et mes frères, en particulier Sabrina et Kamel.
Merci à mon cousin DIB Djamel-eddine, de la bibliothèque du CERIST.

Merci à toutes mes amies, en particulier Radia et Kahina.

Table des matières

INTRODUCTION GENERALE

Le monde des réseaux sans fil a connu un développement vertigineux ces dernières années. Son succès est dû essentiellement à la chute de son coût grâce à la vulgarisation des appareils sans fil ainsi qu'à la miniaturisation des composants électroniques. Un autre facteur qui a participer à ce développement rapide est l'autonomie des appareils avec batterie qui ne cesse de croître.
En particulier, le domaine des réseaux locaux sans fil ad hoc suscite de plus en plus d'intérêt depuis ces dernières années. La particularité de ce type de réseau est qu'il n'a besoin d'aucune installation fixe à l'inverse d'autres types de réseaux (comme par exemple le GSM). Donc, il est facile et rapide à déployer. De plus, les différents composants de ce réseau sont libre de se mouvoir. Ceci résulte en une topologie dynamique susceptible de changer (contrairement, par exemple, aux réseaux satellite) d'une façon imprévisible.

Lorsque les participants n'arrivent pas à s'entendre directement, des nœuds intermédiaires peuvent jouer le rôle des relayeurs par un routage interne. Ce routage interne devient plus complexe lorsque les nœuds bougent. Les protocoles de routage classiques qui s'appliquent aux réseau filaires deviennent inefficaces. D'où la nécessité de créer de nouveaux protocoles qui répondent aux nouveaux besoins et qui prennent en compte les nouveaux paramètres (mobilité, liens asymétriques, nœuds cachés,..).

Dans ce type de réseau, tous les nœuds sont libres de bouger et de se déplacer sans aucune contrainte. Néanmoins, cette nouvelle liberté doit faire face à de nouveaux problèmes et défis : optimisation de la bande passante, assurance de la fiabilité et de la robustesse de routage, offrir une bonne connectivité entre les nœuds mobiles, …etc.
Ce sont ces problèmes qui motivent tous les auteurs de ce domaine pour concevoir ou améliorer des protocoles de routages unicast ou multicast pour ce type de réseau.

Dans ce mémoire je me suis intéressé au routage multicast. Mon travail consiste à améliorer un protocole de routage multicast existant, qu'est le MZRP, par l'introduction de la technique de diffusion optimisée MPR. Le but de ce travail est de générer un protocole de routage multicast optimisé MZRP-MPR, permettant l'optimisation de la bande passante et des délais de transmissions par conséquent.

Ce mémoire est constitués de cinq chapitres organisés comme suit :

- Le premier chapitre intitulé « Les réseaux mobiles ad hoc et le problème de routage », représente une introduction au réseaux sans fil, et plus précisément aux réseaux ad hoc. Dans ce chapitre, quelques protocoles de routage unicast ont été présenté.

- Le second chapitre intitulé « Le routage multicast dans les réseaux mobiles ad hoc » explique le principe de routage multicast au seins de ces réseaux ; ainsi que les différents problèmes rencontrés lors de conception et d'utilisation de ce routage multicast.

- La troisième chapitre intitulé « Quelques protocoles multicast dans les réseaux ad hoc » est consacré à expliquer en détail le fonctionnement de plusieurs protocoles multicast. Vers la fin une comparaison entre ces différents protocoles a été donnée sous la forme d'un tableau.

- Le quatrième chapitre intitulé « Multicast Zone Routing Protocol (MZRP) et MultiPoint Relay (MPR) » décrit en détail le protocole de routage multicast MZRP depuis sa version unicast ZRP ; il décrit aussi la technique de diffusion optimisée MPR. La fin de ce chapitre est consacrée à la présentation du nouveau protocole proposé dans cette thèse qu'est le MZRP-MPR.

- Le cinquième et dernier chapitre donne une analyse qualitative à ce nouveau protocole MZRP-MPR ; le début du chapitre décrits les étapes d'une simulation de la technique MPR ainsi que ses résultats.

Chapitre 1

Les réseaux mobiles ad hoc et le problème de routage

1.1 Les réseaux ad hoc

Un réseau ad hoc est une collection de deux ou plusieurs outils de communication mobiles, dites aussi nœuds mobiles, chacun équipé d'une ou de plusieurs interfaces de communication sans fil. Ils sont aussi communément nommés *MANet* pour *Mobile Ad Hoc Network*. Ces réseaux ad hoc se caractérisent par le fait que les communications entre les nœuds du réseau ne bénéficient d'aucune infrastructure préexistante ou d'appoint (balise, station centrale, site fixe, borne, relais). Un réseau ad hoc doit être facilement déployable, les nœuds pouvant joindre et quitter le réseau de façon totalement dynamique sans devoir en informer le réseau et si possible sans effet de bord sur les communications des autres membres.

1.1.1 Définition

Le terme « ad hoc » est une locution d'origine latine qui signifie « qui convient au sujet, à la situation. » je parle donc de réseaux auto-adaptatifs (capables de s'organiser par eux même). Une autre lecture de la définition peut signifie une propriété d'universalité de ce moyen de communication, comme si ce procédé peut satisfaire tous les besoins en terme de communication entre objets mobiles. La définition la plus concise d'un réseau ad hoc est la suivante : « un réseau ad hoc est un réseau sans fil à contrôle totalement décentralisé.» Dans les réseaux sans fil, toutes les communications entre les nœuds se font par voie aérienne, une interface radio dans chaque mobile permettant d'émettre et de recevoir. Le terme de contrôle décentralisé indique que l'ensemble des opérations pour la découverte et la maintenance du réseau se fait localement par chaque nœud. Chacun gère ses communications à partir des informations connues sur le réseau et sur l'état interne du nœud, dans le but de faire émerger un comportement cohérent du réseau (*e.g*. des routes correctement établies) [1].

En plus de ces deux premiers concepts (le concept de la communication sans fil et celle de contrôle décentralisé) les réseaux ad hoc peuvent inclure aussi le concept de la mobilité. Les nœuds du réseau peuvent alors se déplacer à l'intérieure du réseau, seul ou avec un groupe. La mobilité des nœuds entraîne le changement de la topologie du réseau. Et afin d'assurer la continuité des communications et l'échange d'informations et des services lors de déplacement des nœuds, chacun de ces derniers doit mettre à jour certaines informations et conserver dans un état valide les routes qui commencent, passent ou finissent par lui-même [2].

1.1.2 Applications

Les recherches sur les réseaux ad hoc ont été initiées par le DARPA (*Defence Advenced Research Project Agency*) avec le développement de *PNR* ou *Packet Radio Networks* [3]. Ce protocole, conçu pour l'armée américaine, permet de déployer une infrastructure de communication entre chaque bataillon, par l'intermédiaire de plusieurs véhicules communiquant ensemble. Le fait que ce soit les militaires qui aient commencé les premières expériences sur les réseaux ad hoc n'est pas un hasard.

En effet, cette infrastructure est très adaptée aux environnements « hostiles », car ils sont rapidement déployables, et robustes dans le cas de perte de liens. Ils sont donc particulièrement intéressants pour un système de communication sur les champs de bataille mais aussi dans d'autres environnements. On peut évoquer les sinistres (tremblements de terre, inondations) où l'ensemble des infrastructures existantes a été détruit [4].

Les réseaux ad hoc peuvent aussi utilisés pour relier plusieurs ordinateurs entre eux. Ils sont donc adaptés pour la formation des réunions, où la nécessité temporaire d'une infrastructure de communication est soutenue par l'ensemble des participants. Il existe aussi des recherches proposant d'utiliser les réseaux ad hoc avec les véhicules routiers [5]. On peut entrevoir de nombreuses applications possibles pour un tel usage : distribution d'information au niveau local (risques d'accidents ou d'encombrements), aide automatique à la conduite (feux d'avertissement), téléphonie entre véhicules, etc.

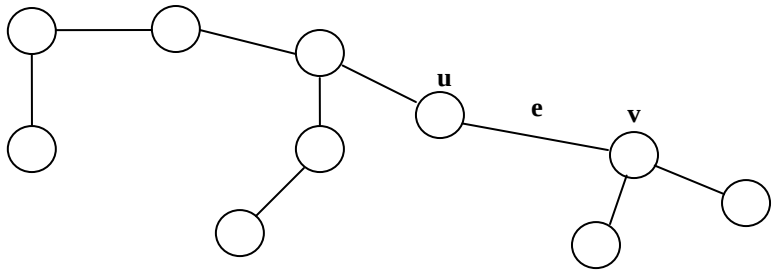
Un autre domaine très intéressant pour les réseaux ad hoc concerne les senseurs [6]. Ce sont des équipements possédant des capacités très limitées (mémoire, processeur, bande passante) et de taille réduite. Ces équipements ont de nombreux domaines d'application : médicale ou militaire par exemple. Ils sont en générale utilisés en grandes quantités, et les réseaux ad hoc permettent alors la liaison entre tous les objets. On peut citer l'exemple des capteurs météorologiques, de surveillance d'un site ou de mesure des constantes d'un être humain.

1.2 Modélisation d'un réseau ad hoc

Usuellement, un réseau de communication est représenté dans sa forme la plus générale par un graphe orienté. Les sommets représentent les routeurs et les arcs modélisent la possibilité d'établir une communication directe entre deux sommets. S'il existe un arc du sommet u vers le sommet v, on dit que le sommet u est un voisin de v tans dit que v est accessible depuis u. Le médium radio étant diffusant par nature, lorsqu'un nœud u diffuse un paquet, tous les nœuds qui sont accessibles depuis u reçoivent une copie de ce message. Les transmissions radio interférent forcément, un nœud v ne peut recevoir un message que si exactement un seul de ses voisins émet. S'il s'avère que plusieurs voisins émettent en même temps, il y a une collision et les messages arrivent bruités et sont inexploitables par le nœud v. Ces contraintes sont proches des modèles de communication proposés pour les réseaux de communication [7]. Il s'agit d'un modèle half-duplex, un port en réception et Δ port en émission.

Néanmoins, modéliser un réseau ad hoc par un graphe simple orienté pose, n'ont pas un réel problème de modélisation car on vient d'exposer les contraintes qu'il faut respecter à tout instant mais plus un problème de représentation. La contrainte Δport implique que pour chaque envoi de message, l'ensemble des sommets accessibles reçoit une copie de message. Guillaume Chelius et Eric Fleury propose dans leur article « *Critère d'évaluation d'arbres multicast ad hoc* » [8], un nouveau modèle de représentation pour les réseaux ad hoc : il s'agit d'utiliser les hypergraphes à la place des graphes simples [9].

Dans notre cas, un réseau va être modélisé par un graphe $G_t = (V_t,U_t)$ où V_t représente l'ensemble des nœuds mobiles qu'existe dans le réseau de communication à l'instant t, et E_t représente l'ensemble des liens radios reliant l'ensemble des nœuds mobiles V_t au même instant t (voir la figure 1.1). Si le lien e= (u, v)$\in$ E_t, on dit alors que les noeuds mobiles u et v sont en mesure de communiquer directement à l'instant t. Cette combinaison peut devenir inexistante à un instant t' car la mobilité d'un ou de plusieurs nœuds du réseau peut changer l'architecture du réseau.



○ : N*œud mobile* $\in$ V_t ——— : Lien de communication radio $\in$ E_t

FIG 1.1 *– Modélisation d'un réseau Ad Hoc*

1.3 L'hétérogénéité des nœuds mobiles

Un réseau ad hoc peut supporter des outils de communication de différentes formes (par exemple : Palmtops, Laptop, téléphone mobile, etc.). Par conséquent, la capacité de calcul, de stockage et d'énergie de ces outils diffère aussi. Chaque nœud du réseau ad hoc doit être capable de détecter et de vérifier la présence de ses voisins. En plus, et à cause de l'hétérogénéité des outils communiquant, ce premier doit être aussi capable d'identifier le type et les attributs correspondant à chaque nouveau voisin détecté.

Comme le réseau ad hoc ne compte sur aucune infrastructure fixe ou centralisée, et comme les nœuds du réseau sont mobiles, toutes les informations que regroupe chaque nœud sur ses voisins vont être échangées au maximum entre les nœuds qui restent dans le réseau. Ces informations vont réfléchir l'état des liens après chaque changement du réseau pour les utiliser en cas de besoin (en tenant compte des divers caractéristiques des outils, la consommation d'énergie représente un facteur critique dans ce cas) [10].

1.4 Les différentes formes de communication sans fil

Les communications sans fil peuvent prendre différentes formes. Pour une paire de nœuds mobiles, la communication entre eux peut durée une certaine période du temps jusqu'à ce que la session se termine ou l'un des deux nœuds change de position. Cette forme de communication s'appelle la communication *point-à-point* (ou *peer-to-peer* en anglais). Une deuxième forme de communication est celle dont deux ou plusieurs nœuds mobiles communiquent entre eux et se déplacent en groupe. Cette communication s'appelle la communication *distance-à-distance* (ou *remote-to-remote* en anglais), et occupe beaucoup plus de temps.
La dernière forme de communication et celle dont plusieurs nœuds communiquent entre eux d'une manière non cohérente et bougent en toute liberté. La session de communication de cette forme est très petite (des coupures très rapprochées), brusque et indéterministe.

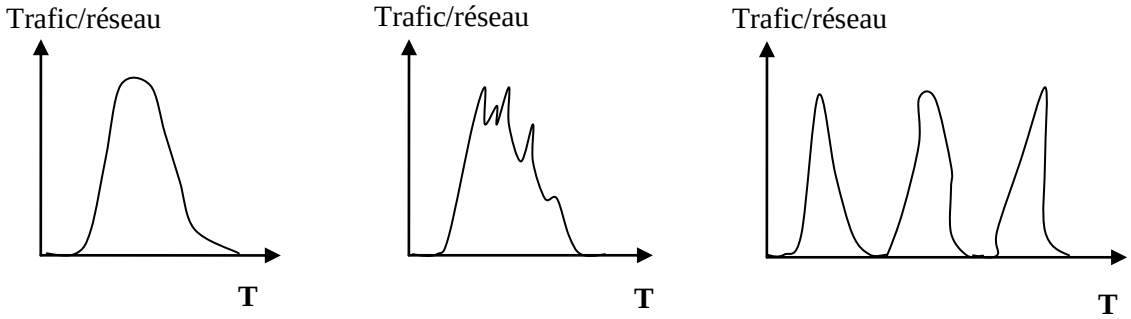


***FIG 1.2** – Le trafic correspondant aux différentes formes de communication sans fil*

1.5 Les différents types de mouvements des nœuds mobiles

Dans ce paragraphe, je vais présenté quelques types de mouvements des nœuds mobiles pouvant affecter la validité des routes d'acheminement de données.

1.5.1 Mouvement des nœuds de route

Dans un réseau ad hoc, une route d'acheminement comprend le nœud « source », le nœud « Destination » et /ou un certain nombre de nœuds intermédiaires. Le mouvement de n'importe quel nœud de cet ensemble affecte directement la validité de la route d'acheminement qui les engendre. Le mouvement de nœud source sera détecté par ses successeurs directs (qui partagent avec lui la même porté radio). Par conséquent, tous les nœuds successeurs du réseau vont être informés des liaisons invalides et omettront par la suite les entrés correspondantes à ces liaisons. La même chose arrive avec le nœud destination, mais dans le sens contraire. Ce sont donc les prédécesseurs directes de la destination qui vont être informés de déplacement de nœud destination, et qui vont par la suite omettre les entrés correspondantes aux liaisons en question. En ce qui concerne le mouvement des nœuds intermédiaires, les opérations de mise à jour sont similaires à celles causées par le mouvement de la destination et de la source à la fois.

Tous ces mouvements ont poussé à concevoir beaucoup de protocoles de routage conventionnels, permettant de répondre avec sympathie sur les problèmes qui se posent lors de changement des liens de communication, causés par la mobilité des noeuds. Généralement, ces protocoles proposent de mettre à jour l'état de tous les nœuds restants dans le réseau, ce qui résulte en la circulation d'une information de routage consistante dans le réseau. Cependant, cette mise à jour inclus la diffusion ou l'inondation du réseau par cette information de routage. Ce qui provoque la sous utilisation de la bande passante et l'augmentation du trafic de signalisation à travers le réseau. D'où, la nécessité de concevoir d'autres protocoles de routage répondant sur ces différents problèmes.

1.5.2 Mouvement des nœuds des liaisons

Dans un réseau ad hoc, un nœud peut jouer le rôle d'un « pont » liant deux ou plusieurs sous ensembles de nœuds mobiles. Le mouvement de ce nœud pont peut fragmenter le réseau en plusieurs sous ensembles de taille plus petite. Parmi les sous ensembles résultant, un seul sous ensemble peut continuer ses communications en gardant les mêmes routes, il s'agit de celui engendrant la source et la destination à la fois.

D'un autre côté, on peut trouver certains nœuds dans le réseau dont le mouvement peut fusionner deux ou plusieurs sous ensembles pour en avoir un de taille plus grande. Ici, l'algorithme de routage accepte cette nouvelle situation ou architecture en mettant à jour les tables de routage de tous les nœuds. Cependant, c'est une solution non efficace aussi. Un système de routage plus efficace doit renoncer cette solution (une mise à jour totale), et essai de mettre à jour uniquement les tables de routage des nœuds affectés par le fusionnement des sous ensembles [10] (voir la figure ci-dessous).

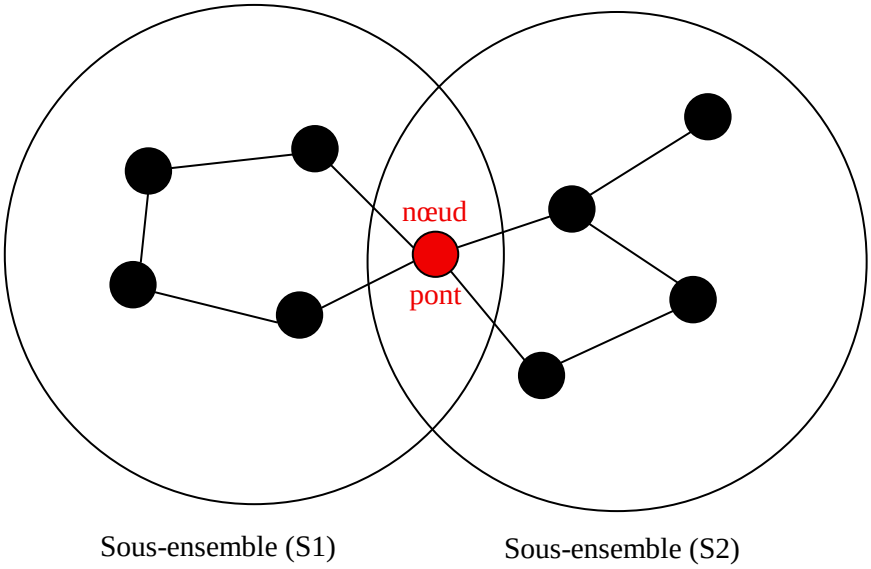


FIG 1.3 *– Fragmentation et fusionnement des sous-ensembles d'un réseau ad Hoc dûe à la mobilité du nœud pont*

1.5.3 Mouvement des nœuds concurrents

Un autre type de mouvement peut exister entre les nœuds du réseau que ça soit la source, la destination ou les nœuds intermédiaires. Il s'agit d'un mouvement concurrent entre ces différents types de nœuds. Ici, il est nécessaire d'avoir certaines règles permettant d'assurer la consistance du réseau quand plusieurs reconfiguration ou réparations s'effectuent sur ce dernier (l'efficacité et la robustesse des nouvelles routes). Un processus pareil doit converger quand la majorité des reconfigurations de routes sont achevées (la mobilité des nœuds diminue).

1.6 L'accès à la couche MAC dans les réseaux ad hoc

Dans un réseau sans fil, deux nœuds mobiles peuvent communiquer en émettant des ondes radios. Ils se partagent un medium unique mais ne peuvent émettre en même temps. En effet, deux émissions simultanées sur le même canal entraînent une perte d'information à l'intersection des deux zones de communication (il s'agit de collision).

Dans un réseau ad hoc, l'accès au médium doit être effectué d'une façon distribuée, à travers un protocole de contrôle d'accès au medium (en anglais, MAC pour Medium Access Control protocol). Contrairement aux réseaux cellulaires, les réseaux ad hoc n'ont pas une station fixe ou centrale que l'on peut compter sur elle. D'où, les systèmes TDMA et FDMA ne sont pas appropriés à ce genre de réseau. En plus, beaucoup de protocoles MAC ne tiennent pas compte de la mobilité des nœuds. Donc, un protocole MAC sans fil doit contester l'accès au canal et empêcher toute sorte de collision sur ce canal.

Dans les réseaux sans fil, il existe des restrictions supplémentaires par rapport aux réseaux filaires. La présence de la mobilité, le problème des noeuds cachés et celui des noeuds exposés compliquent la situation.

1.6.1 Problèmes d'accès au canal dans les réseaux ad hoc

- **L'émetteur ne peut entendre une collision qu'il a engendré :** un nœud mobile ne sait pas quand son émission entre en collision car il ne peut émettre et écouter en même temps. Pour détecter et minimiser les collisions, le protocole MAC doit disposer d'un mécanisme actif (l'écoute du médium pour détecter quand celui-ci est libre) ou passif (attente d'une réponse positive pendant un laps de temps : le *timeout*).

- **Le problème du « terminal caché » :** deux nœuds mobiles A et B, ne partageant pas la même portée radio, peuvent envoyer leurs émission simultanément vers un même récepteur C. Il existe alors la possibilité de collision sur ce dernier. En effet, sans accord préalable, A et B n'ont aucun moyen de prendre conscience de l'autre communication en cours. On dit alors que A est caché de B et que B est caché de A.

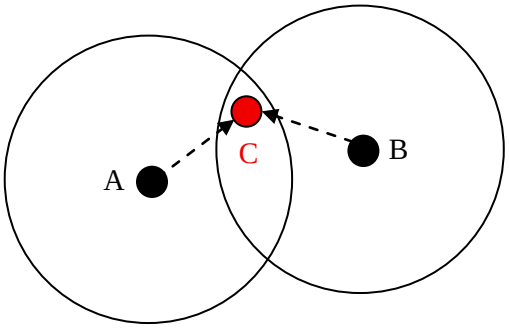
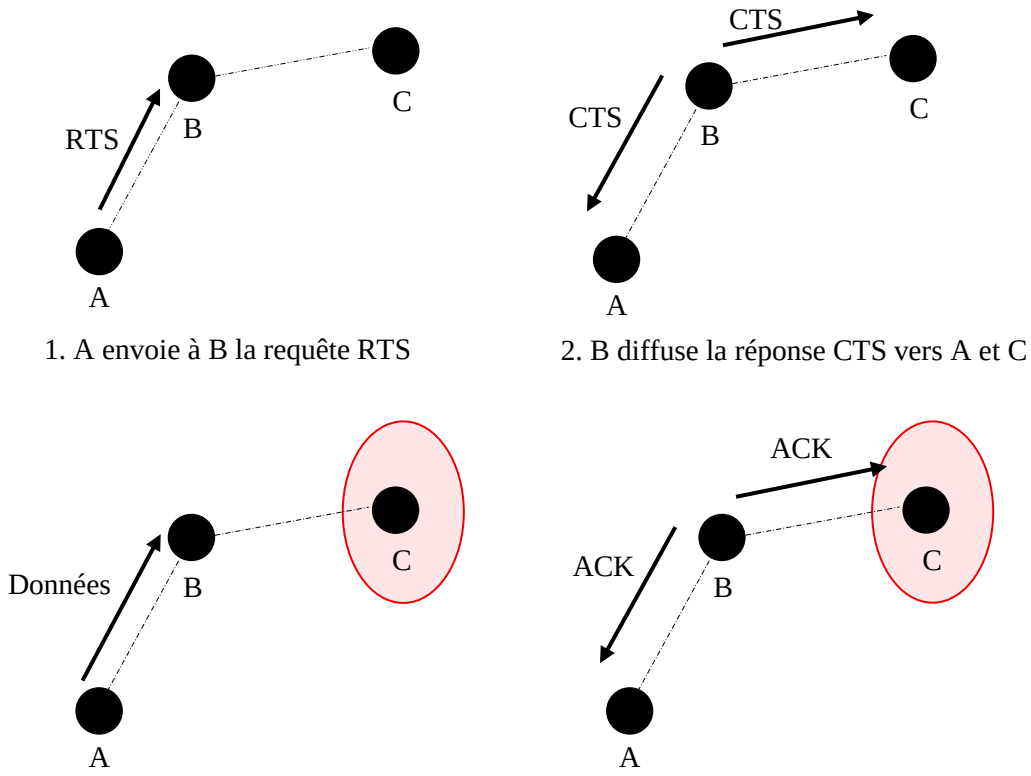


***FIG 1.4** – Le problème de terminal caché (collision au niveau du récepteur C)*

Pour remédier à ce problème, le protocole MAC peut disposer d'un mécanisme permettant d'informer les voisins du récepteur que le canal va être occupé. Une solution qui a été proposée consiste à utiliser des messages de contrôle RTS (pour Request To Send) et CTS (pour Clear To Send), il s'agit des protocoles basés-contention (asynchrones). Un nœud mobile désirant envoyer des données vers un autre nœud commence par diffuser le message RTS pour indiquer son intention d'émission. Le nœud récepteur peut accepter cette transmission en répondant par le message CTS. Ainsi, tous les voisins de l'émetteur et du récepteur vont être informés que le canal en question va être occupé, ce qui permettra d'éviter une collision sur ce canal. Certains protocoles utilisent déjà ce mécanisme de contention comme par exemple : le protocole ALOHA [10], CSMA et ses variantes [11], l'IEEE 802.11b [12]. La figure 1.5 illustre le concept de l'approche RTS-CTS.



1. A envoie à B la requête RTS

2. B diffuse la réponse CTS vers A et C

3. A envoie ses données vers B
C reste bloqué.

4. B affirme la réception des données par le message ACK

***FIG 1.5** – L'utilisation du mécanisme RTS/CTS pour résoudre le problème de terminal caché*

- **L'inconvénient de l'approche RTS-CTS :** la méthode RTS-CTS n'est pas une solution parfaite pour régler le problème des nœuds cachés. Il y a certains cas où cette solution ne peut pas empêcher les collisions. Comme c'est montré dans la figure 1.6, le nœud A désire transmettre ses données vers le nœud B, et le nœud D désire communiquer avec son voisin C. En appliquant le principe de la méthode RTS-CTS, deux collisions possibles peuvent avoir lieu si on suit le scénario suivant. Quand A commence l'envoi du message RTS vers B, ce dernier diffuse le message CTS pour dire que le canal est libre. Si on suppose que le noeud C reçoit ce dernier message simultanément avec la requête RTS envoyée par son voisin D, il est clair alors que le nœud C reçoit la première collision. Par conséquent, il ne peut pas déchiffré la bonne information, d'où il ne comprend pas la requête de D. Le nœud A par contre commence l'envoie de ses paquets de données vers B sans aucun problème. Comme D ne reçoit aucune réponse de la part de C, ce premier rediffuse sa requête une autre fois pour recevoir en fin la réponse CTS de la part de C. Ici, une

deuxième collision peut avoir lieu mais cette fois ci au niveau du nœud B qui peut recevoir les données de A et la réponse CTS de C en même temps.

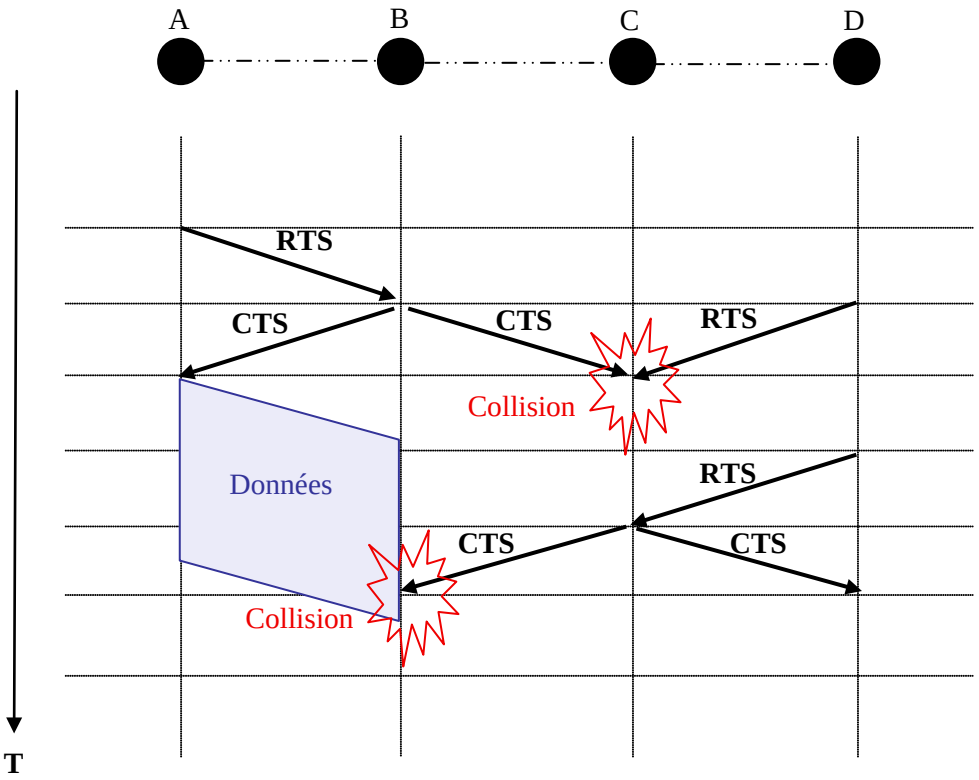


***FIG 1.6** – Inconvénient de la méthode CTS/RTS*

Un autre scénario peut monter la possibilité d'avoir une autre forme de collision. C'est lorsque plusieurs réponses CTS sont envoyées vers différents nœuds voisins en même temps (voir la figure 1.7).

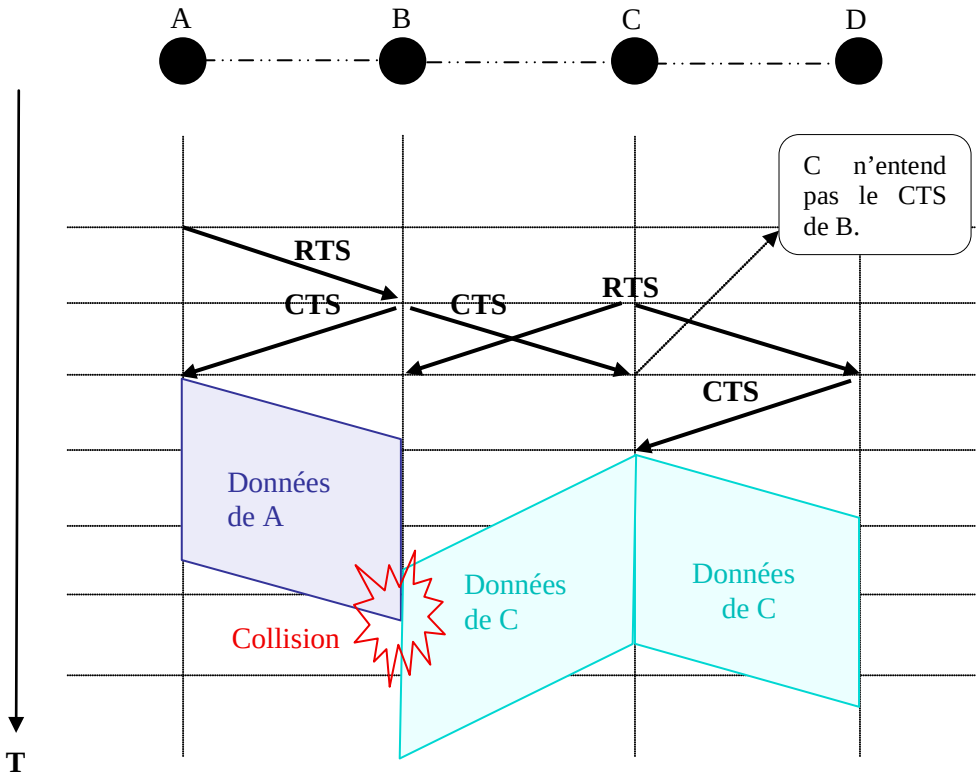


***FIG 1.7** – Autre Inconvénient de la méthode CTS/RTS*

- **Le problème des nœuds exposés :** cette difficulté est le pendant du problème du terminal caché. Lors d'une communication radio entre deux mobiles, leurs voisins respectifs ne peuvent émettre car ils risquent de perturber l'échange en cours. Beaucoup de mobiles se retrouvent alors immobilisés. La figure 1.8 représente ce problème : le message du nœud S vers le nœud D fait croire à A qu'il ne peut pas émettre, alors qu'une communication entre A et B ne générait pas l'autre. Il peut être intéressant de proposer un algorithme qui n'impose pas abusivement des restrictions à l'ensemble des voisins à deux sauts. Par exemple, une solution consiste à varier la portée d'émission des nœuds [1]. Une autre solution consiste à utiliser des antennes directionnelles [13].

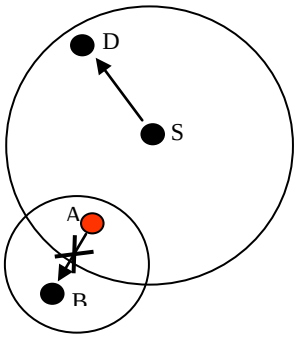


FIG 1.8 *– Exemple du problème de terminal exposé*

1.6.2 Quelques protocoles MAC proposés pour les réseaux ad hoc

Les protocoles MAC peuvent être classés en deux grandes catégories selon l'entité qui commence l'envoie de la requête en premier : l'émetteur ou le récepteur. Comme c'est montré dans la figure 1.9, le récepteur B commence par envoyer à l'émetteur A un message de signalisation RTR pour lui dire qu'il est prêt à recevoir les données de A. c'est une forme de vérification qui permet au récepteur de s'assurer que l'émetteur possède des paquets de données à transmettre. C'est une forme de communication passive car ce n'est pas l'émetteur qui initialise le contacte. En plus, il n'y a qu'un seul message de signalisation émis par le récepteur, en comparant ceci avec le mécanisme RTS-CTS qui possède deux messages de signalisation. Comme exemple de protocole MAC dont la communication est initiée par le récepteur, on peut citer le protocole MACA-BI [14], [15].

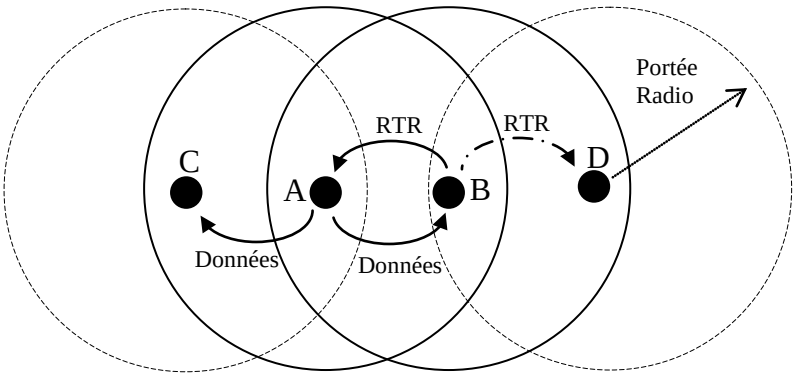


FIG 1.9 *– Principe du protocole MAC initié par le récepteur*

Contrairement à ce premier type de protocole MAC, les protocoles MAC initiés par l'émetteur se caractérisent par l'envoie d'un message de signalisation depuis l'émetteur pour informer le récepteur qu'il à l'intention d'envoyer ses données. Parmi les protocoles MAC initiés par l'émetteur on trouve : le protocole MACA (Multiple Access with Collision Avoidance) [16], MACAW (MACA with Acknowledgment) [17], MACA/PR [18] et FAMA (Floor Acquisition Multiple Access). La figure 1.10 montre un exemple de ce type de protocole. Le nœud A commence par l'envoie du premier message de signalisation RTS vers le nœud B (le récepteur) pour annoncer son désire d'émission. Le nœud B peut répondre par le deuxième message de signalisation CTS s'il est disposé de recevoir les données de A. Par conséquent, le nœud A commence l'envoie de ses paquets de données vers B.

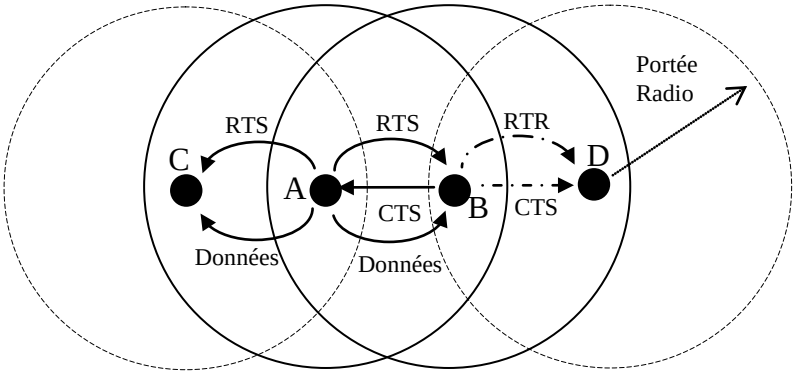


***FIG 1.10** – Principe du protocole MAC initié par l'émetteur*

Plusieurs protocoles MAC ont été proposés pour minimiser les problèmes de reprise sur collision, DBTMA (Dual Busy Tone Multiple Access) [19] [20] [21], [22] propose l'idée d'un système de signalisation n'appartenant pas au canal de transmission. C'est un protocole plus efficace que les modèles proposés en haut, mais il est nécessaire de disposer de plusieurs canaux séparés. Certains protocoles abordent le problème de l'économie d'énergie, comme une version hybride de MACAS, baptisé PAMA (Power-Aware Multi-Access Protocol) [23], [24]. Ce mécanisme utilise un serveur sur l'antenne pour détecter des communications radios dans le voisinage, et pour activer ensuite le module radio. Ainsi, le dispositif d'émission et de réception est utilisé seulement en cas de besoin; il se passe en position sommeil le reste du temps pour économiser l'énergie du mobile.

Un autre protocole MAC, associé à la mobilité du réseau employant des antennes omnidirectionnelles, est le protocole MARCH (Media Access with Reduced Handshake) [25]. Ce protocole vise à minimiser le taux des messages de contrôle dans le réseau.

1.7 Le routage dans les réseaux ad hoc

Le routage (Routing en anglais) est le mécanisme d'ouverture et d'entretient d'une communication entre deux nœuds. L'opération est alors supportée par la source, le destinataire et les relais (les nœuds intermédiaires) supportant l'échange. Dans un premier temps, la source doit trouver le chemin jusqu'au destinataire. Elle peut s'appuyer sur une connaissance préalable du chemin ou demander à d'autres entités un chemin partiel ou complet. Si la source utilise une information incomplète, une chaîne de relais peut se créer jusqu'à joindre le destinataire. Ce dernier s'appuie alors sur les informations reçues pour retrouver le chemin vers la source est ainsi construire le chemin. Dans le cas d'un réseau ad hoc, l'opération de routage se heurte à de nombreuses difficultés car la recherche des routes s'appuie sur des informations dynamiques.

Des mécanismes réguliers (périodiques) ou utilisés seulement lors de la recherche de routes (à la demande) doivent exister pour obtenir une route valable. En clair, les nœuds ne peuvent s'appuyer sur une information statique, et doivent obtenir lors de la recherche de route l'information la plus « fraîche » possible. Trouver un chemin n'est qu'une partie du problème, il faut pouvoir assurer la stabilité des communications car la mobilité des nœuds peut entraîner de nombreuses reconfigurations des chemins. Ainsi, durant la communication, l'ensemble des relais d'une communication va changer plus ou moins fréquemment.

De plus, par la nature même des réseaux ad hoc, les déconnexions peuvent être un événement normal (par oppositions à un événement anormale, dû à une erreur dans un réseau filaire). En effet, un nœud peut se retrouver sans possibilité de joindre le destinataire, simplement par le fait qu'il ne possède pas de voisins ou que le graphe du réseau joignable n'inclut pas le destinataire. Cette erreur oblige la source et/ou certains relais à temporiser des envoies et/ou à informer les applications de la source de cet évènement. Pour plus d'informations, le lecteur pourra se rapporter à un exposé des problèmes [26] et à différentes analyses des protocoles de routage existants [27], [28], [29].

1.7.1 Conception d'un protocole de routage

Afin d'assurer la connexion des réseaux ad hoc au sens classique du terme, toute conception de protocole de routage doit tenir compte des problèmes suivants :

- ***La charge du réseau*** *:* l'optimisation des ressources du réseau renferme deux autres sous problèmes, qui sont l'évitement des boucles et l'empêchement de concentration du trafic autour de certains nœuds ou liens.

- ***La fiabilité des communications* :** l'élimination d'un lien, pour cause de panne ou de mobilité devrait, idéalement, augmenter le moins possible les temps de latence.

- ***L'optimisation des routes* :** la stratégie de routage doit offrir des chemins optimaux et pouvoir prendre en compte différentes métriques de coûts (bande passante, nombre de liens, ressources du réseau,…). La stratégie de routage doit assurer aussi une maintenance efficace des routes avec le moindre coût possible.

- ***Le temps de latence* :** la qualité des temps de latence et des chemins doit augmenter dans le cas où la connectivité du réseau augmente.

1.7.2 Classification des protocoles de routage

Comme le montre la figure 1.11, ces protocoles de routage peuvent être classés en deux grandes familles : les protocoles proactifs (ou Table Driven Protocols) et les protocoles réactifs (Source Initiated ou On-Demand Driven Protocols). D'autres auteurs incluent une troisième famille qui s'appelle les protocoles hybrides. Cette nouvelle famille combine les techniques des deux premières familles.

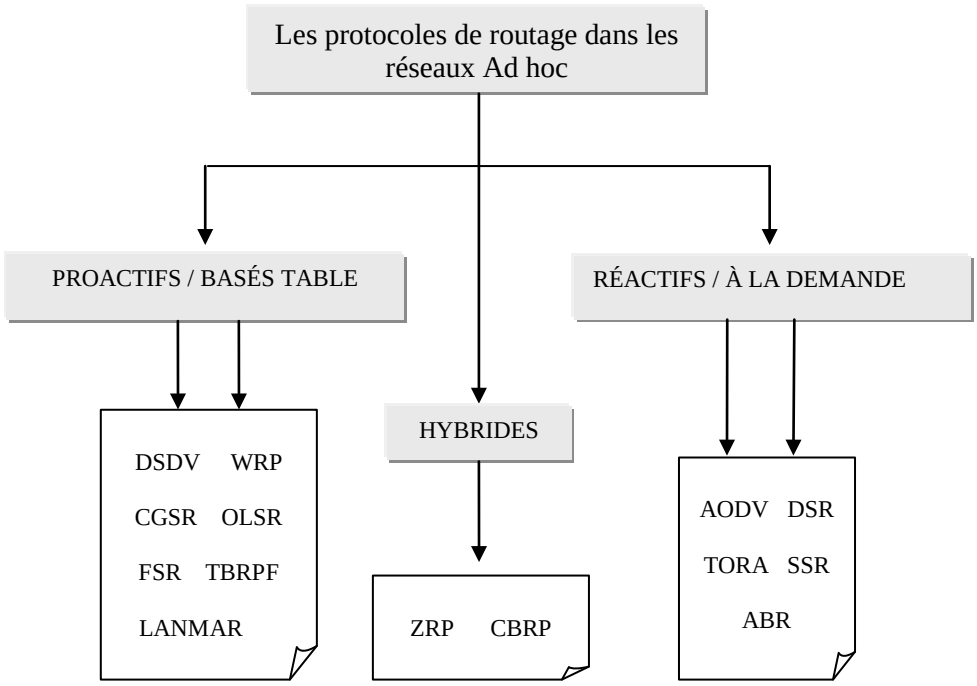


***FIG 1.11** – Classification des protocoles de routage des réseaux Ad hoc*

Avant de citer les différents protocoles de routage qui ont été proposés dans la littérature, il est nécessaire de présenter les différentes techniques utilisées dans les protocoles des trois classes déjà présentées.

- ***La technique à état de lien (Link Stat) :*** cette technique se base sur les informations rassemblées sur l'état des liens dans le réseau (c'est typiquement l'ensemble des liens vers les voisins d'un nœud). Ces informations sont disséminées dans le réseau périodiquement pour permettre aux nœuds de construire une carte complète du réseau (graphe).

- ***La technique à vecteur de distance (Distance Vector) :*** cette technique se base sur un échange entre voisins des informations de distance des destinations connues. Chaque nœud envoie à ses voisins la liste des destinations qui lui sont accessibles et le coût correspondant. Le nœud récepteur met à jour sa liste locale des destinations avec les coûts minimums.

- ***La technique d'apprentissage en arrière (Back Ward learning) :*** dans cette technique, la source commence par inonder le réseau tout entier par sa requête de recherche de route vers une destination donnée. Cette requête est relayée nœud par nœud, chacun de ces nœuds enregistre une trace de la requête avant de la rediffuser une nouvelle fois. Lorsque celle-ci atteints un nœud possédant une route vers la destination ou la destination elle-même, celui-ci envoie à la source un message de réponse sur la route inverse. Le message de réponse passe alors par la route inverse et activera la route en même temps.

- ***La technique de routage source (Source Routing) :*** dans cette technique, le chemin à parcourir par le paquet est inclus dans l'entête du paquet de données. Le routage s'effectue du nœud en nœud en consultant cet entête du paquet. Notons que chaque nœud qui reçoit le paquet supprime son adresse de l'entête avant de le retransmettre. Une fois le paquet arrive à la destination, celui-ci sera délivré à la couche réseau du dernier hôte.

1.7.2.1 Les protocoles de routage proactifs

Les protocoles de cette famille, appelée aussi Table driven Protocoles, tentent de maintenir et de mettre à jour l'information de routage entre les nœuds du réseau. Le principe est inspiré du protocole de routage de Bellman Ford (DBF) [30], utilisé entre autre dans les mises à jour des routeurs sur Internet. Chaque nœud du réseau maintient une ou plusieurs Tables contenant les informations de routage des autres nœuds du même réseau. Quand la topologie du réseau change, chaque nœud diffuse les modification de sa (ses) table(s) à une partie du réseau, afin de maintenir la fraîcheur du réseau. Il existe plusieurs algorithmes qui diffèrent par le nombre de tables nécessaires

que doit maintenir chaque nœud du réseau, et par la méthode de diffusion utilisée lors de changement du réseau. Dans ce qui suit, je vais donné quelques protocoles de cette première famille.

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

Le protocole DSDV [31], [32] (*Destination Sequenced Distance Vector*) de Perkins *et al.* conserve dans une table les destinations possibles dans le réseau, avec pour chaque entrée le voisin à joindre, le nombre de sauts et un numéro de séquence. Cette dernière information est un numéro choisi par la source, pour permettre aux autres nœuds de connaître l'information la plus récente. Pour diffuser les informations de sa table, le mobile émet à ses voisins deux types de paquets (dans le but de réduire la consommation du médium aérien) : « *full dump*» et « *incremental* ». Les paquets de type «*full dump*» sont utilisés pour une mise à jour complète dans laquelle le nœud diffuse la totalité de la table. Les paquets de type « *Incremental* » sont utilisés pour une mise à jour incrémentale dans laquelle le nœud diffuse une partie de la table, il s'agit des entrées qui ont subi des changements).

Le DSDV élimine les deux problèmes de boucle de routage «*Routing loop*» et celui de comptage à l'infinie «*Counting to infinity*». Cependant, une unité mobile doit attendre la réception d'une nouvelle mise à jour de la destination pour mettre à jour l'entrée associée à cette destination dans la table. Ce qui fait que le DSDV est lent.

- **Wireless Routing Protocol (WPR)**

Murthy *et al.* proposent WRP (*Wireless Routing Protocol*) [33], [34]. Le protocole maintient quatre tables : la table de distance (la distance entre le nœud et les différents correspondants), de prochain saut (le voisin à joindre pour contacter un mobile dans le réseau), de coût (la latence entre le nœud et les différents destinataires) et de retransmission de messages (avec les informations de mise à jour). Chaque mobile émet régulièrement un message indiquant les changements dans sa table de routage ; Il diffuse également des demandes de confirmation de présence à ses voisins pour les informer des changements topologiques tout en vérifiant la validité de son voisinage.

Les nœuds connaissent leurs voisins par la réception des ACKs et d'autres messages. Si un nœud n'envoie pas de messages, il doit envoyer périodiquement un message «Hello» pour assurer la connectivité. Sinon, l'absence du message provenant du nœud indique une défaillance de liens. Quand un nœud A reçoit un message «Hello» d'un nouveau voisin B, ce dernier est rajouté à la table de routage du nœud A, de plus, le nœud A envoie une copie de sa table de routage à B.

♦ **Cluster Switch gateway Routing (CSGR)**

Le protocole CSGR (*Cluster Switch Gateway Routing*) [35] proposé par Chiang *et al.* est une architecture reposant sur les groupes. Chaque groupe possède un chef qui se charge des communications à l'intérieur de sa « tribu. » La maintenance des informations de routage est située dans chaque chef (le chef est élu en utilisant un algorithme distribué à l'intérieure de chaque groupe. Le critère le plus important pour ce choix étant la stabilité). Ceux-ci conservent une table de tous les autres chefs accessibles dans le réseau, associés aux nœuds de liaison « Gateway » à utiliser pour les joindre (un nœud de liaison est un nœud qui est à la portée de deux ou plusieurs chefs à la fois).

Chaque nœud maintient une table des membres du groupe dans laquelle chaque nœud mobile du réseau se voit associé son chef. Ces tables sont diffusées périodiquement par chaque nœud en utilisant l'algorithme DSDV. Les nœuds mettent à jour leurs tables des membres de groupe à chaque réception d'une table similaire d'un voisin. En plus, chaque nœud doit aussi maintenir une table de routage qui est utilisée pour déterminer le prochain saut à effectuer pour atteindre la destination. de cette manière, les paquets sont relayés alternativement entre les chefs et les passerelles jusqu'à ce qu'ils atteignent leur destination. Certains nœuds du réseau tels que les chefs et les passerelles ont une charge de communication plus importante, une panne au niveau de ces nœuds affecte la fiabilité du réseau.

Cette approche est intéressante lors de la gestion de groupes de nœuds restant ensembles, car il y a peu de changement dans la topologie locale d'un groupe. Mais si de fréquents changements de groupes interviennent, alors la maintenance devient trop coûteuse.

♦ **Optimised Link State Routing Protocol (OLSR)**

OLSR [36], [37] (*Optimized Link State Routing Protocol*) est un protocole proactif. Comme son nom l'indique, OLSR est un protocole à état de lien optimisé. OLSR offre des routes optimales en terme de nombre de sauts dans le réseau. Dans un protocole à état de lien chaque nœud déclare ses liens directs avec ses voisins à tout le réseau. Dans le cas d'OLSR, les nœuds ne déclarent qu'une sous partie de leur voisinage. L'ensemble des voisins déclarés est choisi de façon à pouvoir atteindre tout le voisinage à deux sauts. Cet ensemble s'appelle l'ensemble des relais multipoint. Les relais multipoint sont utilisés dans le but de minimiser le trafic du à la diffusion des messages de contrôle dans le réseau. De plus, les routes sont construites à la base des relais multipoint.

Pour maintenir à jour toutes les informations nécessaires au choix des relais multipoint et le calcul de la table de routage, les nœuds OLSR ont besoin de s'échanger des informations périodiquement. Pour s'informer du proche voisinage, les nœuds OLSR envoient des messages de HELLOS contenant leur liste de voisins. Ces messages

permettent à chacun de choisir son ensemble des relais multipoint constitué d'un sous-ensemble de voisins. Le deuxième type de message primordial à OLSR s'appelle le message TC. Par ce message les sous-ensembles de voisinage sont déclarés périodiquement dans le réseau, en utilisant ces mêmes relais multipoint. Ces informations offrent une carte de réseau pour la construction de la table de routage.

La table de routage est construite au niveau de tous les noeuds et le routage des données s'effectue saut par saut sans l'intervention d'OLSR dont son rôle s'arrête à la mise à jour de la table de routage de la pile IP.

D'autres protocoles de routage proactifs existent, comme par exemple le protocole TBRPF (Topology Broadcast Based on Reverse-Path Forwarding) [38] dans lequel, les nœuds s'échangent périodiquement leurs voisinages respectifs, et chacun construit un arbre de topologie offrant des routes vers les autres nœuds avec le nombre de sauts minimum. Les arbres de topologie sont ensuite échangés à leur tour entre voisins. Ceci permet de propager les informations de topologie dans tout le réseau.

Il existe aussi un protocole proactif très connu, il s'agit du protocole FSR (Fisheye State Routing) ou œil de poisson [39], [40] qui part du principe qu'un changement de topologie lointain n'a pas une influence significative sur le calcul de la route localement. De ce fait, la fréquence de rafraîchissement se fait en fonction de la distance. Plus un nœud est loin, moins il reçoit fréquemment les mises à jour de topologie locale. Sur la base des informations de voisinage et de topologie, FSR construit sa table de routage. Au cours de routage, le chemin à parcourir se précise en se rapprochant de la destination.

On peut mentionner aussi le protocole Lanmar (Landmark Routing Protocol) [41], [42] et qui peut être qualifié de protocole proactif complexe. Il est différents des protocoles déjà présentés jusqu'à maintenant, car il voit le réseau comme une multitude de sous réseaux logiques hiérarchiques et se base sur le schéma d'adressage du réseau. Lanmar est conçu pour des réseaux très larges composés de plusieurs groupes distincts dont les membres se déplacent généralement ensemble tout en collaborant et communiquant avec le reste du réseau. Il ouvre une nouvelle classe de protocoles, celles de protocoles hiérarchiques pour les très grands réseaux.

Un des avantages des protocoles proactifs, sous réserve d'une mise à jour correcte de l'ensemble des tables, est le fait de trouver rapidement le destinataire sans devoir au préalable effectuer une recherche dans le réseau complet (l'information pour router les messages est immédiatement disponible). De plus, les pertes de chemin sont relativement peu fréquentes, car les changements de topologie sont diffusés automatiquement. Enfin, chaque nœud possède des informations sur l'ensemble du réseau, ce qui peut se révéler pratique pour la couche application. L'inconvénient de ce type de protocole est le coût de maintien des informations de topologie et de routage même lorsqu'il n'y en pas besoin et en absence de trafic de données. Le coût de

maintien des informations de routage génère une consommation continuelle de bande passante.

1.7.2.2 Les protocoles de routage réactifs

À l'opposé de l'approche proactive, l'approche réactive (ou *source-initiated on-demand*) découvre le chemin quand nécessaire. Lorsqu'un nœud cherche à joindre un correspondant dans le réseau, il utilise un protocole de diffusion. La méthode classique de recherche de route consiste à inonder le réseau avec une requête. Cette inondation perturbe tout le réseau puisque tous les nœuds doivent répéter la requête. Une fois atteint, le correspondant peut répondre par un message point-à-point qui informe la source de sa présence et du chemin à suivre pour le joindre. Cette opération, représentée par la figure 1.12, peut être renouvelée à un niveau local pour la maintenance des routes.

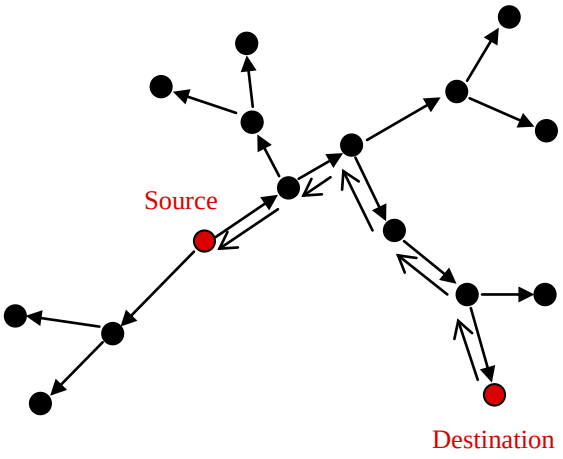


***FIG 1.12** – Principe de recherche de routes par la diffusion*

- **Ad Hoc On-Demand Distance Vector (AODV)**

Le protocole AODV (*Ad hoc On-demand Distance Vector*) [43], [44], de Perking et Royer, appartient à la famille des protocoles réactifs. AODV se base sur le principe des protocoles de routage à vecteur de distance. Une première propriété d'AODV est sa façon d'éviter le problème des boucles et le comptage à l'infinie des protocoles à vecteur de distance classiques en utilisant des numéros de séquence dans ses messages de contrôle. AODV effectue les recherches de routes sur demande et ne sauvegarde que les routes utilisées dans sa table de routage. C'est à dire lorsqu'arrivent des paquets IP

destinés à un hôte pour lequel il n'existe pas de route maintenue. Le routage des paquets de données se base sur cette table et se fait saut par saut.

Pour trouver un chemin, AODV diffuse un message de recherche de route à travers le réseau. Ce message est relayé nœud par nœud. Lorsque ce message atteints un nœud possédant une route vers la destination, ou la destination elle-même, celui-ci envoie à la source un message de réponse par la route inverse. Cette dernière peut alors commencer à envoyer ses données. Par ailleurs, à la réception du type de message de recherche de route, le nœud enregistre une trace de cette requête avant de la diffuser à son tour. Le message de réponse passera par la route inverse et activera la route en même temps jusqu'à la source.

Afin de limiter le coût dans le réseau, AODV propose d'étendre la recherche progressivement. Initialement, la requête est diffusée à un nombre de sauts limité. Si la source ne reçoit aucune réponse après un délai d'attente déterminé, elle transmet un autre message de recherche en augmentant le nombre de sauts maximum. En cas de non réponse, cette procédure est répétée un nombre maximum de fois avant de déclarer que cette destination est injoignable.

A la rupture d'un lien d'une route active, AODV tente de réparer la connectivité localement en diffusant une requête de recherche de route dans le voisinage. Si cette tentative échoue, alors, il envoie un message d'erreur vers la source et la route est détruite dans les tables de routage des nœuds intermédiaires. Ensuite, une nouvelle recherche de route est lancée par la source.

- **Dynamic Source Routing (DSR)**

DSR (Dynamic Source Routing) [45], [46] est protocole réactif proposé par Johnson et Maltz. Sa particularité par rapport aux autres protocoles réactifs MANET est qu'il se base sur le « Source Routing ». C'est-à-dire que le chemin à parcourir par le paquet est inclus dans l'entête du paquet de données. Le routage s'effectue de nœud en nœud en consultant cette entête du paquet. Ce protocole ne génère pas de messages périodiques, et ne réagit que lorsque les changements concernent une communication en cours entre deux nœuds. Tout comme AODV, DSR cherche des routes vers les destinations requises seulement à la demande. Pour ce faire, il diffuse une requête dans le réseau et attend les réponses. A l'obtention des chemins de parcours pour la destination recherchée, il insère la liste des nœuds à traverser dans l'entête de tous les paquets de données à envoyer. Ce mécanisme évite entre autre les boucles en traversant les nœuds. Il permet aussi à ces derniers de récupérer des chemins vers d'autres destinations dans le réseau ; ces chemins sont gardés dans leurs caches respectifs pour une utilisation éventuelle ultérieure.

Plus en détail, DSR est composé de deux mécanismes ; le premier utilisé pour chercher les routes à la demande, et le second s'occupe de maintenance de route des communications en cours.

- Le mécanisme de recherche de route est utilisé lorsqu'un nœud S voulant envoyer des données à un nœud destination D ne trouve pas de route dans son cache. Dans ce cas, S inonde le réseau par une requête de recherche de route. Cette requête est relayée saut par saut par tous les nœuds en rajoutant à chaque fois dans le message l'identifiant du nœud courant. Cette inondation continue jusqu'à atteindre la destination ou bien un nœud possédant une route pour la destination. dans ce cas, la liste des nœuds est inversée et un message de réponse est envoyé à S en unicast. Remarquons qu'en première étape DSR se contente d'envoyer la requête de recherche de chemin à ses voisins, et s'il ne reçoit pas de réponse, alors il la diffuse dans tout le réseau. Cette procédure est répétée un nombre de fois maximum avant de déclarer que la destination recherchée est injoignable.

- Le mécanisme de maintenance de route permet de signaler à S que la topologie vient de changer et que la route utilisée pour atteindre D n'est plus valable. Plus clairement, à la suite d'une rupture de lien entre deux noeuds intermédiaires reliant S à D, un message d'erreur est envoyé à S qui va essayer d'utiliser une route alternative de son cache, sinon rechercher une nouvelle route.

♦ **Temporally-Ordered Routing Algorithm routing protocol (TORA)**

TORA [47] (*Temporally-Ordered Routing Algorithm routing protocol*), est un protocole qui s'adapte à la mobilité des environnements mobiles à grande dynamicité. Le principe de TORA consiste à stocker 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 seront perdus (*rompus*). La principale caractéristique de TORA, est que les messages de contrôle sont limités à un ensemble réduite 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 (*réactifs*).

L'optimisation des routes (i.e. l'utilisation des meilleurs chemins) à 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*), qui est complètement différente des deux classes « *Link State* » et « *Distance Vector* ». TORA est basé sur le principe des algorithmes qui essaient de maintenir la propriété appelée "*Orientation Destination*" des graphes acycliques orientés (*où DAG : Directed Acyclic Graph*).Un graphe acyclique orienté est *orienté destination* s'il y a toujours un chemin possible vers une destination spécifiée. Le graphe devient *non orienté destination*, si un lien (ou *plus*) devient défaillant. Dans ce cas, les algorithmes utilisent 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 une métrique qui s'appelle « Hauteur » du nœud. Chaque nœud possède une hauteur qui l'échange avec l'ensemble de ses voisins directs. Cette métrique est utilisée dans l'orientation des liens du réseau. Un lien est toujours orienté du nœud qui a la plus grande hauteur, vers le nœud qui la plus petite hauteur (voir la figure 1.13). Les concepts de *la hauteur* 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. Le protocole TORA peut être divisé en quatre fonctions de base : création de routes, maintenance de routes, élimination de et optimisation de routes.

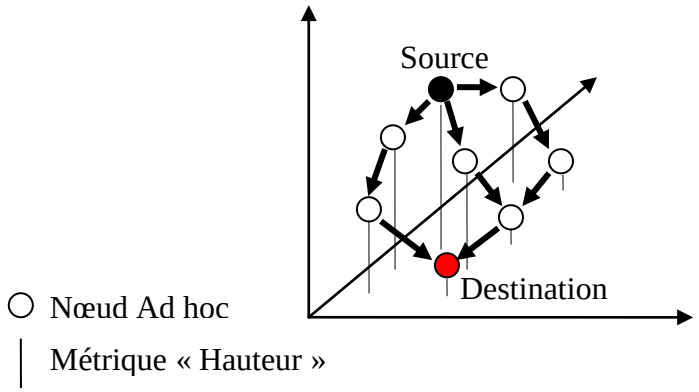


***FIG 1.13** – Création des routes dans le protocole unicast TORA*

Le processus de création (*ou de découverte*) de routes, pour une destination donnée, crée un DAG orienté vers cette destination. L'algorithme commence à l'état où la hauteur (le *paramètre d'ordre de propagation*) de la destination est initialisée à zéro, et la hauteur du reste des nœuds est indéfinie (*i.e. égale à NULL*).

Le nœud source diffuse un paquet QRY (*query*) spécifiant l'identificateur de la destination, *ID-destination*, qui identifie le nœud pour lequel l'algorithme est exécuté. Un nœud qui a une hauteur indéfinie et qui reçoit le paquet QRY, rediffuse le paquet à ses voisins. Un nœud qui a une valeur de taille différente de NULL, répond par l'envoi d'un paquet UPD (*update*) qui contient sa propre hauteur. Lors de la réception du paquet UPD, le nœud récepteur affecte la valeur de la hauteur contenant dans le paquet reçu incrémentée par un (01), à sa propre hauteur, à condition que cette valeur soit la plus petite par rapport à celles des autres voisins. De cette façon, un DAG est crée du nœud source vers le nœud destination (voir la figure 1.14).

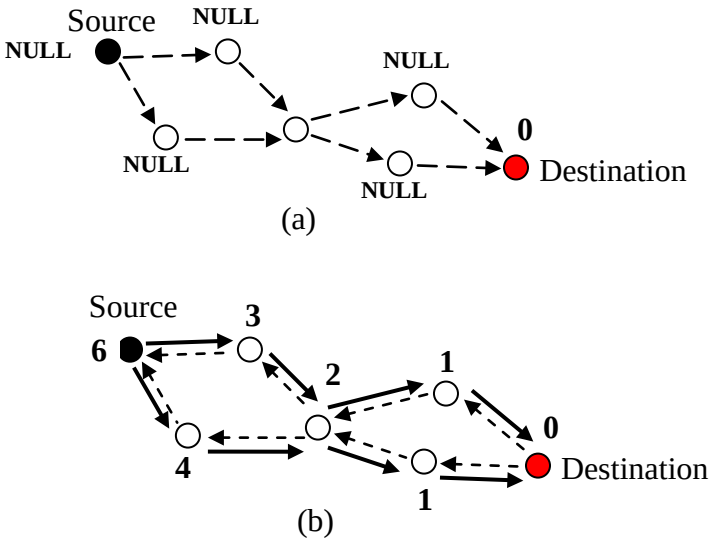


***FIG 1.14** –Diffusion du paquet QRY (a), Réception du paquet UPD (b) et création de DAG orienté destination.*

A cause de la mobilité des nœuds dans les réseaux ad hoc, des routes du DAG peuvent être rompues, dans ce cas une maintenance de routes doit être effectuée afin de rétablir un DAG pour la même destination. Quand un nœud *i* détecte une défaillance (*sachant qu'il ne possède pas de suivants valides vers la destination*), il lance un *nouveau niveau de référence*, cela est effectué comme suit : le nœud *i* ajuste sa hauteur pour qu'elle représente le maximum des hauteurs des nœuds voisins. Le nœud *i*, transmet par la suite un paquet UPD contenant la nouvelle hauteur. Par conséquent tous les liens vont être orientés du nœud *i* vers ses voisins, car la hauteur de *i* est devenue maximale. La diffusion du paquet UPD *inverse* le sens de tous les liens qui participent dans les chemins, où une défaillance est détectée; ce qui indique à la source l'invalidité des chemins rompus.

La fonction de suppression du TORA est effectuée en diffusant un paquet CLR (*clear*) dans le réseau, et cela afin de supprimer les routes invalides qui sont sauvegardées localement par les nœuds du réseau. Cela est fait, par exemple, dans le cas de détection de partitions.

D'autres protocoles existent, comme par exemple SSR (*Signal Stability based Routing*) [48] ou ABR (*Associativity-Based long-lived Routing*) [49]. Très proches d'AODV, ils tiennent compte de la fiabilité des liens, soit en étudiant la puissance du signal en réception (dans le cas de SSR), soit en écoutant les changements topologiques dans le voisinage (dans le cas d'ABR). À partir de ces informations, ils peuvent associer une mesure qualitative à chaque connexion. Ainsi, lors de la construction de la route, un indice de fiabilité est calculé pour chaque route créée, avec une préférence pour les liens ayant un coefficient de stabilité élevé.

Les algorithmes réactifs se révèlent plus adaptés dans le cas de réseaux ad hoc avec une taille importante et/ou une forte mobilité. Le fait qu'ils découvrent la route quand le besoin se fait ressentir permet d'éviter de réserver constamment une partie des communications pour indiquer les changements de topologie. Mais le problème qui se pose lors de changement de topologie est que ce type de protocole est obligé de réagir rapidement pour trouver une autre alternative à la route endommagée. En conséquent, ces mécanismes de maintenance sont lourds à gérer et créent des sursauts de trafic de contrôle à chaque tentative de recherche ou de réparation de route. Ces mécanismes entraînent un délai d'attente et une bufferisation des paquets de données dus au défaut de routes.

1.7.2.3 Les protocoles de routage hybrides

Les protocoles hybrides combinent les deux idées des protocoles proactifs et réactifs. Ils utilisent un protocole proactif, pour apprendre le proche voisinage (par exemple voisinage à deux ou trois sauts) ; ainsi ils disposent des routes immédiatement dans le voisinage. Au-delà de cette zone prédéfinie, le protocole hybride fait appel aux techniques des protocoles réactifs pour chercher des routes. Avec ce découpage, le réseau est partagé en plusieurs zones, et la recherche de route en mode réactif peut être améliorée. A la réception d'une requête de recherche réactive, un nœud peut indiquer immédiatement si la destination est dans le voisinage ou non, et par conséquent savoir s'il faut aiguiller ladite requête vers les autres zones sans déranger le reste de sa zone. Ce type de protocole s'adapte bien aux grands réseaux, cependant, il cumule aussi les inconvénients des protocoles réactifs et proactifs : messages de contrôle périodiques, plus, le coût d'ouverture d'une nouvelle route.

♦ **Zone Routing Protocol (ZRP)**

Le ZRP (*Zone Routing Protocol*) de Haas [50] est un protocole hybride qui utilise un protocole réactif au niveau local (*i.e.* avec des voisins situés à une distance inférieure à k sauts) et un protocole proactif pour le routage entre groupes (appelés *routing zone*). Pour l'opération de recherche de route, chaque nœud connaît le moyen de joindre les autres dans son groupe. Si le mobile a besoin de contacter un nœud à l'extérieur de cette zone, il utilise le protocole IERP (*Interzone Routing Protocol*) qui envoie une demande de route aux voisins situés à la périphérie du groupe. Ces derniers vérifient alors si le destinataire est dans leur groupe et lui transmettent le message si celui-ci est présent. Dans le cas contraire, l'algorithme est réitéré : chacun de ces nœuds procède à une demande de route vers les nœuds en périphérie de son groupe. Une fois le chemin trouvé, le nœud destinataire renvoie un message point-à-point vers la source. Pour éviter un coût important des méthodes de diffusion entre IERP, certaines optimisations locales à chaque groupe sont proposées dans [50].

L'opération de diffusion est donc une brique majeure de nombreux protocoles de routage. En effet, la diffusion existe naturellement dans les protocoles réactifs, mais aussi dans certains protocoles proactifs (par exemple OLSR) et dans les protocoles hybrides. La diffusion caractérise donc en grande partie les performances du protocole. Dans le cas d'AODV par exemple, il n'y a aucune optimisation de la diffusion, les performances sont donc médiocres et limitées, ce qui réserve ce protocole pour des réseaux de petite taille. Plus généralement, l'utilisation naïve du mécanisme de diffusion se solde par de nombreuses complications, notamment avec les nombreux accès simultanés au médium aérien. Dans le chapitre 3, je présente une analyse du problème de la diffusion, avec une présentation des protocoles de diffusion déjà existants et certaines solution permettant d'éviter l'inondation et la sous utilisation du réseau ad hoc quand il s'agit d'une communication multicast que je vais présenté en détails dans le chapitre 2.

Un autre protocole hybride existe, il s'agit du protocole CBRP (Cluster Based Routing Protocole) [51]. Contrairement au protocole précédent, le CBRP divise le réseau en plusieurs zones différentes (appelées clusters) qui se chevauchent et peuvent s'entrecouper. La formation des clusters est très simple. En effet, c'est le nœud avec le plus petit identifiant qui est élu comme chef de cluster. Néanmoins pour éviter des changements fréquents dans les structures des clusters, un nœud membre (non chef) n'est pas autorisé à défier le chef même si son identifiant est plus petit. Deux chefs de clusters ne doivent pas avoir de liens directs entre eux. Si, par la suite d'un changement de topologie ceci se produit, alors, il faut reconstruire les clusters et celui possédant l'identifiant le plus faible est élu comme chef. Les nœuds se trouvant à l'extrémité des clusters sont appelés nœuds passerelles (Gateways) qui permettent de relayer les informations entre les clusters.

Le tableau 1.1 suivant résume les principales caractéristiques des protocoles présentés. Tous les protocoles essaient d'éviter les boucles soit en utilisant des numéros de séquences, soit comme DSR en utilisant le routage par la source. AODV et DSR ont beaucoup de ressemblance à part l'utilisation de la notion de routage par la source DSR. En revanche, DSR peut offrir plusieurs routes pour destination données.

	Sans boucle	Plusieurs routes possibles	Distribué	Type	Sécurité	Messages de contrôle périodiques
ABR	non	non	oui	réactif	non	oui
AODV	oui	non	oui	réactif	non	non
CBRP	oui	oui	oui	hybride	non	oui
CSGR	oui	non	oui	proactif	non	oui
DSDV	oui	non	Oui	proactif	non	oui
DSR	oui	oui	oui	réactif	non	non
FSR	oui	non	oui	proactif	non	oui
LANMAR	oui	non	oui	proactif	non	oui
OLSR	oui	non	oui	proactif	non	oui
SSR	non	non	oui	réactif	non	oui
TBRPF	oui	non	oui	proactif	non	oui
TORA	non	oui	non	réactif	non	non
ZRP	oui	oui	oui	hybride	non	oui
WRP	non	non	oui	proactif	non	oui

***TAB 1.1** – Caractéristiques des protocoles de routage unicast dans les réseaux Ad hoc*

J'ai présenté quelques protocoles de routage proactif qui se basent essentiellement sur les échanges d'information périodiques, des protocoles réactifs qui se basent sur la recherche de routes à la demande, et des protocoles hybrides qui combinent ces deux premiers types de protocoles. Cette présentation n'est pas exhaustive et il existe une multitude de protocoles qui proposent des solutions de routage plus ou moins différentes tels que STAR [52], Mobilemesh [53] (et ses différents composants [54] et [55]).

Il existe une autre classe de protocoles qui utilisent les informations géographiques et la position des nœuds pour trouver les routes. Parmi ces protocoles on peut citer LAR [56]. Cette approche pose des problèmes lorsque l'environnement ne permet pas d'avoir des informations GPS exactes et le fonctionnement du réseau se trouve pénalisé surtout en « indoor ». par exemple, des nœuds peuvent être très proches l'un de l'autre, mais, des obstacles peuvent faire que les ondes ne passent jamais (forêt, grands immeubles, …). Ou encore, les nœuds peuvent se trouver loin l'un de l'autre, mais la

réflexion des ondes radios et les échos rendent la communication possible entre les nœuds. La possibilité de communiquer n'est donc pas liée de façon déterministe à la distance physique.

On trouve aussi une autre famille de protocoles qui partent du principe qu'il faut économiser l'énergie totale du réseau et tentent d'acheminer les données via des routes consommant le moins possible les batteries des nœuds intermédiaires. Ce choix peut engendrer des routes plus longues que les routes optimales réduisant ainsi le débit total de bout en bout. Dans cette catégorie on peut citer les propositions PARO [57] et PAMAS [24].

1.8 Autres défis des réseaux ad hoc

Après avoir présenté les problèmes inhérents au routage et à l'accès à la couche MAC, cette section propose un rapide survol des autres défis, et les travaux actuels qui en découlent, autour des réseaux ad hoc.

- **Changement d'échelle ou Scalabilité:** l'augmentation du nombre de nœuds ou de la taille du réseau peut entraîner des diminutions de performances au niveau des protocoles. Il peut être intéressant de voir le comportement des protocoles dans de telles conditions

- **Problème de l'énergie :** chaque entité du réseau consomme de l'énergie, surtout l'émission et la réception des communications. De nombreux travaux s'intéressent à la réduction de la consommation. Une des idées est d'offrir une politique d'alternance des communications entre les nœuds. Une autre idée est de réduire la portée des nœuds, tout en maintenant une connexité complète du réseau.

- **Sécurité :** au vu du modèle totalement décentralisé des réseaux ad hoc, la sécurité de cette architecture devient un point très important. Elle peut se situer au niveau des données, où il est nécessaire d'établir un système de chiffrage et/ou d'authentification entre la source et le destinataire. Cette mesure sert alors à éviter que tous les mobiles dans le réseau (qu'ils soient relais du message ou simple mobile entendant les communications dans leur voisinage) puissent espionner ou même modifier les échanges entre ces deux correspondants. De même, le problème d'authentification, bien connu dans le réseau Internet, nécessite la mise en place de serveurs de clés et/ou de certificats de façon totalement décentralisée. Pour plus d'information, on pourra se référer à [59].

- **Intergiciels :** une fois mise en place les protocoles de bas niveau (routage, découverte), il existe aussi l'installation de protocoles au niveau applicatif, dans

le but d'exploiter les capacités du réseau. Le but est de pouvoir offrir certaines possibilités comme la qualité de service, l'utilisation d'applications distribuées et la distribution de services à l'intérieur du réseau [60], [61].

- **Connexion à Internet :** concernant la connexion à Internet, un mobile peut éventuellement être relié par plusieurs réseaux ad hoc ou par réseau filaire (lorsque l'utilisateur rentre chez lui et connecte son ordinateur à son modem). La nécessité d'un routage à plus grand grain, c'est-à-dire d'une possibilité de gérer la mobilité sur Internet quel que soit son point d'accès, doit exister. Une solution élégante comme Mobile IP existe, et permet d'attribuer à un mobile une adresse IP permanente quelle que soit sa position dans le réseau des réseaux. Mais son utilisation, de manière transparente dans les réseaux ad hoc, reste néanmoins à étudier.

1.9 Autres notions de routage dans les réseaux ad hoc

- **Multihoping :** dans les réseaux ad hoc, la communication entre deux nœuds quelconque du réseau peut passer par un certain nombre de nœuds intermédiaires, ainsi la communication engendre plusieurs sauts pour arriver à la bonne destination. C'est pour cette raison qu'on l'appelle communication Multi-saut ou Multihop en anglais. Contrairement aux réseaux ad hoc, la communication dans les réseaux cellulaires n'engendre qu'un seul saut entre la source et la destination, d'où sort le terme « single hop ».

- **Inondation :** l'inondation ou la diffusion totale consiste à faire propager le paquet de données ou de signalisation dans le réseau entier. Un nœud qui initie l'inondation envoie le paquet à tous ses proches voisins. Ce comportement se répète avec ces derniers jusqu'à ce que le paquet en question atteint tus les nœuds du réseau. Notons quand même que durant l'inondation, les nœuds du réseaux peuvent appliquer certaines opérations ou traitement afin d'empêcher certaines complications tel que : le bouclage et la duplication des messages.

- **Concept de groupe :** dans la communication en groupe, les messages sont transmis à des entités abstraites ou groupes , les émetteurs n'ont pas besoins de connaître les membres du groupe destinataire. La gestion des membres d'un groupe permet à un élément de joindre à groupe, de se déplacer ailleurs, puis d'y revenir une autre fois. C'est en ce sens que la communication en groupe assure une indépendance de la localisation, ce qui la rend parfaitement adaptée à des topologies de réseaux configurables.

- **Temps de latence :** c'est le temps que requière un paquet pour traverser le réseau d'un nœud d'entrée vers un nœud de sortie. Ce délai dépend de type de média de transmission (temps de propagation), du nombre d'équipement réseau traversé (temps de traitement) et enfin de la taille des paquets (temps d'émission).

- **Gigue :** c'est la valeur de variation maximale d'acheminement sur le réseaux. Ce paramètre est très important pour les applications multimédia et temps réel qui requièrent des inter-paquets relativement stables

- **Taux de perte :** c'est le rapport du nombre d'octets perdus sur le nombre total d'octets émis.

- **Bande passante :** c'est la capacité de transmission d'une liaison du réseau mesurée en bit par seconde. La bande passante disponible sur un lien dépend de ses caractéristiques techniques et du nombre de flots de données.

- **Disponibilité :** c'est le rapport de temps de bon fonctionnement du service sur le temps total d'ouverture de service.

- **Arbre de routage :** c'est une structure de routage regroupant un certains nombre de nœuds du réseau liés entre eux de telle manière à ce que chaque nœud de l'arbre possède au plus un seul père. Un arbre de routage peut être partagé entre les nœuds (même priorité) ou spécifié à une source ou à un sous groupe de nœuds (centralisé).

1.10 Conclusion

Dans ce chapitre j'ai présenté les réseaux ad hoc, leurs caractéristiques et le problème de routage au sein de ce type de réseaux. J'ai présenté aussi quelques défis et certaines notions de base que l'on aura besoin dans les chapitres suivants. Ces derniers vont être consacré spécialement au routage multicast dans les environnements mobiles ad hoc. Certains protocoles de routage multicast vont être présenté avec plus de détail dans le chapitre qui va suivre, pour en tirer enfin un protocole de routage multicast, et qui représentera par la suite l'objet d'une amélioration par l'intégration de certaines techniques de routages.

Chapitre 2

Le Routage Multicast dans les Réseaux mobiles ad hoc

2.1 Introduction

La plupart des applications Internet utilisent des communications point à point. Mais le multicast est en train d'évoluer régulièrement et son utilité ne cesse de ce justifier jour après jour. Des applications telles que la téléconférence, ou les jeux en réseau seront d'une utilisation quotidienne encore plus importante qu'aujourd'hui et à l'échelle mondiale. L'IETF s'est penché sur la gestion du multicast depuis plusieurs années et a déjà réservé une plage d'adresse spécifique à ce genre de communications. Une adresse de groupe IP/multicast utilisée est une adresse IP de classe D comprise entre 224.0.0.0 à 239.255.255.255. Le but de la communication multicast est d'améliorer l'utilisation de la bande passante en évitant d'établir inutilement des communications point-à-point entre tous les intervenants. Parmi les protocoles multicast les plus connus définis par l'IETF, on peut citer DVMRP [62], PIM [63], CBT [64], et MOSPF [65].

Concevoir un protocole de routage multicast reste une tâche difficile. La difficulté réside dans le choix de la structure à mettre en place. Et différentes options se présentent : former un arbre centralisé sur un noyau partagé, maintenir un arbre par source, utiliser une grille reliant tous les membres d'un groupe, ou encore opter tout simplement pour l'inondation en essayant de l'optimiser si possible.

Quelle structure s'adapte le mieux aux réseaux sans fil, quels sont les avantages et les inconvénients de telle structure ?

Ce qui est possible dans les réseaux filaires peut s'avérer fastidieux et coûteux si ce n'est pas impossible dans le monde sans fil, et inversement. Ce chapitre présente le routage multicast dans les réseaux filaires et les réseaux sans fil tout en soulignant la difficulté de la mise au point de ces derniers. Différentes structures et technologies concernant le routage multicast sans fil seront présentées, ainsi que des protocoles visant à fournir la fiabilité, la scalabilité et la robustesse en même temps.

2.2 Routage multicast dans les réseaux filaires

Le multicast désigne la possibilité qu'un seul paquet IP peut atteindre différents destinataires référencés par l'adresse du groupe destinataire de ce paquet. Pour cela, le paquet IP en question ne fait pas référence précisément à une adresse destination de classe A, B ou C, mais à une adresse de groupe, dite de classe D. Les destinataires sont eux référencés par un protocole comme IGMP comme étant "abonnés" au groupe désigné par l'adresse de classe D, utilisée dans le champ adresse destination du paquet IP. La classe D va de 224.0.0.0 à 239.255.255.255. Acheminer de façon optimale un paquet IP à tous ses destinataires est loin d'être une chose simple. La tâche de routage multicast est accomplie par les routeurs mettant en œuvre un protocole de routage multicast d'une part, et gérer d'autre part les abonnements des clients qui eux ne jouent pas le rôle des routeurs et se trouvent généralement dans des réseaux locaux ; ce qui est assuré par un protocole de gestion des groupes Internet, déjà mentionné, IGMP (Internet Group Management Protocol) [66].

Plusieurs techniques et protocoles de routage multicast filaire ont été proposés et utilisés avec succès. Dans les paragraphes qui suivent, je présenterai ces techniques et ces protocoles tout en soulignant leurs inconvénients lorsqu'ils sont appliqués dans les réseaux sans fil.

2.2.1 L'architecture multicast IP

En 1989, Deering [67] a proposé l'architecture multicast IP pour permettre des communications point à multipoint dans les réseaux filaires (TCP/IP) et l'Internet. Dans cette architecture, un groupe multicast est définit par une adresse IP unique de la classe D représentant son adresse IP multicast. Afin de recevoir des paquets multicast, un récepteur doit joindre un groupe multicast particulier. L'émetteur multicast, par contre, peut transmettre des paquets multicast vers n'importe quel groupe multicast sans devoir être membre d'un groupe multicast particulier, et sans chercher à connaître le récepteur.

La distribution des paquets multicast est manipulée par les routeurs du réseau via l'aide des protocoles de routage multicast. Pour cette raison, tous les routeurs du réseau doivent supporter le routage multicast. En générale, c'est une architecture simple et flexible, malgré le fait que les émetteurs multicast n'ont aucun contrôle à effectuer à travers le processus de distribution multicast qui peut rendre la politique administrative et de sécurité encombrante. En dépit de ces inconvénients, cette architecture est largement acceptée et représente une base pour la conception de plusieurs protocoles de routage multicast.

2.2.2 Les tunnels multicast et le MBone

Le fait que tous les routeurs du réseau doivent supporter la fonction multicast, le déploiement multicast à grande échelle dans l'Internet s'avère impossible. Afin de régler ce problème, un paquet multicast est encapsulé dans un paquet unicast et est transmis d'un réseau multicast vers un autre réseau multicast, où il sera décapsulé et traité en tant qu'un paquet multicast. La connexion ou la liaison sur laquelle le paquet encapsulé passe s'appelle « *tunnel multicast* » ou « *mtunnel* », comme le montre la figure 2.1. Dans cette figure, le paquet multicast est envoyé depuis l'émetteur du réseau multicast #1 vers un autre membre du réseau multicast #2. Le routeur multicast RM1 encapsule le paquet multicast pour générer un nouveau paquet unicast puis, il l'envoie sur le tunnel multicast qui a été établi entre RM1 et RM2. Le paquet est décapsulé par RM2, et le paquet multicast décapsulé est envoyé vers le membre multicast du réseau multicast #2. De ce fait, les routeurs d'un réseau ne seront pas tous obligés de supporter le principe de multicast.

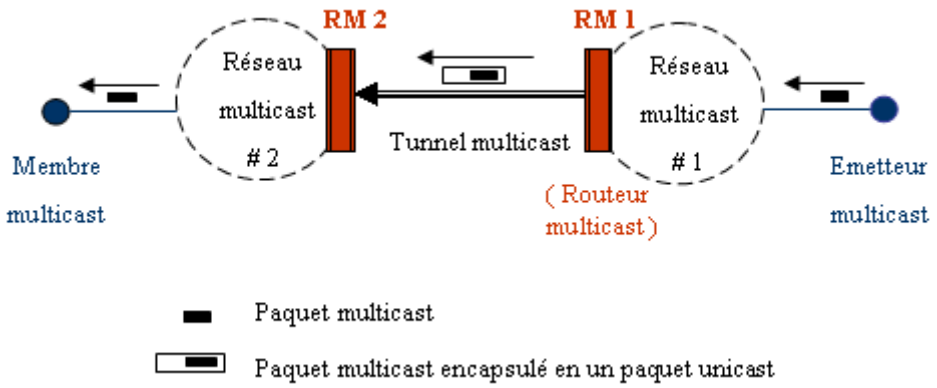


FIG 2.1 – *Concept des tunnels multicast dans le MBone*

En 1992, le MBone (Semi Permanent Multicast Backbone) [68] a été établi sur l'Internet en utilisant les tunnels multicast. Le protocole multicast DVMRP (Distance Vector Multicast Routing Protocol) [69] a été utilisé pour assurer le routage multicast dans le MBone. Cependant, n'importe quel protocole de routage multicast peut être utilisé au niveau du réseau local sauf qu'il faut fournir les mécanismes nécessaires assurant son interopérabilité avec le DVMRP.

2.2.3 Les algorithmes de routage multicast filaires

Actuellement, plusieurs protocoles de routage multicast filaires sont disponibles. Tous ces protocoles essaient de fournir des services multicast efficace par la création et la maintenance de certaines structures de routage qui regroupent tous les membres du groupe multicast en question. Les algorithmes de routage multicast filaire peuvent être classés en quatre catégories :

- ***Les algorithmes utilisant la technique de l'inondation (Flooding)*** qui se caractérise par la simplicité et l'efficacité. C'est la technique qui offre plus de redondance dans les chemins, par conséquent, les algorithmes associés sont les plus robustes et les plus tolérants aux ruptures des liens. Cependant, elle présente l'énorme inconvénient de consommation de bande passante.

- ***Les algorithmes mettant en œuvre le principe du chemin inverse RPF (Reverse Path Forwarding)***. Le principe de fonctionnement de ces algorithmes [70], [71], consiste à réémettre un paquet en avant s'il est reçu depuis un nœud se trouvant sur le meilleur chemin inverse menant à la source. Cet algorithme permet à la diffusion de suivre les chemins optimaux jusqu'à atteindre tous les nœuds du réseau.
  Il existe une deuxième version de cette catégorie qui s'adapte mieux aux diffusions multicast vers un groupe de noeuds au lieu de la totalité des nœuds. Cette deuxième version consiste à propager le premier paquet dans tout le réseau comme décrit précédemment, les chemins traversés dessinent ainsi un arbre centré sur la source émettrice, ensuite les branches inutiles seront supprimées. C'est-à-dire les chemins dont les feuilles n'appartiennent pas au groupe multicast. Ceci est effectué récursivement en remontant les branches à partir des feuilles. Tout nœud ne possédant pas de branches contenant des abonnés demande à son parent (via le nœud duquel il a reçu le paquet multicast diffusé la première fois) d'être supprimé. Cette deuxième technique se base alors sur le mécanisme de « Diffusion et Purgation »[72] et le principe de RPF en même temps. Les protocoles de routage multicast filaires DVMRP (Distance Vector Multicast Routing protocol) et PIM-DM (Protocol Independent Multicast Dense Mode) [73] implémentent à leur manière le principe de RPF.

- ***Les algorithmes se basant sur les informations topologiques***. Dans le protocole MOSPF (Multicast Open Shortest Path First) [65], les arbres de diffusion sont calculés au fur et à mesure depuis la diffusion du premier paquet en se basant sur les informations fournies par OSPF et les abonnées. Cependant, ces mécanismes demandent un taux de contrôle très élevé et ne peuvent pas s'adapter facilement avec le comportement mobile des nœuds dans les réseaux mobiles ad hoc.

- ***D'autres algorithmes sont conçus autours d'arbres centrés sur un noyau*** [64], [74]. Le principe de cette famille d'algorithmes est de construire un arbre unique par groupe multicast géré par son noyau, et auquel se rattachent les sources et les récepteurs. Les paquets multicast sont envoyés en premier lieu vers le nœud central, où ils seront acheminés par la suite sur les branches de l'arbre vers tous les membres multicast. Comme exemple, on peut citer les protocoles CBT [75], et PIM-SM [76], [77] qui utilisent la notion d'arbre partagé avec un noyau appelé « *point de rendez-vous* ». Le choix du noyau, son emplacement, et sa migration éventuelle jouent un rôle important sur l'efficacité de ces algorithmes. Cette approche possède un taux de contrôle bas mais le chemin suivi n'est pas nécessairement court. En plus, dans les réseaux dynamiques, la capacité peut diminuer dynamiquement à moins que le nœud central et l'arbre partagé puissent s'adapter assez rapidement à la mobilité [13].

2.2.4 IGMP (Internet Group Management Protocol)

IGMP (Internet Group Management Protocol), défini dans [66] et [78], permet aux machines de déclarer leur abonnement à un ou plusieurs groupes auprès du (des) routeur(s) multipoint sur le réseau local dont elles dépendent, soit spontanément soit après interrogation du (des) routeur(s). Les routeurs d'un même réseau local élisent l'un d'entre eux pour gérer les abonnements vers des groupes multicast. Celui-ci diffusera alors les paquets de données multicast destinés à ce ou ces groupes.
Ce protocole existe pour le moment en trois versions distinctes, chacune des deux premières versions fait l'objet d'une RFC, alors que la troisième est encore à l'état de draft à l'IETF.

2.3 Routage multicast dans les réseaux ad hoc

Le routage multicast représente l'un des principaux sujets de la nouvelle technologie sans fil par laquelle on tente d'assurer le bon acheminement des paquets de données depuis un ou plusieurs émetteurs vers plusieurs récepteurs.

D'après ce qu'on a déjà vue dans les paragraphes précédents, les protocoles de routage multicast classiques existants ne peuvent pas être appliqués directement sur les réseaux mobiles ad hoc. Actuellement, plusieurs protocoles de routage unicast existent pour l'environnement mobile ad hoc. Le but principal de ces algorithmes est d'assurer la communication unicast (point-à-point) dans un environnement très dynamique et avec un taux minimal de contrôle. Ces mêmes critères sont demandés dans les protocoles de routage multicast. En plus, ils doivent prendre en considération la mobilité du réseau, qui peut être due à la mobilité des sources, des destinations, des nœuds intermédiaires, ou de la mobilité de groupe multicast entier.

2.3.1 Les problèmes de routage multicast dans les réseaux ad hoc

Vu les limitations des réseaux ad hoc, la construction des routes doit être faite avec un minimum de contrôle et de consommation de bande passante. L'étude et la mise en œuvre des algorithmes de routage multicast sont des problèmes complexes. Les groupes sont dynamiques et évoluent au cours du temps, la topologie du réseau peut alors changer fréquemment, ce qui nous mène à étudier les problèmes suivants avant la conception de tout nouveau protocole multicast :

- **L'efficacité** : le protocole doit minimiser la charge du réseau (les messages de contrôle périodiques et la circulation des paquets de données redondants). Cependant, l'optimisation des ressources du réseau renferme deux autres sous problèmes qui sont : les boucles de routage et la concentration du trafic au niveau de certains liens et nouds.

- **La robustesse** : le protocole doit trouver des chemins alternatifs aux chemins détruits par cause de panne ou de mobilité des nœuds. La réparation de ces pannes doit augmenter le moindre possible les temps de latence.

- **L'adaptabilité** : la topologie d'un réseau ad hoc peut changer à travers le temps, c'est-à-dire qu'un réseau ad hoc très mobile peut avoir une certaine stabilité dans une partie du réseau ou dans sa totalité, d'une manière similaire, un réseau ad hoc statique peut subitement devenir très mobile. Différents mécanismes de routage multicast appropriés aux réseaux de grande ou de faible mobilité doivent être développés (De même si le réseau est dispersé ou dense). L'idéale est que les nœuds du réseau ad hoc soient capables de s'adapter aux différents changements que peut subir le réseau en se déviant dynamiquement entre les différents mécanismes multicast. Ceci doit se faire avec un minimum d'efforts et d'inconvénients (c'est-à-dire la perte des paquets).

- **La mobilité illimitée** : Quelques solutions multicast s'adaptent avec la mobilité discrète par laquelle les périodes de mouvement sont interrompues par les périodes de repos. D'autres solutions proposent des limites sur la direction, la vitesse et le nombre de nœuds qui bougent simultanément. Par contre, il faut imposer le cas d'un réseau avec une mobilité extrême et continue de tous les composants.

- **Le multicast intégré** : les solutions multicast pour les réseaux mobiles ad hoc différents beaucoup de celles proposées pour les réseau filaires (une raison principale est le taux de transmission). Dans le but d'offrir un service multicast intégré, de nouveaux mécanismes doivent être développés pour l'interopérabilité des solutions multicast filaires et sans fil.

2.3.2 Les différentes structures de communication multicast

Le support de distribution multicast définit la construction des chemins sur lesquels les paquets de données peuvent transiter pour arriver en fin aux membres de groupe multicast considéré. La conception d'un protocole de routage est une tâche difficile. La difficulté réside dans le choix de la structure de routage à mettre en place. Et différentes options se présentent : former un arbre centralisé sur un noyau partagé, maintenir un arbre par source, utiliser une grille reliant tous les membres d'un groupe,ou encore opter tout simplement par l'inondation en essayant de l'optimiser si possible.

2.3.2.1 La diffusion (*Flooding/Broadcasting*)

La diffusion (*broadcasting*) est l'opération de transmettre un message à l'ensemble du réseau[*]. Les problèmes de cette opération (*broadcast*) dans un réseau sans fil ont fait l'objet de nombreux travaux avec, comme but premier, la réduction de nombre de réémissions lors de la diffusion d'un message à l'intégrité du réseau. Dans cette méthode, aucune structure ou architecture n'est imposée. Et afin d'éviter le problème d'inondation (*Broadcast Storm*) [79], qui peut être causé par les boucles de diffusion et les collisions excessives, certains mécanismes doivent être présents.

L'avantage de cette méthode est que chaque membre du groupe multicast reçoit sûrement le paquet diffusé par la source, on parle alors de la robustesse. En plus, cette technique s'adapte bien avec les réseaux à grande mobilité. Cependant, la bande passante est sévèrement consommée à cause des messages redondants.

2.3.2.2 Les arbres multicast (*Multicast Trees*)

L'origine de cette structure revient à celle des protocoles multicast filaires. Les protocoles multicast filaires, dont la robustesse ne représente pas la condition nécessaire mais plutôt l'efficacité qui représente leur point critique, se basent essentiellement sur des structures en arbre. On trouve ici deux types d'arbre : « Arbre basé source » (*Source-Based Tree = SBT*) et « Arbre basé noyau » (*Core-Based Tree = CBT*). Ces deux structures vont être présentées avec plus de détails dans ce qui suit.

[*] En français, le terme diffusion spécifie deux types d'opérations : la transmission d'un message à ses voisins et la transmission d'un message à l'ensemble du réseau. En anglais, on utilise dans le premier cas le mot narrowcast et dans le second cas le mot broadcast.

- **Arbre multicast basé source (*SBT: Source-Based Multicast Tree*)**

Dans ce système, un Arbre multicast est établi et maintenu par chaque source dans chaque groupe multicast. Ainsi, dans un environnement avec G groupes multicast dont chacun possède S sources, on aura à établir et à maintenir ($G * S$) arbres multicast. Dans cette méthode, chaque paquet multicast est acheminé de la source vers tous les membres du groupe multicast sur la branche la plus efficace.
La construction de l'arbre multicast est basée sur certains critères et selon les différents objectifs des applications :

- ***L'arbre de « Steiner »*** (*ST* : *Steiner Tree*) par exemple, est conçu pour minimiser le coût total dans le réseau (le coût entre la source et tous les membres du groupe multicast). Ceci est souhaitable dans beaucoup de situations, mais l'algorithme exact n'est pas faisable (*NP-Complet*).

- ***L'arbre de plus court chemin*** (*SPT* : *Shortest Path Tree*) minimise le coût entre la source et chaque membre du groupe multicast individuellement. Cette méthode est simple et plus utilisée.

D'une manière générale, le système *SBT* souffre principalement de deux problèmes. Le premier problème est la scalabilité: lors de l'ajout de nouveaux nœuds dans le réseau, et plus particulièrement les nœuds sources, la charge d'établissement et de maintenance de chaque arbre dans le réseau augmente aussi. Le deuxième problème est que dans la plus part des cas, il est nécessaire de connaître à priori l'information topologique du réseau. Dans les réseaux ad hoc, les changements topologiques fréquents, dues à la mobilité des nœuds, deviennent un facteur signifiant dans l'augmentation de la charge de contrôle du réseau, surtout lorsque le nombre d'arbres affectés par ces changements est important. Et par conséquent, il sera nécessaire de les réparer.

- **Arbre Multicast Basé Noyau (*CBT: Core-Based Multicast Tree*)**

L'approche *CBT* est plus scalable par rapport à l'approche *SBT*. Ici, au lieu de construire un arbre par source dans tous les groupes multicast, un unique arbre partagé est construit pour regrouper tous les membres du groupe multicast considéré. Les paquets multicast sont distribués le long des branches de cet arbre pour arriver enfin à tous les membres du groupe multicast (les destinations). Pour établir un tel arbre, on commence tout d'abord par la désignation d'un nœud spécifique appelé « *Noyau* » ou « *Nœud Cœur* » (*Core Node en anglais*), qui prendra la responsabilité de création et de maintenance de l'arbre partagé (c'est à dire de toutes les opérations d'ajout, d'élimination et/ou de déplacement des nouds dans le groupe). D'où, la nécessité d'avoir *un « Algorithme de sélection du noyau »* (*A Core Selection Algorithm en englais*) à partir de l'ensemble des nœuds du groupe multicast.

L'arbre partagé *CBT* peut être soit un arbre unidirectionnel ou bidirectionnel. Dans un arbre *CBT* unidirectionnel, les paquets de données multicast doivent être envoyés obligatoirement en unicast vers le noyau qui représente la racine de l'arbre partagé. A partir de ce noyau, les paquets de données multicast seront distribués sur les différentes branches de l'arbre jusqu'à ce qu'ils arrivent à tous les membres du groupe multicast en question. Cependant, dans un arbre bidirectionnel, les paquets multicast envoyés par n'importe quelle source vont être distribués directement sur les différentes branches de l'arbre sans devoir passer par le noyau. Il est clair donc que le système bidirectionnel est plus efficace en terme de la performance de communication et de coût total d'acheminement[*].

Généralement, l'arbre *CBT* est de type *MST (Minimum Spanning Tree)*, c'est à dire arbre à écartement minimal, car le chemin entre l'émetteur et le récepteur n'est pas nécessairement le plus court chemin (*SPT*). L'arbre partagé *CBT* est moins efficace que l'arbre *SBT* en terme de routage multicast, mais l'arbre partagé réduit sévèrement la charge et le coût de routage car ici le protocole n'as pas besoin de connaître l'information topologique du réseau.

L'un des inconvénients de l'approche *CBT* est que le trafic reste concentrer sur les liens partagés, ce qui résulte en une grande tendance de congestion sur ces liens. En plus, les paquets multicast ont une tendance d'être acheminés sur les chemins les moins optimaux car ils sont obligés de transiter sur les différentes branches de l'arbre partagé. Aussi, le noyau, qui est le composant le plus critique dans ce système, devient ici le seul point de défaillance. Par conséquent, la conception d'un protocole de routage multicast CBT robuste et efficace nécessite la fourniture de certains mécanismes supplémentaires permettant l'adaptation dynamique de l'arbre partagé dans une configuration plus efficace, et de se remettre du problème du noyau. Ces mécanismes supplémentaires engendrent bien sûr des coûts additionnels. Les protocoles CBT, PIM-SM, AMRoute [80], AMRIS [81], et AODV [82], utilisent l'approche CBT avec différents mécanismes d'efficacité et de robustesse.

2.3.2.3 La maille multicast (*Multicast Mesh*)

Les deux systèmes précédents utilisent l'arbre multicast car celui-ci a fait preuve de sa grande efficacité en terme de coût d'acheminement. L'arbre de distribution multicast connecte uniquement les nœuds nécessaires pour l'acheminement des paquets de données dans un graphe acyclique. Cependant, dans les réseaux mobiles ad hoc, le taux de changement des liens peut être assez important que les reconfigurations fréquentes de l'arbre deviennent indésirables.

[*] Le coût d'acheminement est définit par le nombre de transitions nécessaires pour acheminer le paquet multicast vers la destination.

La maille multicast offre une autre alternative en établissant une grille dans chaque groupe multicast. Cette grille regroupe tous les nœuds qui se trouvent dans le groupe multicast, ainsi, on aura plusieurs chemins qui mènent vers la même destination. L'utilisation d'une seule structure réunissant tous les membres d'un groupe multicast peut offrir, et à n'importe quel moment, des chemins alternatifs aux chemins endommagés en cas de panne de nœuds/liens. Ceci permet alors d'éviter les fréquentes reconfigurations de la structure d'acheminement et minimise en plus le taux des ruptures imprévues des sessions en cours. Tout cela réduit la charge du protocole de routage appliqué en terme d'information topologique.

Cependant, l'utilisation de la maille génère l'acheminement inutile des paquets multicast sur tous les chemins redondants de la maille, causant ainsi un coût d'acheminement supplémentaire (consommation inutile de la bande passante). En plus, et contrairement à l'arbre multicast, les nœuds de la maille peuvent avoir plusieurs parents en même temps. Ce qui peut causer le problème des boucles d'acheminement, dont tout protocole de routage multicast essaie de l'empêcher par l'ajout de certaines procédures ou mécanismes d'évitement de boucles. Le protocole CAMP (Core assisted Mesh Protocol) [88] est un exemple de protocole multicast basé maille.

Une nouvelle structure est générée à partir de ces deux dernières structures, il s'agit de la structure hybride générée par la combinaison d'un arbre et d'une maille. Cette nouvelle combinaison essaie d'offrir l'efficacité de l'arbre et la robustesse de la maille en même temps. Un exemple à cette nouvelle classe est le protocole *AMRoute* (Ad hoc Multicast Routing Protocol) [77].

2.3.2.4 Le groupe d'acheminement (*Forwarding Group*)

Dans ce système, on associe à chaque groupe multicast un sous-ensemble de nœuds appelé « Groupe d'acheminement ». L'établissement et la maintenance de ce groupe d'acheminement nécessitent la fourniture de certaines procédures permettant de joindre n'importe quel membre de groupe multicast à partir de groupe de nœuds d'acheminement correspondant. Dans cette méthode, il s'agit de maintenir un groupe de nœuds au lieu de maintenir des liens qui constituent par exemple un arbre ou une maille. Ce qui simplifie beaucoup la tâche aux nœuds, car ici et contrairement aux structures d'arbres ou de mailles, les nœuds n'ont pas à maintenir « les informations d'état » pour transmettre les paquets de données à leurs voisins. Tout paquet de données multicast arrivant pour la première fois (qui n'est pas dupliqué) à un nœud du groupe d'acheminement est rediffusé à ses voisins. D'où peu de nœuds intermédiaires sont obligés de sauvegarder les informations d'état. Des chemins alternatifs existent aussi dans ce système. Le protocole multicast ODMRP (On Demand Multicast Routing Protocol) [84] et le protocole LBM (Location Based Multicast) [85] utilisent ce système de groupe d'acheminement.

2.3.2.5 L'acheminement multicast basé localisation

Dans ce type d'acheminement, la distribution des paquets de données est basée sur ce qu'on appelle « Information de localisation » des nœuds intermédiaires et des nœuds récepteurs. Contrairement au modèle multicast IP, ici, le groupe de communication multicast est défini par une région géographique appelée « Région Multicast ». L'identificateur de cette région multicast est spécifié dans l'entête de chaque paquet multicast envoyé par la source. Si la source elle même n'existe pas dans la région multicast réceptrice, elle doit demander l'aide de certains nœuds externes constituant ce qu'on appelle « Région d'acheminement ». Tous les nœuds de cette région doivent acheminer les paquets de données qu'ils reçoivent à leurs bonnes destinations. Il est important d'assurer que la région d'acheminement couvre la région multicast dans le sens où on peut atteindre n'importe quel nœud de la région multicast depuis la région d'acheminement.

L'inconvénient de cette méthode est le taux de contrôle très élevé ainsi que la grande charge d'acheminement du moment où la source et les nœuds d'acheminement utilisent « la diffusion » lors d'envoie des paquets de données vers la région multicast. Le protocole LBM (Location-Based Multicast) [85] repose essentiellement sur cette méthode. Les améliorations de ce protocole visent à intégrer de nouveaux mécanismes permettant de minimiser la charge et le coût d'acheminement global.

2.3.2.6 L'arbre multicast basé stabilité (Stability-Based Tree)

Toutes les structures présentées précédemment n'ont pas pris en compte le taux de stabilité des routes et des liens des structures utilisées. Ici, on démontre comment « l'associativité » peut être appliquée dans le routage multicast au sein des réseaux ad hoc pour le contrôle de stabilité des liens.

Ce système construit un arbre multicast basé source en se reposant essentiellement sur « *la stabilité d'association ou liaisons* » entre chaque nœud et ses voisins. Ainsi, lorsque l'arbre est stable, on aura moins de reconfigurations à effectuer. Un très bonne exemple à ce système est le protocole ABAM (Associativity- Based Ad hoc Multicast) [13].

2.3.3 Les technologies de routage multicast

Les systèmes de routage multicast présentés précédemment sont de systèmes très classiques et traditionnels possédant bien sûr certains inconvénients comme par exemple la « *scalabilité* » signifiant la dégradation significative des performances du réseau quand la taille de celui ci change. Beaucoup d'efforts ont été consacrés ces dernières années afin de résoudre les anomalies des anciens protocoles. Ces efforts peuvent être classifiés dans les quatre catégories suivantes :

- **Overlay-based multicasting** ; cette méthode consiste à réserver les informations d'états du protocole au niveau des membres du groupe multicast. Ceci pour éviter l'explosion du réseau quand tous les nœuds participants dans le routage multicast manipulent ces informations.

- **Backbone-based multicasting** ; dans cette méthode, l'information d'état est sauvegardée uniquement dans un sous ensemble de nœuds représentant le noyau ou le cœur du réseau (Backbone).

- **Stateless multicasting ;** dans cette méthode, on n'a pas besoin de sauvegarder les informations d'état. Il s'agit de protocoles sans informations d'état.

- **D'autres protocoles multicast**; dans cette catégorie, on utilise des méthodes explicites permettant de réduire la charge du contrôle de certains protocoles existants. C'est une forme d'extension directe des protocoles traditionnels.

2.3.3.1 Overlay-based multicasting

La gestion des informations d'état dans les protocoles de routage multicast consiste à mettre à jour périodiquement les tables de routage dans le but de maintenir l'exactitude des structures de routage utilisées (maille ou arbre). Les protocoles traditionnels utilisent tous les nœuds membres ou non membres du groupe multicast pour sauvegarder et maintenir les informations d'état et de routage. Par conséquent, ils souffrent tous du problème de « scalabilité ». Une solution à ce problème consiste à minimiser l'ensemble de nœuds qui doivent sauvegarder ces informations d'état en affectant cette tâche uniquement aux noeuds membres du groupe.

Dans cette catégorie on construit une infrastructure virtuelle en dessus des liens physiques liant uniquement les membres du groupe multicast. Les liens du réseau virtuel sont en réalité des tunnels unicast dans le réseau physique. Le réseau virtuel s'occupe des fonctionnalités de routage multicast, alors que le réseau physique correspondant se charge de fournir un service data-gramme unicast de très bonne

qualité. Parmi les protocoles appartenant à cette catégorie on peut citer le protocole AMRoute [77] et le protocole PAST-DM [86].

2.3.3.2 Backbone-based multicasting

Dans cette catégorie les protocoles utilisent une autre manière pour sauvegarder les informations d'état. Ils choisissent quelques nœuds pour former un backbone virtuel, les informations d'état ne peuvent être sauvegardées et manipulées que par l'ensemble des nœuds de dernier.

Dans les réseaux ad hoc, il existe quelques protocoles multicast basés sur la méthode de backbone [87], [52] mais utilisant tous le principe d'ensemble dominant. Le principe d'ensemble dominant est comme suit, chaque nœud dans le réseau est considéré soit comme un voisin du 1er saut d'un nœud du backbone ou soit comme un nœud du backbone lui même. Le problème qui se pose ici est que le nombre de nœuds du backbone est pratiquement le même nombre de nœuds du réseau. Par conséquent, il n'y a pas une réduction effective de la charge du contrôle lorsque la distribution multicast est attribuée aux nœuds de ce backbone.

Un protocole de routage multicast *adaptatif* basé backbone est comme un protocole de routage ordinaire, sauf qu'il est basé sur une approche hiérarchique à deux niveaux. Le backbone qui est composé des nœuds cœurs responsables de l'acheminement des paquets de données; la maintenance et l'acheminement des informations d'états sont effectués uniquement à l'intérieure d'un autre *groupe local* enraciné à un nœud du backbone et formant un arbre d'acheminement. Cette méthode offre donc une bonne combinaison entre la qualité de système de diffusion à l'intérieure du backbone et l'efficacité de système d'arbre construit.

L'approche de routage multicast *adaptatif* basé backbone crée ce qu'on appelle l'« *ADB* » (Adaptive Dynamic Backbone), de telle sorte que les nœuds qui n'appartiennent pas au backbone soient très loin que possible des nœuds du backbone, dans le sens où la mobilité apparue dans la zone locale d'un nœud externe ne dépasse pas les nœuds du backbone. L'approche *ADB* crée une sorte de forêt avec plusieurs arbres de différentes hauteurs. Si la zone locale d'un arbre est relativement statique, la hauteur de cet arbre peut devenir importante, sinon, l'arbre doit être petit. D'où la nomination adaptatif, car selon l'environnement local du réseau courrant, les structures de routage changent d'état.

En premier lieu, chaque nœud du réseau se considère comme étant un nœud du backbone et envoie le message « Hello » à tous ses voisins. A la réception de ce message, chaque voisin calcule une certaine valeur appelée « Hauteur » représentant une sorte de métrique formé des trois champs suivants **:** « *la fréquence de panne de lien détecté* », **«** *la capacité ou le degré de connectivité restant* » et *« l'identificateur* ». On

peut représenter cette métrique par le triplet (*nlff*$^{-1}$, *degree*, *id*). Après avoir calculé la hauteur, le nœud voisin peut décider soit de rester un nœud du backbone ou de devenir un fils du backbone et ce selon la valeur de la hauteur calculée. Si le nœud devient un fils, il sauvegarde l'information de son père, et quand il diffuse son prochain message « Hello » vers ses voisins, il indique l'information du son père et remplace la valeur de son champ *nlff* par la nouvelle valeur *nlff* incrémenté par un. De cette manière, on peut construire le backbone, et les arbres enracinés aux nœuds de ce backbone ne vont pas être trop grandes à cause de la limitation de « la contrainte *nlff* incrèmentable » et « la contrainte limite de saut ». La hauteur des arbres dépend de la mobilité des nœuds dans cette zone [88].

Chaque nœud maintient aussi la table «*Core Forwarding Table* » dans laquelle il peut sauvegarder les informations de quelques nœuds du backbone qui ne sont pas ses parents via des messages « Hello » de ses voisins qui n'appartiennent pas au même nœud du backbone que lui. Il sauvegarde juste les informations de routage de ces nœuds. Dans une période « *core-forward-update* », chaque nœud fils envoie à son parent sa table « Core Forwarding Table ». Vers la fin, ces tables atteignent les nœuds du backbone ainsi, les nœuds du backbone peuvent connaître les autres nœuds du backbone. Quand un nœud désire joindre le groupe multicast, il envoie juste le message « Join-Request » à son parent. Si le parent n'est pas encore membre de l'arbre multicast, il continue l'envoie ascendant de ce message (vers ses parents) et se met comme étant un nœud d'acheminement, sinon, il ignore le message. De cette façon, l'arbre multicast d'acheminement est construit, cet arbre est enraciné depuis un nœud de backbone et franchissant tous les membres de ce groupe local. Lors d'envoie d'un paquet de données, un nœud membre du groupe va localement diffuser ce paquet de données, et les nœuds d'acheminement vont rediffuser le paquet. Quand le paquet atteint le nœud du backbone, ce dernier va le relayer dans un rang de backbone en l'encapsulant dans un message « *CORE-FORWARD* » et l'achemine vers les nœuds du backbone du 1er saut choisis à partir de sa table « *Core Forwarding Table* ». Quand les autres nœuds du backbone obtiennent le message encapsulé, ils enlèvent l'entête du message « *CORE-FORWARD* » et effectuent le routage multicast au niveau de groupe comme décrit précédemment.

La topologie de backbone est beaucoup plus simple et stable en la comparant avec la topologie de tout le réseau, mais les nœuds de ce backbone restent les points critiques du réseau. Ce qui impose des limites sur la scalabilité horizontale du réseau et sur le nombre de groupes locaux (les arbres), car tous les paquets de données vont passer à travers le même nombre de nœuds du backbone.

2.3.3.3 Stateless Multicasting

Dans tous les protocoles de routage présentés jusqu'à présent, les informations d'état sont maintenues par certains nœuds dans le réseau, et ce dans le but de préserver et mettre à jour les structures de routage utilisées. Actuellement, la désignation ou la conception des protocoles de routage multicast se réorientent vers les protocoles dits « *Stateless Multicasting* » ou protocoles multicast sans états. Dans cette catégorie aucune information d'état n'est sauvegardée dans aucun noeud du le réseau. Le but étant de minimiser les charges de communication dans tout le réseau.

Deux nouveaux protocoles utilisant ce principe ont été conçus ; le protocole DDM (Differentiel Destination Multicast) [86] et le protocole LGT (effective Location-Guided Tree construction algorithm) [87]. Ces protocoles vont être présentés avec détails dans le prochain chapitre.

2.3.3.4 D'autres protocoles multicast

Dans cette section il s'agit d'une classe qui concerne les extensions faites sur certains protocoles de routage dans les réseaux ad hoc. Ces extensions sont soit le passage d'un protocole de rouage unicast en un protocole de routage multicast, comme il est le cas pour le protocole unicast CEDAR [88] et son extension multicast MCEDAR [52]. Une autre forme d'extension concerne les protocoles multicast engendrant différents problèmes d'application, un bonne exemple est le protocole DCMP (Dynamic Core Based Multicast Routing Protocol) [89] qui représente l'extension du protocole ODMRP [81]. Le but du la nouvelle extension consiste à réduire la charge du contrôle en réduisant le nombre de nœuds d'acheminement de l'ancien protocole. Plus de détails sur cette extension vont être présentés dans le prochain chapitre.

2.3.4 Méthodes du développement d'un routage multicast fiable

Des efforts ont été consacrés ces dernières années pour le développement des protocoles de routage multicast fiables dans les réseaux ad hoc. Trois méthodes ont été mises pour fournir certains points ou degrés de fiabilité au niveau de la couche réseau. Ces trois méthodes sont : « *Nack-based method* », « *Flooding* », « *Gossip method* ».

Dans le routage multicast filaire, toutes ces trois méthodes ont été étudiées et exploitées avec un grand succès de fiabilité. Mais, peu de travaux ont été réalisés dans le monde sans fil. Dans cette section, on va introduire deux protocoles de routage multicast fiables, le premier est basé sur la méthode «*Nack-based method* », le deuxième est basé sur la méthode « *Gossip method* ».

2.3.4.1 Le protocole multicast *RALM*

La fiabilité est un aspect très important dans le routage multicast ad hoc. Cependant, peu de travaux ont été réalisés dans ce domaine. Le protocole *RALM* (Reliable Adaptive Lightweight Multicast Protocol) [90] est conçu pour étudier chacun des deux problèmes suivants : « la fiabilité » et « le contrôle de congestion » au sein des réseaux ad hoc. Ce protocole utilise le mode « Round-bobin » pour délivrer les données à un membre de groupe multicast par unité du temps. En cas de perte de données, le protocole peut garantir la fiabilité en utilisant l'approche « Send-and-Wait » pour contrôler la congestion.

Dans ce protocole, la source commence par l 'envoie multicast des données par des taux spécifiques (selon l'application). En cas de perte de données, cette première peut être détectée par l'envoie unicast en arrière d'un message « *Nack* » depuis certains nœuds membres du groupe multicast. La source initialise donc la phase de découverte en rajoutant les identificateurs de nœuds qui ont envoyé le message « *Nack* » à sa liste « Receive List ». La source choisit par la suite les nœuds de cette liste un par un pour effectuer l'opération de découverte jusqu 'à ce que le liste devienne vide. A chaque nœud choisi, la source retransmet en multicast les données perdues en mettant l'identificateur de ce nœud dans l'entête de chaque paquet retransmis. Uniquement le nœud membre du groupe possédant le même identificateur que celui de l'entête des paquets est permis de répondre en unicast par le message « *ACK* » et ce après avoir acquis tous les paquets de données perdus. Les autres nœuds membres qui ont perdu cette même donnée peuvent l'acquérir aussi du moment où la source envoie les paquets de données en multicast. Ces derniers nœuds ne sont permis de répondre par le message « *ACK* » que lorsqu'ils sont choisis par la source dans ses prochaines sélections de la liste RL. En cas où la source ne reçoit aucune réponse « *ACK* » depuis le membre concerné, elle retransmet la donnée un certain nombre de fois jusqu'à ce qu'une certaine période du temps prédéfini s'écoule. En fin, le nœud membre est supprimé de la liste « Receive List » de la source et ainsi de suite jusqu'à ce que la liste devienne vide. La source peut donc retourner à son application originale pour continuer l'envoie des paquets de données restants. On peut très bien remarquer que dans ce protocole, la source est une horloge de soi même car, elle assure un bon contrôle de congestion du moment où la continuité d'envoie des paquets de données est conditionnée par la réception du message « ACK ».

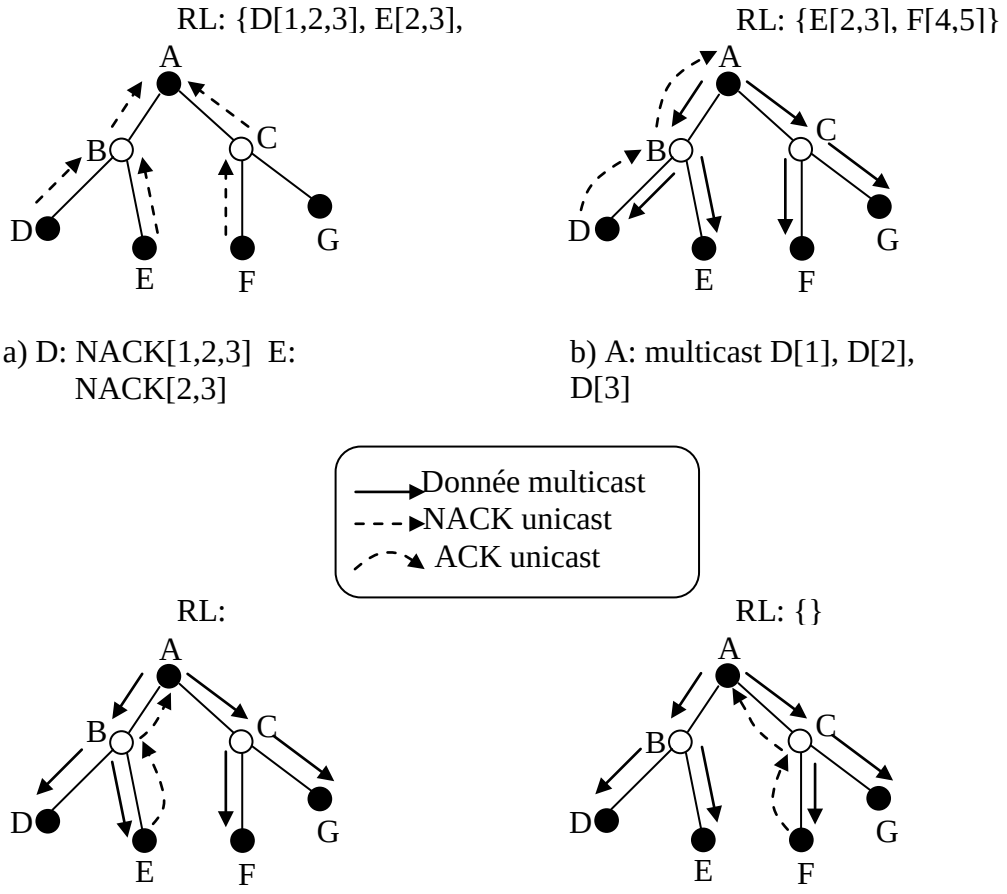


FIG 2.2 *– Fonctionnement du protocole RALM*

Comme le montre la figure 2.2, si la source A reçoit la négation « *NACK* », y compris les indices des paquets de données perdus, depuis les récepteurs D, E et F (D perd les paquets [1, 2, 3] ; E perd les paquets [2, 3] et F perd les paquets [4, 5]), alors la liste, initialement vide, RL (Receive List) du nœud A devient RL: {D [1,2,3], E [2,3], F [4,5]}. Par conséquent, le nœud A commence sa procédure de correction en choisissant en premier lieu le nœud D avec les indices des paquets perdus. A envoie en multicast les paquets {1, 2, 3} un par un. Après avoir reçu tous les paquets perdus, le nœud D répond par l'envoie unicast du message « *ACK* » au nœud A qui supprime par la suite l'identificateur de noeud D de sa liste RL. Les autres nœuds récepteurs qui ont perdus les mêmes données que A reçoivent les mêmes paquets mais ils ne peuvent pas répondre par le message « *ACK* » car ils ne possèdent pas le même identificateur que celui de l'entête de chacun des paquets retransmis par A. Une fois que A supprime D de

sa liste, il choisit le nœud E et lui envoie les paquets {2, 3}, cette fois ci, c'est le nœud E qui doit répondre à A par « *ACK* » car son identificateur correspond sûrement à l'identificateur mis dans l'entête des paquets {2, 3}. Et ainsi de suite jusqu'à ce que A traite tous les nœuds de sa liste LR.

Le protocole *RALM* est un protocole multicast fiable, basé taux (Rate-based) et contrôleur de congestion à la fois. Le but principal de *RALM* est de fournir la fiabilité du protocole multicast, mais pas de l'approche de routage elle-même comme le font les autres protocoles. *RALM* convient spécialement avec les groupes multicast de petite taille, car l'approche « Round-bobin » limite la scalabilité.

2.3.4.2 Le protocole multicast probabiliste *RDG*

Le protocole *RDG* (Route Driven Gossip) [91] est conçu dans le but de fournir un routage multicast fiable et probabiliste pour les réseaux ad hoc. Les protocoles déterministes conçus pour les réseaux ad hoc souffrent principalement de rapport existant entre la fiabilité et la scalabilité, et ce à cause de l'instabilité (la mobilité) des nœuds dans le réseau. En réalité, les protocoles déterministes existants ne fournissent en aucun cas la garantit de fiabilité. Les protocoles déterministes établissent des routes exactes pour faire passer les paquets de données à tous les membres du groupe multicast et avec un taux de données important. A cause des problèmes de congestion du réseau, de la panne de la couche liaison ou de mouvement des nœuds, les récepteurs multicast ne peuvent pas recevoir toutes les données correctement. Par contre, si on laisse le protocole multicast être probabiliste, en utilisant par exemple la méthode « Gossip-based » (qu'est une méthode probabiliste basé sur le bavardage), il sera possible qu'on atteint une bonne fiabilité et avec un taux de contrôle minimal.

L'idée principale de la méthode probabiliste basée sur le bavardage (Gossip-based method) dans le routage est qu'il n'y a pas de pré-établissements déterministes des routes, chaque nœud membre possède des routes vers d'autres nœuds membres. Par conséquent, lorsqu'un nœud membre du groupe multicast reçoit un message, il retransmet ce message vers une petite partie de ce groupe multicast. Après certaines itérations, ce message atteint tous les membres du groupe. Ce type de protocole utilise un mécanisme qui contrôle la redondance du message circulant entre les différents membres du groupe dans le but d'atteindre une grande fiabilité.

Des protocoles multicast basés sur cette méthode existent déjà pour les réseaux filaires et les réseaux ad hoc. Le nouveau protocole *RDG* est conçu dans le but de fournir une fiabilité probabiliste et pour des spécifications pratiques (selon les applications demandées). *RDG* utilise un système de bavardage pur, tous les messages de contrôle, tous les paquets de données, tous les accusés négatifs et toutes les informations vont bavarder. Ce protocole n'a pas besoin d'une primitive multicast au niveau de la couche réseau, il peut être développé sur la base de n'importe quel

protocole de routage réactif (comme le protocole *DSR* par exemple). Le protocole *RDG* possède trois différentes sessions :

- ***La session de jointure*** « Join Session », concerne un nœud désirant joindre le groupe multicast pour transmettre ou recevoir des données. Le nœud diffuse le message « *Group Request* » dans le réseau pour chercher les membres du groupe en question. Ce message est identique à celui de *DSR* sauf que le protocole *RDG* utilise l'identificateur du groupe multicast comme étant une adresse cible. Chaque membre du groupe multicast recevant le message « *Group Request* » met à jour sa structure « *View* » (une structure de données utilisée pour sauvegarder les informations de voisinage) en rajoutant l'identificateur de ce nouveau membre. Le membre récepteur répond ensuite par le message « *Group Reply* » et avec une probabilité « *P-Reply* ». Lors de la réception de cette réponse, le nœud initiateur mis à jour sa propre structure « *View* ». De cette façon, chaque membre du groupe peut avoir une vue partielle du groupe multicast ainsi que des routes lui permettant d'atteindre d'autres membres dans le groupe.

- ***Les deux sessions de bavarder / laisser*** (*Gossip/leave*) concerne le routage multicast. Chaque nœud possède un buffer pour sauvegarder ses données. Chaque nœud génère périodiquement le message « *Gossip* » à « F » autres nœuds choisis aléatoirement depuis sa structure « *View* ». Ce message « *Gossip* » inclue certaines données tirées depuis son propre buffer et contient aussi les données qu'il désire recevoir. Les données choisies vont être supprimées depuis le buffer après qu'elles soient bavardées « T_q » fois. Après, les membres recevant ce message de bavardage vont mettre à jour leurs propres buffers de données et répondent avec les données demandées par l'émetteur avec leurs propres messages « *Gossip* ».
  Quand un nœud membre désire quitter le groupe, il envoie un autre message « *Gossip* » de la même manière précédente, pour que tous ses voisins éliminent ses informations depuis leurs structures « View ». Quand la quantité d'information qui se trouve dans la structure « View » d'un membre s'abaisse en dessous d'un seuil prédéfini, ce membre doit réinitialiser la session de jointure « *Join-Session* ».

Même si ce protocole n'est pas déterministe mais probabiliste, à travers la méthode de bavardage utilisée « *gossip method* », il peut atteindre la fiabilité probabiliste avec uniquement une dégradation modeste, en cas de mobilité ou de scalabilité. Comme exemple à ce protocole, soit un groupe de 10 nœuds, 5 parmi eux sont des membres multicast (des récepteurs). On définit la valeur de « F » à 1, la valeur de « T_q » à 2. Le tableau 2.1 regroupe les structures « View » de tous les membres multicast à un instant donné. La figure 2.3 représente le processus de propagation d'une certaine donnée en utilisant la méthode de bavardage initiée par le nœud 1. On peut bien remarquer que même si les membres n'ont pas une vue globale de toutes les liaisons du groupe

multicast, tous les membre peuvent obtenir le paquet de donnée après exactement 3 tours.

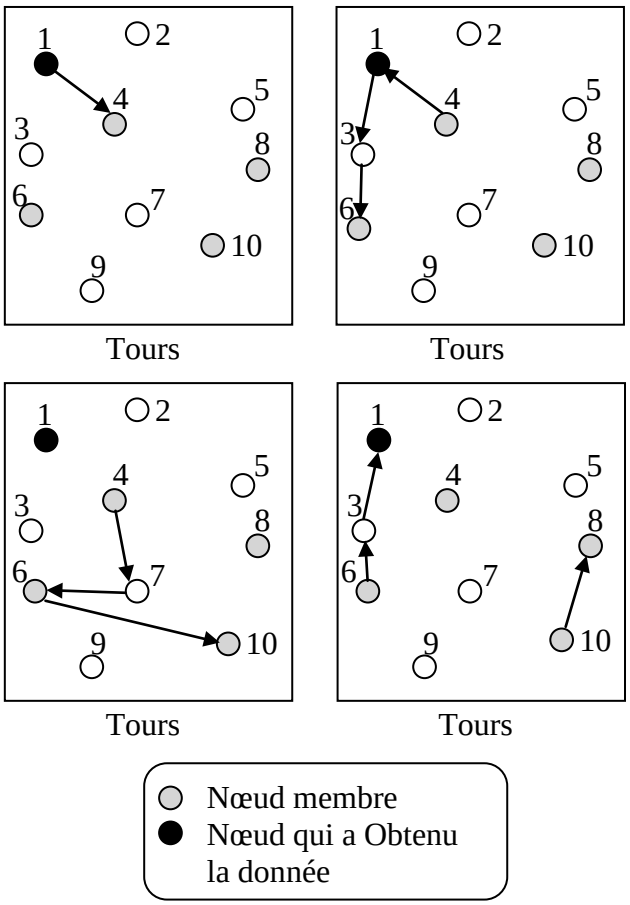


FIG 2.3 *– Fonctionnement du protocole probabiliste RDG*

V	1	4	6	8	10
1	★	★	★		
4	★	★	★	★	
6	★		★		★
8		★		★	★
10			★	★	★

TAB 2.1 *– Structures actives « View » des membres dans le protocole probabiliste RDG*

('★' se trouvant dans la zone $V_{i\ j}$ signifie que le membre 'j' est dans la structure « View » de 'i')

RDG utilise donc une technique de diffusion contrôlée par une méthode probabiliste, qui est la méthode de bavardage « Gossip method », pour délivrer les paquets de données aux membres du groupe multicast. C'est en quelque sorte un protocole qui utilise la technologie « Stateless multicasting » aussi.

2.4 Conclusion

Les protocoles multicast filaires ont été bien établis. Dans les réseaux mobiles ad hoc, concevoir un protocole de routage multicast est un vrai défi à cause de la nature dynamique, sans infrastructure des MANET. Malgré le nombre important de travaux qui ont été réalisés dans ce domaine, peu d'entre eux peuvent fournir un protocole multicast satisfaisant l'objectif général de routage multicast (qui marche avec le maximum d'applications).

Dans ce chapitre, j'ai commencé par donner un petit résumé sur le routage multicast dans les réseaux filaires ainsi que quelques protocoles multicast proposés dans ce type de réseau. Après, j'ai passé aux réseaux mobiles ad hoc où j'ai exposé les différents problèmes rencontrés lors de conception d'un protocole de routage multicast. Plusieurs structures de routage multicast ont été enchaînées l'une après l'autre en y expliquant leurs fonctionnements ainsi que leurs avantages et inconvénients. Ensuite, quatre nouvelles technologies ont été expliquées (*Overlay-based multicasting*, *backbone-based multicasting*, *stateless-based multicasting*, d'autres *protocoles d'extension*). Chacune de ces technologies a sa propre manière de sauvegarder les informations d'état ou de ne pas le faire. D'une façon générale, le but de toutes ces technologies consiste à minimiser la charge de contrôle afin d'améliorer plus la performance des protocoles de routage. En fin, j'ai donné une vue générale de deux protocoles multicast (*RALM* et *RDG*) qui ont imposés leurs propres efforts spéciaux dans l'aspect de fiabilité. Le premier utilise une méthode de fiabilité basée sur les accusés négative « *NACK* » lors de détection d'erreurs, en rajoutant des mécanismes de contrôle de congestion et des mécanismes de découverts d'erreurs. Le deuxième essai d'offrir la fiabilité à travers une méthode de fiabilité probabiliste pour remplacer la méthode déterministe, et ce afin d'atteindre une meilleur performance. Les futures travaux dans la conception des protocoles de routage multicast pour les réseaux ad hoc essaient tous de réduire la charge du contrôle générale dans le réseau afin d'offrir la scalabilité et la robustesse. Quelques idées principales visant à satisfaire les conditions de routage multicast ad hoc, comme la localisation des informations d'état, et les mécanismes de fiabilité, ont besoin d'études supplémentaires.

Chapitre 3

Quelques protocoles multicast dans les réseaux ad hoc

3.1 AMRoute: Ad hoc Multicast Routing protocol

AMRoute [77] représente une nouvelle approche de routage multicast IP robuste en exploitant les arbres de distribution multicast et les noyaux logiques dynamiques. AMRoute crée un arbre de distribution multicast partagé entre les émetteurs et les récepteurs uniquement, les liens de cet arbre sont des tunnels unicast bidirectionnels. Les autres nœuds du réseau ne sont pas concernés de cette distribution multicast et l'utilisation des tunnels unicast implique que la structure d'arbre n'a pas besoin de réajustements en cas de changement de la topologie du réseau. Par conséquent, le trafic de signalisation et la perte de données sont réduits. AMRoute n'est pas sensé de contrôler le mouvement du réseau et de sauvegarder les informations de voisinage, pour cela un protocole de routage unicast non spécifique doit être disponible afin d'accomplir ces tâches. Certains nœuds de l'arbre sont désignés par AMRoute comme étant les noyaux logiques, et ils sont responsables des opérations de contrôle comme la détection des membres du groupe multicast et la construction de l'arbre. Cependant, l'inconvénient majeur de AMRoute est la génération des boucles. AMRoute possède trois phases principales :

- ***Phase de création de la maille*** Au début, chaque membre du groupe se considère comme le noyau de son propre segment qui ne contient que lui. Chaque noyau diffuse $Join-\mathrm{Re}q\langle source\ IP, Multivast\ Group\ IP, Message\ id, TTL\rangle$ d'une manière périodique en augmentant la durée de vie (TTL) à chaque fois. Ce message permet de découvrir tous les noyaux qui se trouvent dans les autres segments du même groupe multicast. Quand les autres membres du groupe n'appartenant pas au même segment que l'émetteur reçoivent ce message, ils répondent tous par un autre message $Join-Ack\langle Source\ IP, Multicast\ Group\ IP, Message\ id, TTL\rangle$. Ainsi, un tunnel bidirectionnel est établit entre le noyau émetteur et chaque noeud récepteur pour former en fin une seule maille dans laquelle va coexister plusieurs noyaux. Pour cela un algorithme spécial nommé « *Core Resolution Algorithm* » est exécuté pour sélectionner un seul noyau pour la maille (un simple algorithme de résolution consiste par exemple à choisir le nœud qui possède la plus grande adresse IP).

Dans AMRoute, uniquement les nœuds du noyau ont le droit de diffuser les messages *Join-Req*, alors que la réponse *Join-Ack* peut être envoyé par tous les membres du groupe multicast. Ceci permet de limiter la consommation de la bande passante d'une part, et d'éviter l'établissement d'un nombre important de liens incidents au noyau logique sélectionné.

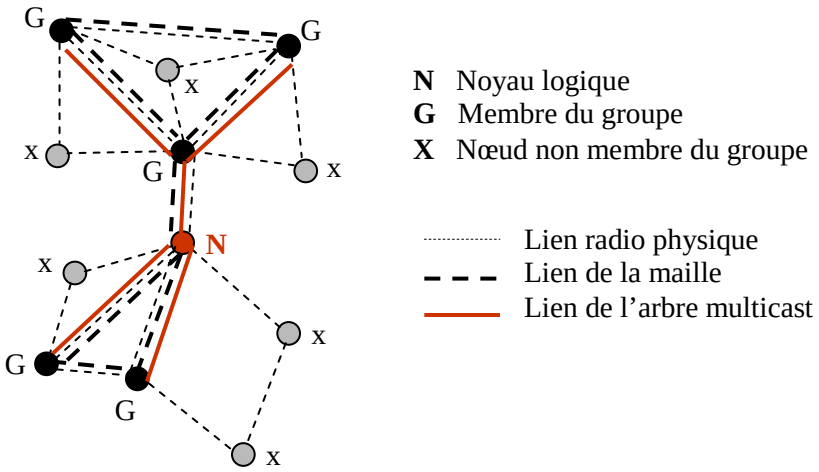


FIG 3.1 – *Création de la maille et de l'arbre dans AMRoute*

❖ ***Phase de création de l'arbre*** Le noyau de la maille est responsable d'initier la procédure de création de l'arbre. Pour cela, le noyau diffuse périodique le message $Tree-Create\langle Souce\ IP, Multicast\ Group\ IP, Message\ id, Seq.\#\rangle$ sur tous les tunnels bidirectionnels incidents. Uniquement les nœuds membres du groupe multicast (émetteurs ou récepteurs) ont le droit de rediffuser ce message vers leurs voisins dans la maille. A l'aide de ces messages l'arbre multicast est construit lien par lien. En effet, à la réception non dupliquée du message *Tree-Create* le membre du groupe récepteur relais ce message vers ses autres voisins via les tunnels de la maille et marque le lien source comme un lien appartenant à l'arbre de distribution (voir la figure 3.1). Quand un nœud reçoit Tree-Create dupliqué, il envoie le message $Tree-Create-Nack\langle Source\ IP, Multicast\ Group\ IP, Message\ id, Seq.\#\rangle$ sur le chemin antécédent pour qu'il l'enlève et le considère comme étant un lien de la maille uniquement. Le routage des données s'effectue via les tunnels formés entre les nœuds du groupe multicast uniquement.

❖ ***Phase de maintenance*** Si un nœud désire quitter le groupe multicast, il envoie le message $Join-Nak\langle Source\ IP, Multicast\ Group\ IP, Message\ id, TTL\rangle$ une unique fois vers ses voisins. S'il reçoit ultérieurement n'importe quelle donnée ou message de contrôle depuis ce groupe, il peut envoyer une autre fois le message *Join-Nak*.

3.2 AMRIS: Ad hoc Multicast Routing Protocol utilising Increasing id-numberS

AMRIS [78] est un protocole multicast désigné pour fonctionner indépendamment des protocoles de routage unicast. L'idée de ce protocole est d'assigner à la demande un identifiant unique à chaque nœud participant à la session multicast. AMRIS construit un arbre de routage multicast partagé entre les nœuds participants à la session multicast. L'arbre partagé a pour racine un noeud particulier appelé *Sid*. Ce dernier est la source ayant initié la session multicast (si la session possède plus d'une source, le Sid sera tout simplement élu depuis l'ensemble des sources). Les identifiants des autres nœuds du réseau sont incrémentés à chaque fois que l'on s'éloigne de la racine *Sid* ; de cette façon le *Sid* aura toujours la plus petite valeur numérique. Les identifiants vont permettre aux nœuds de joindre ou quitter la session multicast sans perturber la structure de l'arbre, ainsi la maintenance des liens cassés est effectuée localement sans avoir besoin d'une autorité centrale. AMRIS possède deux phases principales :

- ***Phase d'initialisation de l'arbre*** Initialement, le nœud *Sid* diffuse le message $New-Session\langle msm-id, Multicast\ Session\ id(adr\ \text{IP de}\ classe\ D), Routing\ Tables\rangle$, tout nœud recevant ce message génère son propre msm-id en calculant une valeur supérieur et non consécutive à celle de l'émetteur, ainsi il y aura des décalages entre les msm-id de l'émetteur et des récepteurs. Les récepteurs remplacent la valeur de msm-id de l'émetteur par leur msm-id, ainsi que certains métriques de routage dans le message *New-Session (NS)* avant de le rediffuser à leur tour. Un nœud peut recevoir plusieurs messages *NS*, afin de pouvoir traiter chacun de ces messages avant de le rediffuser, une gigue (une certaine période) est introduite entre la réception et la rediffusion de ce message (voir la figure 3.2).

  AMRIS maintient aussi une table d'état des voisins telle que $Neighbor-StatusTable\ \langle\ Neighbor\ unique\text{-}id\ (IP), msm-id, \text{Re}lation\ (père/fils)\ \text{Re}maining\ Timeout\ Value, Routing\ Metric\ \rangle$ Cette table est mise à jour par le contenue des messages *NS*. Chaque nœud diffuse une balise périodique contenant $\langle\ Node\ Unique\text{-}id\ (IP), msm\text{-}id\ \&\ Status\ (member/non\ member), Parent\ msm\text{-}id, Child\ msm\text{-}id\ \rangle$. Cette balise permet de signaler la présence de ce nœud à ses voisins, et l'absence de celle-ci permet de détecter les ruptures de liens et donc un changement de topologie.

  Un nœud N voulant joindre le groupe multicast, unicast le message *Join-Req* à l'un de ses parents potentiels (un parent potentiel est tout voisin possédant une valeur msm-id inférieur à celle de N). Quand ce parent potentiel P reçoit *Join-Req*, il répond par un autre message *Join-Ack* s'il est déjà membre de la session multicast ou bien il a réussit à s'accrocher à un parent potentiel à lui par le même procédé. Sinon, il envoie le message *Join-Nack*. Le nœud N qui ne trouve pas un parent

potentiel lui permettant de joindre la session multicast par ce procédé, utilise un autre mécanisme de construction de branche appelé *BR* (*Branch Reconstruction*). Ce mécanisme consiste à diffuser un message *Join-Req* avec un nombre de sauts limité à R. Les voisins recevant ce message le transforment en un message *Join-Req-Passive* avant de le rediffuser à son tour. Lorsqu'un nœud appartenant à l'arbre multicast reçoit ce message, il répond par le message *Join-Ack-Passive*. Le nœud N peut recevoir plusieurs messages de ce type donnant une possibilité de plusieurs branches. Il choisit alors le meilleur chemin (en se basant sur les métriques de routage) et renvoie un message *Join-Conf* pour activer les liens menant à l'arbre. Ce processus est répété en incrémentant R à chaque fois jusqu'à l'obtention d'une réponse ou à l'expiration du délai de recherche.

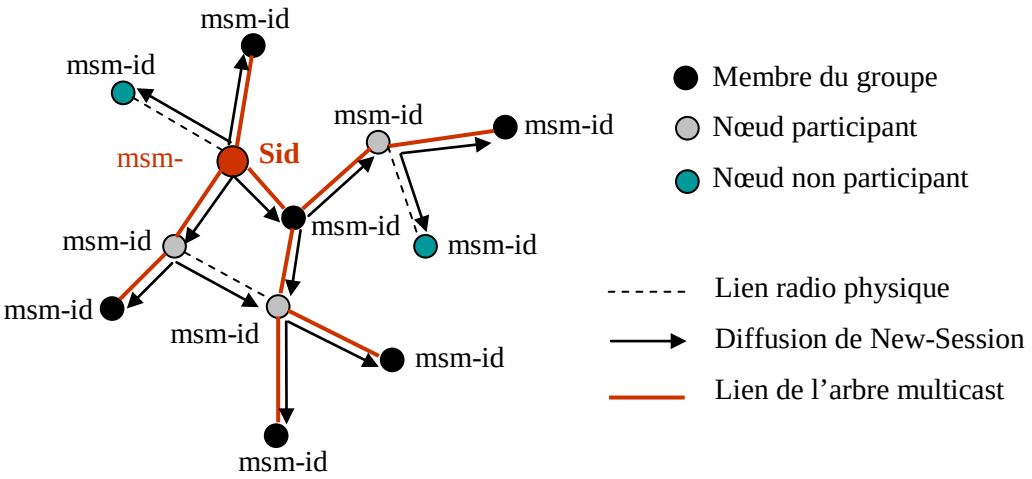


FIG – 3.2 *Création de l'arbre dans AMRIS*

❖ ***Phase de maintenance de l'arbre*** Ce mécanisme fonctionne continuellement pour d'assurer que les nœuds restent connectés à l'arbre multicast durant toute la session de communication. Dans le cas de changement de topologie ou de rupture d'un lien, la transmission de données est interrompue et le nœud possédant la plus petite valeur msm-id (le fils) est responsable de la reconstruction de ce lien. Dans le cas où l'on trouve un parent potentiel distant d'un seul saut et membre de l'arbre multicast alors, il sera très simple de reconstruire le lien endommagé. Dans le cas contraire, le fils exécute le mécanisme de reconstruction de branche (BR).

3.3 ODMRP: On-Demand Multicast Routing Protocol

ODMRP [92] est un protocole de routage multicast basé sur l'utilisation d'une maille reliant tous les membres et toutes les source d'un groupe multicast. Cette maille sert à acheminer les paquets de données via la diffusion fournissant ainsi des chemins redondants. Ce protocole ne maintient pas les informations d'états du réseau, mais il applique à la demande des procédures de création et de maintenance des routes suite à une phase de requête. ODMRP peut coexister avec des protocoles de routage unicast comme par exemple DSR [45], ZRP [93], TORA [94], AODV [44], et ABR [95]. Il peut fonctionner aussi très efficacement en tant que protocole unicast. Ainsi, un réseau ad hoc équipé d'ODMRP n'a pas besoin d'un autre protocole de routage unicast. ODMRP convient plus avec les réseaux ad hoc caractérisés par une bande passante limitée, une topologie qui change fréquemment, et une capacité modeste.

- ***La création de la maille*** La création de la maille se base essentiellement sur les deux actions suivantes :

  - ***Join-Request*** Lorsqu'une source multicast désire envoyer des données vers un groupe multicast, elle diffuse périodiquement un message *Join-Request* $\langle SourceIP, Multicast\ Group\ IP, \Pr evious\ hop\ IP, Seq.\#, TTL, Hops \rangle$. Cette transmission périodique permet de rafraîchir les liaisons entres les voisins et de mettre à jour les routes établies. Les nœuds recevant ce type de message mettent à jour les entrées de leur *cache* $\langle Multicast\ Group\ IP, Source\ IP, Seq.\#, \Pr evious\ Hop\ IP \rangle$, puis chaque nœud vérifie si le message *Join-Request* reçu est dupliqué ou pas en comparant (*Source IP* et *Seq.#)* dans son cache. Si le paquet est dupliqué alors il sera immédiatement refusé, sinon une nouvelle entrée est crée dans le cache du récepteur. Après avoir incrémenté la valeur de « *Hops* », si cette nouvelle valeur ne dépasse pas la valeur du *TTL* alors, le récepteur met son adresse IP dans le champ « *Previous hop IP* » et rediffuse le paquet à son tour. Lorsque le message atteint un récepteur multicast, ce dernier insert/ met à jour l'information dans une table *Member-Table* $\langle Multicast\ Source\ IP, Time\_Stamp) \rangle$ que maintient chaque membre du groupe multicast. (Cette table permet au récepteur multicast de vérifier l'expiration de la source, si aucun message *Join-Request* n'est reçu depuis une source existant dans *Member-Table* dans un délai *Mem-Timout* alors, cette source sera éliminée de la table). Le récepteur multicast génère ensuite la réponse *Join-Reply*.

  - ***Join-Reply*** Le récepteur multicast diffuse *Join-Reply* $\langle Multicast\ Group\ IP, Sender\ IP, Next\ Hop\ IP \rangle$ après chaque période *Join-Reply-Refresh* et tant qu'il y a des entrées dans sa table *Member-Table*. Quand un nœud reçoit *Join-Reply*, il

vérifie si la valeur de *Next-Hop-IP* est identique à son IP. Si c'est le cas, le nœud réalise alors qu'il est sur le chemin menant à la source et faisant ainsi membre du groupe d'acheminement. Ce nœud positionne ensuite son flag *Forwarding-Group-Flag* (FG-Flag) et diffuse son propre message *Join-Reply*, le champ *Next-Hop-IP* porte la valeur de *Previous-Hop-IP* du message Join-Request reçu. De cette façon, chaque membre de groupe d'acheminement diffuse *Join-Reply* sur le chemin le plus court jusqu'à ce qu'il atteint la source multicast. Ce processus permet donc d'installer ou mettre à jour la route liant la source avec chaque récepteur multicast formant ainsi une maille de nœuds (les nœuds de groupe d'acheminement). La figure 3.3 donne un exemple de maille conçu par ODMRP. Les données sont routées via les noeuds du groupe de routage offrant ainsi des routes redondantes en cas de rupture des liens intermédiaires [96].

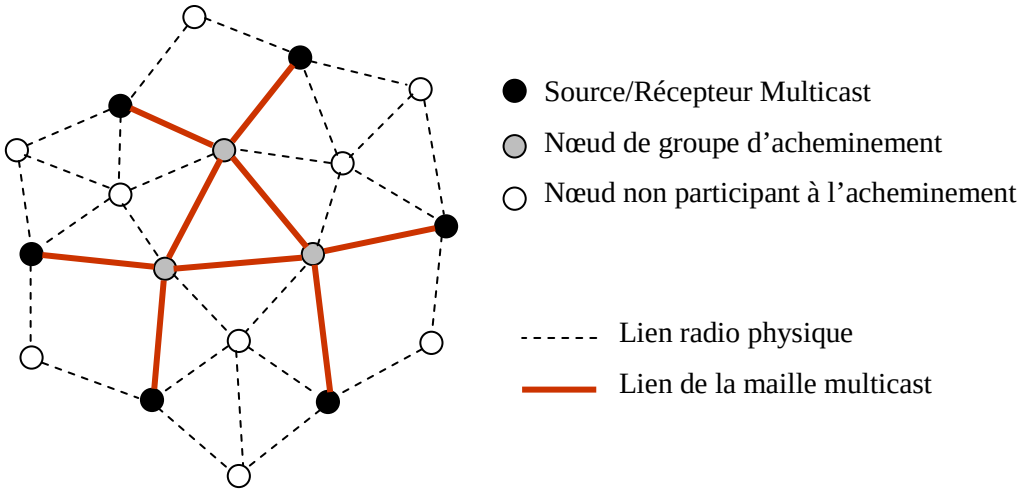


FIG 3.3 – *Exemple de maille dans ODMRP*

❖ ***Maintenance de la maille*** Il n'existe pas un paquet de contrôle précis à envoyer pour qu'un nœud puisse quitter le groupe multicast. Si une source multicast désire quitter le groupe, elle s'arrête tout simplement d'envoyer les paquets *Join-Request* tant qu'elle n'a pas de paquets de données à envoyer dans le groupe. Si un récepteur multicast désire quitter un groupe donné, il enlève l'entrée correspondante depuis sa table *Member-Table* et n'envoie plus de message *Join-Reply* à ce groupe. Les autres nœuds d'acheminement peuvent quitter le groupe multicast aussi sans devoir envoyer un message de contrôle, leur départ peut être détecté par les messages de rafraîchissement périodiques.
ODMRP utilise la prédiction de la mobilité permettant aux sources de reconstruire les routes avant qu'elles ne soient rompues.

3.4 Patch-ODMRP

Patch-ODMRP [97] est une extension du protocole multicast ODMRP. Dans ODMRP, les nœuds d'acheminement représentent les nœuds se trouvant sur le chemin le plus court entre la source et les membres du groupe multicast, ces nœuds forment ce qu'on appelle la maille d'acheminement. Quand le nombre de sources dans le groupe multicast est trop petit, la maille d'acheminement est toujours formée d'une manière dispersée et les chemins redondants peuvent ne pas être disponibles. Pour cela, il est nécessaire de reconfigurer fréquemment la maille, ce qui résulte en une grande charge de contrôle. Pour résoudre ce problème, la nouvelle extension Path-ODMRP propose un système de rapiècement local (*Local Patching*) au lieu des fréquentes reconfigurations de la maille. Dans Patch-ODMRP, chaque nœud du groupe d'acheminement vérifie continuellement s'il y a des symptômes de détachement de la maille au niveau de son entourage. Quand un nœud d'acheminement trouve un tel symptôme, il essaie de s'accrocher à la maille via un chemin temporaire en diffusant localement certains messages de contrôle.

Dans Patch-ODMRP, chaque nœud d'acheminement vérifie localement la possibilité de se détacher de la maille en utilisant une balise de la couche MAC (définie dans IEEE 802.11 [98]), dans le but de vérifier son voisinage. En comparant cette information avec celle qui se trouve dans le champ « *Previous Hop IP* » de la Table *Forwarding-Group-Table* <*Multicast Group IP, Source IP, TTL, Previous Hop IP*>, ce dernier sera alors capable de détecter si son nœud d'acheminement prédécesseur a perdu contact avec lui ou pas. Si c'est le cas, le nœud détectant commence la procédure de rapiècement en diffusant un message de contrôle « *Advertisement Message (ADVT)* » pour annoncer qu'il a perdu contact avec son prédécesseur. Ce message contient des champs spécifiant « *Multicast Group IP », « Source IP », et « Hop Count »* qui sépare le nœud émetteur de la source multicast. Le message *ADVT* est envoyé avec un nombre de sauts limité (2 ou 3 sauts) dans le but de localiser ou borner sa diffusion.

Quand un nœud reçoit le message *ADVT*, il mis à jour l'information de routage du nœud émetteur. En plus, s'il est un nœud d'acheminement servant la même source multicast que l 'émetteur, et s'il est proche de la source multicast spécifiée que l'émetteur lui même (en vérifiant le champ « *Hop Count* » inclus dans *ADVT*), il génère le paquet « *Patch* » et le renvoie vers le nœud émetteur de *ADVT*. Si par contre le nœud récepteur n'est pas qualifié pour connecter l'émetteur d'*ADVT* au groupe d'acheminement, il achemine ce paquet vers ses voisins à condition que la valeur de *TTL* soit supérieure à zéro.

L'émetteur du message *ADVT* peut recevoir plusieurs messages *Patch*. Il choisit alors le chemin le plus court qui mène à la source et qui a été traversé par le message Patch choisi. La figure 3.4 montre exemple de fonctionnement de la procédure de rapiècement. Quand le nœud A perd contacte avec le nœud B, il diffuse le message *ADVT* dans son premier voisinage (vers C, D, E). Uniquement le nœud E peut répondre

en envoyant le message *Patch*, car ce dernier possède une valeur de « *Hop Count* » inférieur à celle du nœud A (c'est à dire que E est plus proche de la source que l'émetteur de *ADVT* lui même). Les nœuds C et D ne peuvent pas répondre car ils ont une valeur de « *Hop Count* » supérieure à celle de A. Et donc, A peut continuer à recevoir ses paquets de données depuis son voisin E au lieu de B.

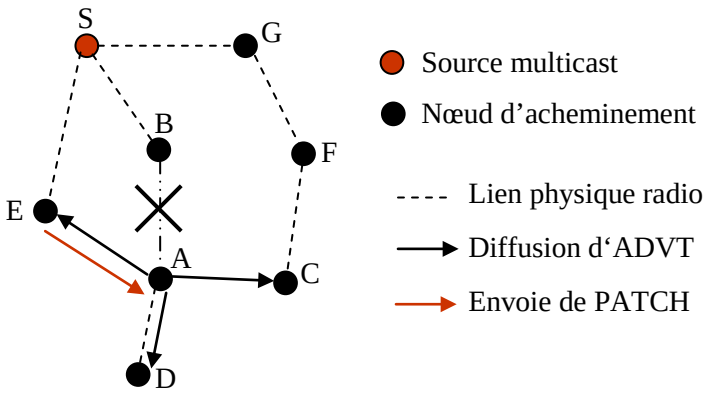


FIG 3.4 – *Exemple de la procédure Patching*

Dans Patch-ODMRP, le nœud successeur initialise la procédure de rapiècement, si la procédure échoue, les sources multicast n'auront aucune idée sur la panne, et ainsi elles ne sont pas capables de réagir convenablement contre cette panne. La procédure de rapiècement introduit aussi de nouveaux nœuds dans le groupe d'acheminement, l'augmentation de ces nœuds permet d'augmenter le nombre de paquets de données acheminés dans la maille d'acheminement. En comparant Patch-ODMRP avec ODMRP, les résultats de simulation ont montré que la procédure de *Patching* améliore le taux de distribution de paquets de données (c'est à dire la robustesse), et réduit la charge de contrôle. Il a été aussi montré que la performance est beaucoup plus large quand le degré de mobilité des noeuds est important mais par contre, l'efficacité de transmission se dégradée sévèrement.

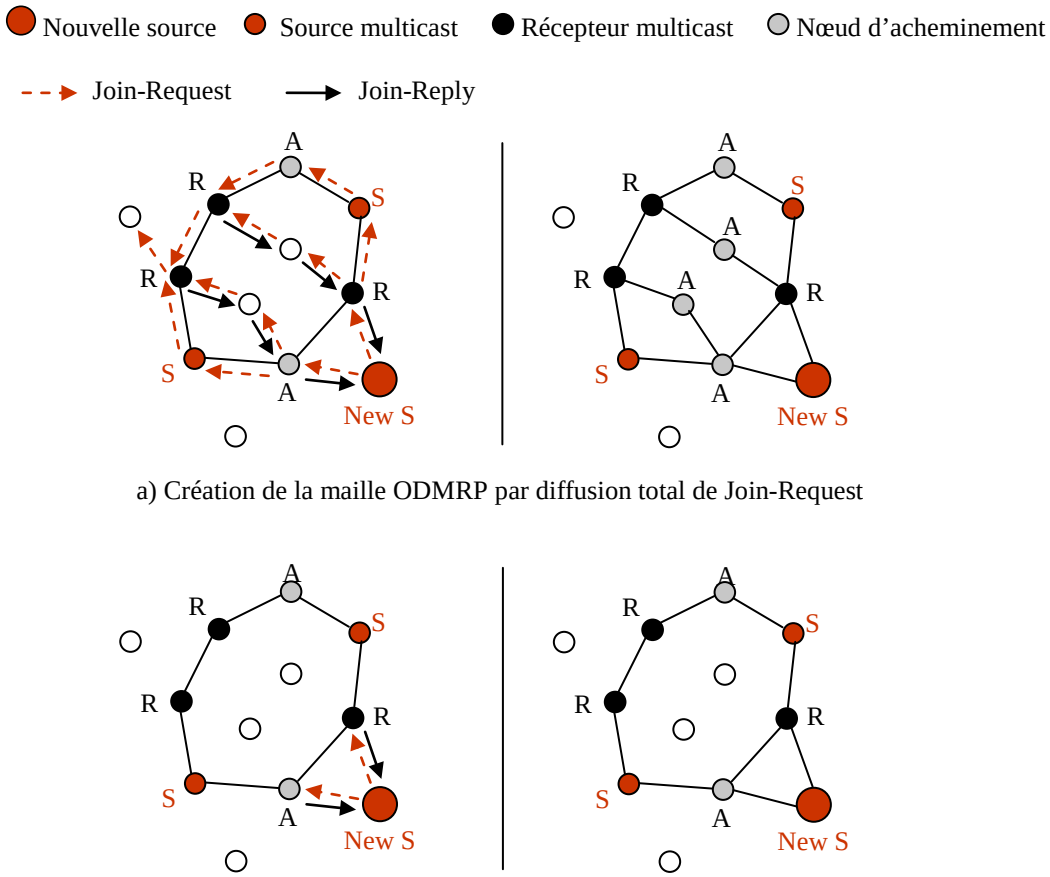
3.5 PE-ODMRP: Performance Enhanced ODMRP

PE-ODMRP [99] est une extension du protocole multicast ODMRP. Malgré que le protocole ODMRP est simple et robuste contre la mobilité, lorsque le nombre de sources multicast de la maille d'acheminement est important, ce protocole souffre beaucoup des charges de contrôle excessive et des transmissions de données redondantes. Ce qui peut causes la collision, la congestion et la contention de canal. Dans le but de résoudre ces problèmes, la nouvelle solution PE-ODMRP tente de renforcer la performance du protocole ODMRP. PE-ODMRP limite la zone de transmission des messages de contrôle en établissant une maille d'acheminement dispersée. A travers la comparaison de performance des deux protocoles ODMRP et PE-ODMRP, les résultats de simulation ont montrés que PE-ODMRP réduit la charge du contrôle et la transmission des données redondantes.

De la même manière que ODMRP, PE-ODMRP se compose de deux actions: l'action d'interrogation et l'action de réponse. Cependant, PE-ODMRP réduit la charge de contrôle via la limitation de la zone de transmission de la requête Join-Request diffusée périodiquement par chaque source multicast. PE-ODMRP utilise pour cela deux drapeaux en même temps dans sa table *Forwarding Group Table*, il s'agit de *Query Blocking Flag (QB-Flag)* et de *Query Blocking Timer (QB-Timer)*. Quand un noeud recevant *Join-Reply* depuis un voisin, il vérifie si son adresse IP correspond à la valeur de champ « *Next-Hop-IP* » du message reçu, si c'est le cas alors, ce dernier positionne son *FG-Flag* et son *QB-Flag* à 1. Si un nœud dont le *QB-Flag* est positionné à 1 reçoit le message *Join-Request* initié par une source du même groupe multicast, il génère immédiatement la réponse *Join-Reply* et l'envoie vers la source au lieu de rediffuser la requête *Join-Request* vers ses voisins. Notant que tous les nœuds de la maille peuvent générer et envoyer Join-Reply dans PE-ODMRP. L'autre drapeau *QB-Timer* est initialisé à une valeur « *Mesh-Expiration-Interval* » lorsque le drapeau *QB-Flag* correspondant est mis à 1, et il s'arrête après l'expiration de cette valeur. Pour rafraîchir la maille d'acheminement à causes des changements fréquents de la topologie, la valeur« *Mesh-Expiration-Interval* » doit être plus petite que la période de rafraîchissement de route.

Lorsqu'une maille d'acheminement est établi par une source multicast, elle peut être remployée par d'autres sources du même groupe multicast. Ainsi, une source n'est pas forcée de s'accorder à toutes les destinations multicast via les plus courts chemins de la maille d'acheminement. PE-ODMRP permet aux sources de construire des arbres les connectant non pas à toutes les destinations mais à quelques noeuds de la maille construite précédemment par d'autres sources multicast. L'union de tous ces arbres constituent une maille plus dispersée par rapport à celle d'ODMRP. La figure 3.5 montre un exemple de création de la maille d'acheminement dans le protocole ODMRP et dans le protocole PE-ODMRP. Dans le premier cas, la requête *Join-Request* est diffusée dans la totalité du réseau, dans le deuxième cas par contre, cette même requête est envoyée vers deux nœuds uniquement qui sont des membres de la maille

d'acheminement. Cette dernière maille peut être donc remployée par la nouvelle source pour la distribution des ses paquets de données vers toutes les destinations multicast.



a) Création de la maille ODMRP par diffusion total de Join-Request

b) Création de la maille PE-ODMRP par diffusion locale de Join-Request

FIG 3.5 – *Comparaison entre la création de la maille d'ODMRP et de PE-ODMRP*

Le groupe d'acheminement de PE-ODDMRP peut ne pas fournir des chemins optimaux entre les sources et les destinations multicast, car les destinations ne construisent pas forcément des arbres de plus court chemin. En outre, la maille est reconfigurée à chaque intervalle de rafraîchissement de routes. Pour ces raisons, la gigue séparant l'envoie des paquets de données dans PE-ODMRP doit être supérieur à celle dans ODMRP.

3.6 ODMRP-MPR: ODMRP with Multipoint Relay

ODMRP-MPR [100] représente une nouvelle extension du protocole ODMRP en introduisant une technique limitant le rang de diffusion qui s'appelle la technique MPR (*Multipoint Relay*). Cette technique est utilisée pour réduire la charge de contrôle, assurer une haute scalabilité, et résoudre efficacement le problème des liens unidirectionnels des communications sans fil. PE-ODMRP préserve aussi toutes les qualités d'ODMRP comme par exemple le taux de données élevé et l'efficacité d'énergie même en cas de changement fréquent de topologies.

❖ ***La technique MPR et l'algorithme de sélection*** Chaque nœud du réseau diffuse périodiquement un message de contrôle *Hello* contenant la liste des voisins à partir desquels ce noeud peut recevoir des paquets. Quand un nœud N reçoit un message *Hello* d'un voisin M et dont la liste des voisins contient l'adresse de N, ce dernier rajoute l'adresse de M à sa propre liste des voisins et valide le drapeau « *Hearable Flag* » correspondant à M (M s'appel voisin directionnel dual de N). A travers les messages *Hello,* N peut facilement obtenir la liste de ses voisins distants d'un certain nombre de sauts (2- sauts par exemple).
Il existe des algorithmes de sélection des voisins MPR basés sur des heuristiques très complexes [36], et qui peuvent offrir des ensembles MPR presque minimaux. Selon la conception du protocole ODMRP-MPR, l'algorithme de sélection correspondant consiste à former la liste MPR à partir des voisins les plus récents dans le réseau en préservant la redondance dans cette liste MPR. Prenant par exemple un nœud x désirant former sa liste MPR en utilisant l'algorithme de sélection suivant:

1) Initialiser la liste MPR(x) à vide.
2) Mettre en ordre tous les voisins directionnels duaux de x par rapport au temps de réception des derniers messages *Hello*. Le dernier message *Hello* prend la première position dans la séquence des listes de voisins S(x) = {N1, N2, ...Nk}. Supposant que les voisins de x distants de deux sauts forment l'ensemble Q(x).
3) Sélectionner après la première liste de voisins, soit Ni ($i \in \{1, 2, 3, \ldots k\}$) depuis S(x). Si un voisin de la liste Q(x) correspond à un élément de la liste Ni alors, ajouter tous les éléments de Ni à la liste MPR(x) et supprimer ces éléments de la liste Q(x). Supprimer aussi la liste Ni de S(x).
4) Si S(x) ou Q(x) est vide alors, l'algorithme s'arrête et MPR(x) contient les listes désirées. Sinon, refaire ces opérations depuis le point 2) et continuer les calculs.

La liste MPR(x) est aussi comprise dans le message *Hello*. Ainsi, chaque élément de la liste des voisins de x est soit un voisin MPR(x) (s'il est dans la liste MPR(x)), soit un voisin Non-MPR de x (s'il n'est pas dans la liste MPR(x)). Les autres noeuds qui ne sont pas dans la liste des voisins de x peuvent recevoir les messages *Hello* de x. Ils ne valident aucun drapeau, mais rediffusent tous les paquets de x, comme le font les éléments de MPR(x).

❖ ***La création de la maille et des routes multicast*** ODMRP-MPR crée une maille de distribution de données suivant la demande des sources multicast. Comme le montre la figure 3.6, chaque source S désirant envoyer des paquets de données diffuse périodiquement la requête *Join-Request* en utilisant la technique de diffusion MPR. Cette requête contient le champ *Query-Seq* servant à détecter les duplications et le champ *Hop-Count* dénotant la distance de transmission. ODMRP-MPR contient aussi un paramètre spécial *Flood-Freq* (>=1). La source S diffuse le message *Join-Request* dans la totalité du réseau après avoir diffusé ce même message un certain nombre de fois identique à la valeur du paramètre *Flood-Freq* mais avec la diffusion MPR.

Quand un nœud N reçoit *Join-Request* envoyé par la source S depuis son prédécesseur F, il rafraîchit sa table de routage unicast si nécessaire et retransmet cette requête, à son tour. L'algorithme choisit ensuite les liens duaux les plus courts et les plus récents de N comme étant des routes unicast, en se basant sur les valeurs de *Hop-Count* et *Query-Seq* du message *Join-Request* reçu. Cet algorithme vérifie si F est un voisin dual de N. Si N n'appartient pas à l'ensemble des voisins Non-MPR de F (incluant même les voisins MPR et les nœuds qui n'appartiennent pas à la liste des voisins de F) et si le message *Join-Request* n'est pas dupliqué alors N le retransmet.

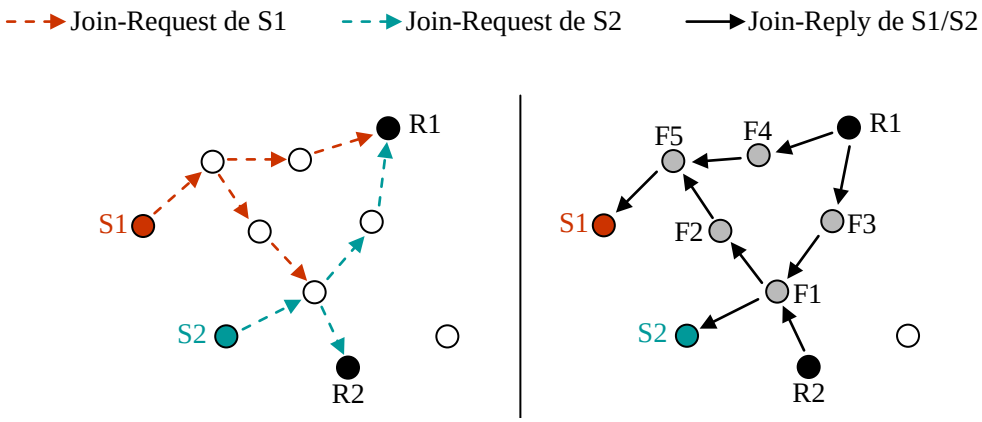
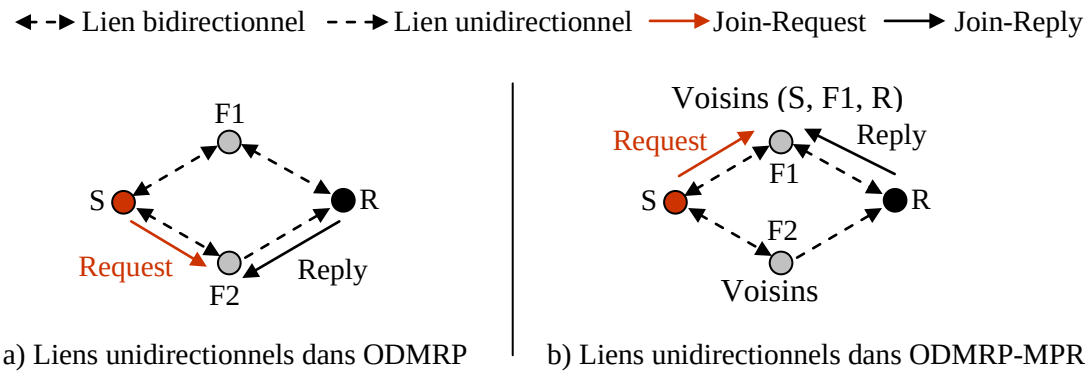


FIG 3.6 – *Création de la maille d'acheminement dans ODMRP-MPR*

Quand un récepteur multicast R reçoit *Join-Request*, il ajoute la source à sa liste des sources multicast. R diffuse périodiquement le message *Join-Reply* vers ses voisins, ce message contient la liste des sources multicast du réseau, et le saut suivant menant à chaque source. Quand un voisin reçoit *Join-Reply*, il vérifie si la valeur du champ *Next-Hop-IP* correspond à son adresse IP. Si c'est le cas, le voisin réalise qu'il est sur le chemin qui mène à la source et valide ensuite son drapeau *FG-Flag* (il devient un membre du groupe d'acheminement), puis il diffuse son propre message *Join-Reply*. Ce message est propagé par les membres du groupe d'acheminement jusqu'à ce qu'il atteint la source multicast via le chemin le plus court. Ce processus permet donc de construire/mettre à jour les routes multicast entre les sources et les récepteurs multicast pour construire en fin la maille d'acheminement.

❖ ***Maintenance de la maille*** Dans ODMRP-MPR, aucun message de contrôle n'est envoyé pour qu'un nœud puisse joindre ou quitter le groupe. Si une source multicast désire quitter la maille, elle s'arrête simplement d'envoyer les messages *Join-Request*. Ainsi, et après quelque temps, cette source ne va apparaître dans aucun message *Join-Reply* au tant que source multicast. Si un récepteur désire quitter un groupe multicast particulier, il n'envoie plus de messages *Join-Reply* à ce groupe. Les nœuds de groupe d'acheminement sont dénotés par Non-nœuds d'acheminement s'ils ne sont pas rafraîchis durant une certaine période (ils ne reçoivent plus de message *Join-Reply*).

Des analyses qualitatives ont été effectués sur ce protocole ont permis d'étudier la charge de contrôle, la redondance des messages *Join-Request* et de résoudre en plus le problème des liens unidirectionnels. Avec les informations de voisinage, les nœuds peuvent connaître ses voisins directionnels duaux et ses voisins unidirectionnels. Ainsi, ces noeuds peuvent éviter de sélectionner les liens unidirectionnels pour l'envoie des réponses *Join-Reply* et la distribution des paquets de données (voir la figure 3.7).



a) Liens unidirectionnels dans ODMRP b) Liens unidirectionnels dans ODMRP-MPR

FIG 3.7*- Résolution du problème des liens unidirectionnels dans ODMRP-MPR*

D'après la figure 3.7, F2 et R sont liés entre eux par un lien unidirectionnel (F2 peut envoyer des paquets vers R mais le contraire est faux). Dans ODMRP (Figure 3.7 (a)) R envoie sa réponse *Join-Reply* vers F2, ce qui résulte en une mauvaise création de la maille d'acheminement. Dans ODMRP-MPR, F2 diffuse ses messages *Hello* sans inclure R dans sa liste des voisins, ainsi R apprend qu'il y a un lien d'un seul sens avec F2 et sélectionne alors F1 comme nœud prédécesseur menant à la source S (Figure 3.7 (b)). La maille d'acheminement dans ce cas ne souffre pas des troubles des liens unidirectionnels. La détection des liens unidirectionnels n'est pas trop exacte, et à cause de l'intervalle d'envoie des messages *Hello* et de leurs pertes, les nœuds préférent les liens bidirectionnels stables et peuvent utiliser les liens unidirectionnels détectés comme des liens alternatifs aux liens bidirectionnels non disponibles.

D'après les résultats de simulation, OMRP-MPR hérite beaucoup de qualités de OMRP, améliore la scalabilité et traite bien le problème de liens unidirectionnels. En outre ODMRP-MPR peut coopérer avec un autre protocole unicast comme OLSR.

3.7 ODMRP-GPS

L'article [92] présente une méthode permettant l'adaptation de la fréquence de diffusion de *Join-Request* via un mécanisme de prédiction de mobilité dans lequel tous les nœuds sont supposés équipés d'un système de positionnement global GPS (*Global Positioning System*) [a]. Le GPS permet de fournir les informations de position et de mouvement de chaque nœud, ces informations servent à prédire la durée pendant laquelle les routes d'acheminement demeurent valides. Ainsi, lorsque la destruction d'une route est sur le point de se produire, la source multicast commence à diffuser ses paquets Join-Request. Dès qu'une liaison entre deux nœuds du groupe d'acheminement soit sur le point de se détruire, tout le chemin correspondant devient invalide et nécessite par conséquent une reconstruction.

Quand une source multicast envoie *Join-Request*, elle ajoute à ce denier l'information de sa position, sa vitesse et sa direction. Puis, elle initialise un nouveau champ *Min-LET* (*Minimum Link Expiration Time*) à une valeur prédéfinie *Max-LET-Value*. Lors de réception de Join-Request, chaque voisin distant d'1 saut prédit le temps d'expiration du lien (*LET*) qui le relie avec son prédécesseur, il calcule pour cela sa vitesse relative et sa distance de son prédécesseur. La valeur minimale entre ce temps d'expiration (*LET*) calculé et celui de champ *Min-LET* du paquet reçu sera rajouté au paquet du voisin. Quand un récepteur multicast reçoit Join-Request, il prédit à son tour la valeur *LET* du dernier lien avec son prédécesseur. La valeur minimale, entre *LET* calculée et le *Min-LET* reçu, occupe un nouveau champ *RET* (*Route Expiration Time*) ou le temps d'expiration de la route entre ce récepteur multicast et la source initiatrice.

Un récepteur multicast peut recevoir plusieurs *Join-Request* depuis la même source multicast, il peut choisir donc la route la plus stable qui le relie avec cette source. Pour cela, il doit attendre une certaine période du temps pour qu'il puisse recevoir le maximum nombre de messages *Join-Request* associés de leur *LET*, et de décider par la suite laquelle des routes sera utilisée pour envoyer ses données vers la source. Le choix de la route se base essentiellement sur la valeur *LET* des *Join-Request* reçus, celui possédant la plus grande valeur représentera la route la plus stable menant à la source ; le récepteur diffuse alors la réponse *Join-Reply*. La valeur *REL* associée sera ajoutée à cette réponse.

Un nœud d'acheminement peut recevoir plusieurs réponses *Join-Reply* possédant différentes valeurs *RET*, ce nœud d'acheminement choisit alors celui qui possède la plus petite valeur *RET*. Comme ça, la source peut construire rapidement une nouvelle route alternative à celle qui soit sur le point de se détruire via la diffusion de message *Join-Request*. Bien sûr, la borne minimale et la borne maximale de la diffusion de *Join-Request* doivent être prédéfinies pour éviter ainsi un rafraîchissement assez rapide ou assez long de la route qui est sur le point de se détruire.

Dans le but d'obtenir une meilleure route, presque tous les nœuds d'acheminement se trouvant sur le chemin liant le récepteur multicast à la source doivent attendre une certaine période du temps. Cette période sert à réduire l'acheminement redondant des messages *Join-Reply* associés de différentes valeurs *RET*. La source aussi doit attendre une certaine période pour qu'elle puisse recevoir toutes les réponses *Join-Reply* possibles et de choisir enfin celle qui possède la plus petite valeur *RET*.

La destruction d'un lien affecte la validité de toute la route associée. Dans la pratique, le choix des routes d'acheminement basé sur la mobilité relative des noeuds peut ne pas aider en grande chose dans le sens d'ajuster la fréquence d'envoie de *Join-Request*. En effet, la fréquence d'envoie des messages *Join-Request* est toujours importante à cause de la mobilité aléatoire des nœuds dans les réseau.

La mobilité et l'information de localisation d'un nœud peuvent être fournies par le GPS ou un autre équipement. Cependant, si le service de localisation est inclus, des charges de contrôle supplémentaires permettant de trouver les localisations des récepteurs seront ajoutées. Ces charges deviennent importantes si le nombre des récepteurs est grand. Aussi, la propagation de l'information de position de chaque nœud individuel à travers le réseau produit de grandes charges. Par conséquent, l'utilisation de l'information de mobilité relative semble d'être une manière simple et efficace pour aborder la mobilité propre entre les nœuds.

3.8 DCMP: Dynamic Core based Multicast routing Protocol

DCMP [89] est une autre extension du protocole ODMRP qui tente de réduire la charge de contrôle en classifiant dynamiquement les sources dans les catégories « Passive » ou « Active ».

Les sources multicast de DCMP sont rangées dans trois catégories ; les sources actives, les sources passives et les sources actives du noyau. Les sources actives sont similaires aux sources multicast d'ODMRP, dans le sens où elles diffusent régulièrement les paquets de contrôle *Join-Request*. Les sources actives du noyau (les nœuds u noyau) sont celles qui agissent comme des nœuds du noyau pour le compte d'une ou plusieurs sources passives. Les nœuds du noyau sont responsables de créer la maille multicast. A chaque fois qu'une source passive désire envoyer ses paquets multicast, elle les envoie vers le nœud du noyau associé à elle qui, à son tour, le renvoie vers la maille.

Le nombre maximum de sources passives que puisse supporter un noeud du noyau est limité par le paramètre *Max-Pass-Size*. La distance en nombre de sauts séparant un nœud du noyau d'une source passive est bornée par le paramètre *Max-Hop*. Ces deux paramètres sont utilisés dans le but de limiter le nombre des nœuds du noyau, et pour garantir la robustesse basique permettant de fournir une redondance moyenne dans les

chemins de routage. Car, si le nombre des nœuds dans le noyau est très petit, la redondance dans la maille ne sera pas suffisante pour garantir des transmissions fiables.

Un drapeau additionnel *Core-Acceptance* est ajouté à la requête *Join-Request*. Il est mis à 1quand la source multicast émettrice est capable de supporter des sources passives ; sinon, il est à 0. Quand une source active reçoit *Join-Request* depuis une autre source, cette première devient une source passive si les conditions suivantes sont vraies:

- Le drapeau *Core-Acceptance* de la requête reçue est mis à 1.
- La distance traversée par cette requête est inférieure ou égale à *Max-Hop*.
- L'ID de la source réceptrice est inférieur à l'ID de l'émettrice de *Join-Request*.

Dans ce cas, la source émettrice est appelée *To-Be-Core-Source* et la source réceptrice est appelée *To-Be-Passive-Source*. La source *To-Be-Passive-Source* envoie le paquet *Pass-Req* à la source *To-Be-Core-Source*, en mettant le champs *Core-Req* à 1 et en ajoutant son propre ID dans le champs *Passive-Source-ID* de ce paquet. Si la source *To-Be-Core-Source* est capable de supporter la source passive, elle lui revoie le paquet *Confirm*. Elle devient ensuite le nœud noyau de cette source *To-Be-Passive-Source* lors d'acheminement des paquets multicast dans le réseau. Si le compteur des sources passives que puisse supporter le nœud du noyau atteint le seuil *Max-Pass-Size*, lors du réception d'une nouvelle requête *Join-Request*, ce nœud du noyau met le drapeau *Core-Acceptance* à 0, jusqu'à ce que le compteur redevient inférieur à *Max-Pass-Size*. Quand la source *To-Be-Passive-Source* reçoit le paquet *Confirm*, elle change son état de source active à source passive. Elle ne diffuse plus de requête *Join-Request* jusqu'à ce qu'elle redevient source active une autre fois.

Les sources passives rafraîchissent leur état par l'envoie du paquet *Pass-Req* vers leur nœuds du noyau associés, après chaque réception de la requête *Join-Request* diffusée par ceux ci. Si un nœud du noyau échoue dans la réception de message *PassReq* depuis l'une de ses sources passives qui les supporte, et après avoir diffusé *Join-Request*, il supprime de sa mémoire l'information qui concerne cette source passive et décrémente le compteur qui compte le nombre de sources passive qu'il supporte. Si ce compteur atteint la valeur 0, ce nœud du noyau devient une source active. D'autre part, si un nœud passive reçoit *Join-Request* depuis son nœud du noyau, qui a traversé une distance plus grande que *Max-Hop*, cette source passive change son état à l'état active et renvoie le message *Pass-Req* dont le drapeau *Core-Req* est mis à 0, vers son nœud du noyau associé. Quand ce nœud du noyau reçoit ce paquet, il supprime de sa mémoire l'information concernant cette source passive et décrémente son compteur qui compte les sources passives qui les supporte.

DCMP réduit la charge de contrôle en réduisant le nombre de sources actives. Cependant, la réduction des sources actives peut affaiblir la robustesse de la maille car la redondance fournit par la maille est proportionnelle au nombre de sources dans le réseau. Si une source passive lie entre son nœud du noyau et les récepteurs, des routes

sous optimales pour l'acheminement des données seront disponibles, produisant un grand délai de distribution de données. De plus, l'instabilité peut subsister à cause des fréquents changements dans les états des sources entre passive et active.

Pour mieux expliquer comment l'instabilité des états peut causer des transmissions inefficaces, je donne l'exemple suivant. Supposant qu'une source passive A est supportée par un nœud du noyau B. Peu de temps après, et à cause de la mobilité, le nœud A devient un nœud du noyau. Par l'envoie de sa requête périodique *Join-Request*, la maille (contenant A) est étendue par l'introduction d'un certain nœud C dans le groupe d'acheminement. Après cela, il est possible que le nœud A devient une source passive supportée par un autre nœud du noyau par exemple, et cela, après avoir effectuer des calculs locaux lors de réception de *Join-Request*. Le nœud A pour l'instant n'envoie plus de requête *Join-Request*. Cependant, le noeud C continue à agir comme étant un membre de groupe d'acheminement avant l'expiration de sa durée, et c'est ce qui génère l'inefficacité du transmission.

3.9 CAMP: Core Assisted Mesh Protocol

CAMP [80] est conçu pour le routage multicast dans les réseaux ad hoc. Il généralise la notion des arbres basés noyau introduits pour le routage multicast Internet par des mailles fournissant plus de connectivité que les arbres. Le noyau est utilisé pour limiter le taux de contrôle dont les récepteurs multicast ont besoin pour rejoindre leur groupe multicast en cas de mobilité, et pour maintenir le groupe multicast aussi. CAMP nécessite l'utilisation d'un protocole unicast spécifique lui permettant de calculer les distances exactes vers toutes les destinations et dans un temps fini.

Dans CAMP, tous les nœuds du réseau maintiennent un ensemble de tables ainsi que les informations de routage et de voisinage. En plus, tous les nœuds membres maintiennent un ensemble de caches contenant les informations du dernier paquet de donnée reçu et les requêtes envoyées par les voisins connus. CAMP classifie les nœuds du réseau en deux catégories : membre duplexe et membre simplexe. Les membres duplexe sont tous les nœuds membres de la maille, alors que les membres simplexes sont utilisés pour créer des connections à sens unique entre les nœuds émetteurs seulement et le reste de la maille multicast. Les nœuds du noyau sont utilisés pour limiter le flux des paquets *Join-Request*.

Un nœud désirant joindre la maille multicast consulte d'abord sa table de routage pour vérifier s'il existe des voisins qui sont déjà membre de cette maille. Si oui, ce nœud annonce sa liaison avec ce voisin à travers le message *Camp-Update*. Sinon, soit qu'il propage la requête *Join-Request* vers l'un des nœuds du noyau de cette maille, ou essaie de joindre un routeur en utilisant le mécanisme « *Expanding Ring Search* » pour diffuser les requêtes. N'importe quel membre duplexe de la maille peut répondre avec le message *Join-Ack*, qui est propagé vers l'initiateur de la requête.

Périodiquement, chaque récepteur multicast révise son cache afin de vérifier si ses paquets de données sont reçus depuis ses voisins qui se trouvent sur le chemin inverse le plus court qui mène à la source. Dans le cas contraire, ce nœud envoie soit le message « *Hearbeat* » ou le message « *Pus-Join* » vers la source et sur le chemin inverse le plus court. Ce processus permet d'assurer que la maille ne contient les chemins les plus courts inverses depuis tous les récepteurs vers tous les émetteurs. Les nœuds choisissent et rafraîchissent périodiquement leurs présentateurs de la maille multicast en diffusant des messages de mise à jour. Ces présentateurs sont des nœuds voisins et sont responsable de rediffuser tous les paquets de données non dupliqués qu'ils reçoivent. Un nœud peut arrêter le fait de présenter un sous ensemble de nœuds si ces derniers ne rafraîchissent pas leurs connections. Il peut ainsi quitter la maille multicast s'il n'est pas intéressé à la session multicast et ne jouera plus le rôle de présentateur d'aucun nœud voisin. La figure 3.8 illustre la différence entre la distribution des paquets de données dans une maille multicast et dans l'arbre de distribution multicast associé.

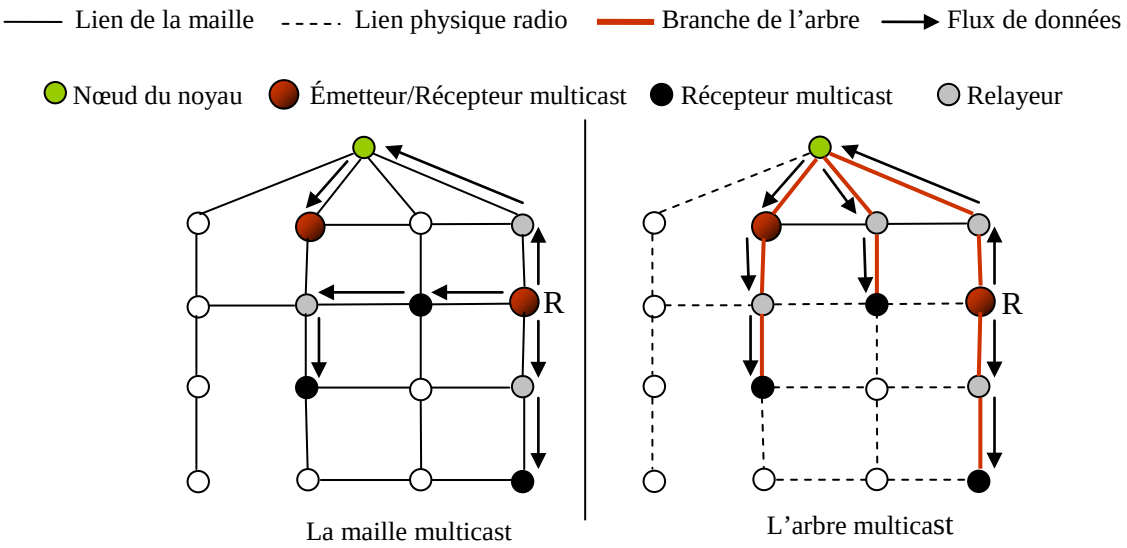


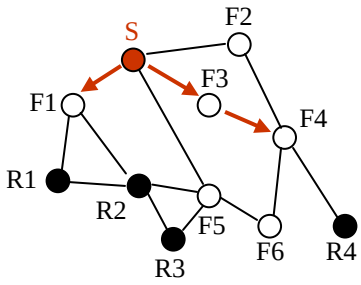
FIG 3.8 – *Circulation de flux d'information depuis le relayeur R*

CAMP peut fonctionner avec certains protocoles unicast, les protocoles unicast basés sur l'algorithme de *Bellman-Ford* ne peuvent pas être utilisés dans CAMP. Dans CAMP, les destinations distantes possèdent peu de chemins redondants que les destinations proches au centre de la maille, et durant la mobilité des nœuds, ces premiers peuvent perdre la réception correcte des paquets de données. Ainsi, plus de retransmissions de paquets sont nécessaires pour les récepteurs loin plus que les récepteurs proches au centre.

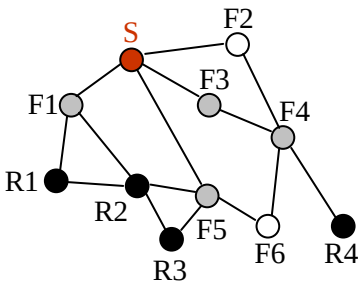
3.10 FGMP: Forwarding Group Multicast Protocol

FGMP [101] est un protocole de routage multicast conçu pour les réseaux Ad hoc. Dans ce protocole, au lieu de former un arbre multicast, un groupe de nœuds chargé d'acheminer les paquets multicast est désigné selon les requêtes envoyées par les membres. Le routage multicast est effectué par la suite à travers une diffusion limitée dans ce groupe. Ce dernier est rafraîchit périodiquement pour mettre à jour la topologie et les liens utilisés. Le routage multicast basé sur un groupe d'acheminement prend la caractéristique diffusante des transmissions sans fil, réduit la charge de stockage et d'occupation des canaux, améliorant ainsi la performance et la scalabilité. L'autre innovation clé par rapport aux protocoles filaires consiste à utiliser la notion des drapeaux au lieu des liens successeurs et prédécesseurs, marquant ainsi une grande robustesse contre la mobilité. La capacité des reconfigurations dynamiques rend le protocole plus convenable avec la mobilité du réseau.

Dans FGMP, la source diffuse juste les données, et uniquement les nœuds d'acheminement sont chargés de la rediffuser. Chaque nœud d'acheminement maintient uniquement deux paramètres, le premier est un drapeau indiquant l'état de ce nœud et prend soit la valeur 1 s'il s'agit d'un nœud d'acheminement, sinon la valeur 0 dans le cas contraire. L'autre paramètre est le timer. Chaque nœud d'acheminement est effectif seulement avant l'expiration de son timer. Ce protocole est similaire au protocole ODMRP, seulement la façon de création du groupe d'acheminement qui diffère. La création et la maintenance de ce groupe d'acheminement peuvent être achevées de deux manières différentes, soit de la manière *Sender-Initiate* ou de la manière *Receiver-Initiate*. Dans le premier cas, c'est l'émetteur qui initie la procédure de création de la maille, dans l'autre cas c'est le récepteur qui se charge de cette tâche. Généralement, ces deux cas sont tout à fait identiques. Le premier cas *Sender-Initiate* est plus efficace lorsque le nombre des sources multicast dans le groupe dépasse le nombre de récepteurs multicast. Dans le deuxième cas *Receiver-Initiate*, chaque récepteur diffuse périodiquement ses liaisons avec ses voisins. Dans ce cas, l'émetteur collecte ces informations pour créer et mettre à jour une table des membres. Après, ce dernier crée aussi une table d'acheminement en se basant sur sa table des membres et quelques autres tables de routage préexistantes. Enfin, cet émetteur diffuse sa table d'acheminement vers tous ses voisins. Uniquement les noeuds voisins se trouvant dans la liste *Next-Hop-List* de la table d'acheminement reçue acceptent cette table et créent par la suite leurs propres tables d'acheminement. Ils rediffusent à leur tours leur tables d'acheminement vers leur voisins, et ainsi de suite jusqu'à ce qu'il atteint le récepteur vers la fin. Par cette manière de transmission de la table d'acheminement, les nœuds d'acheminement seront sélectionnés. La figure 3.9 illustre un exemple de création de groupe d'acheminement dans FGMP.



a) Diffusion des tables d'acheminement



b) Création de groupe d'acheminement

Récepteur	Saut suivant
R1	F1
R2	F1
R3	F5
R4	F3

c) Table d'acheminement du nœud S

Récepteur	Saut suivant
R4	F4

d) Table d'acheminement du nœud F3

FIG 3.9 – *Création de groupe d'acheminement dans FGMP*

Des résultats de simulations préliminaires montrent que le protocole proposé est plus robuste contre la mobilité par rapport à la version qui se base sur une structure de routage globale (Vecteurs de distances). Il possède une performance beaucoup mieux que les systèmes multicast basés arbre comme DVMRP et les arbres partagés. La raison de cette performance supérieure est la charge de contrôle minimale et la détection facile des liens endommagés. La scalabilité de stockage est améliorée aussi d'une façon remarquable à travers le routage à la demande, et surtout dans les grands réseaux possédant des liens dispersés.

3.11 MAODV: Multicast AODV

MAODV (Multicast Ad hoc On-Demand Distance Vector) [102] est un protocole multicast conçu pour le protocole de routage unicast AODV. Il utilise les mêmes principes et formats de requêtes de recherche de route vers les destinations. MAODV établit un arbre multicast partagé centré sur un noyau pour chaque groupe multicast du réseau. Le noyau est appelé *Leader* du groupe puisque c'est le premier participant au groupe qui le gère et le maintient en place. Pour cela, il diffuse périodiquement dans le réseau un message « *Group Hello* » contenant le numéro de séquence du groupe, le groupe multicast concerné et l'identificateur du *Leader* du groupe.

Lorsqu'un noeud N désire joindre le groupe multicast, il diffuse la requête de route *RREQ* munie de l'adresse du groupe comme destination et de drapeau validé « J » indiquant son désir de jointure. Si N est le premier à joindre ce groupe, il est certain qu'il ne va recevoir aucune réponse. Après avoir passé une certaine durée, le nœud crée un nouveau groupe et se déclare comme son *Leader* en diffusant périodiquement le message « *Group Hello* ». Le numéro de séquence de ce message est initialisé et incrémenté périodiquement par le *Leader* du groupe. Les nœuds membre de ce groupe enregistrent ce numéro de séquence pour l'utiliser quand ils répondent sur *RREQ*. Le message « *Group Hello* » inclut aussi l'information de nombre de sauts. Ultérieurement, les nœuds de l'arbre seront capables de connaître la distance qui les sépare de *Leader* lorsqu'ils reçoivent le message « *Group Hello* ».

Comme le montre la figure 3.10, si un nœud R décide de joindre le groupe, qui existe déjà, il diffuse la requête *RREQ*. Ce nœud peut recevoir plusieurs réponses *RREP* depuis les nœuds de l'arbre multicast. (D'autres nœuds rediffusent la requête *RREQ* s'ils ne connaissent aucune route qui les mène vers le *Leader* ou s'ils ne sont pas encore membres de l'arbre multicast, jusqu'à ce qu'ils atteignent un nœud de l'arbre multicast, et ce dernier nœud d'arbre leur renvoie la réponse *RREP*). Chaque réponse reçue depuis un membre du groupe multicast implique l'existence d'une route depuis ce groupe vers le nœud R. A cause de la topologie d'arbre utilisée dans MAODV, le noeud R doit explicitement choisir l'une des routes proposées et l'activer par la suite en envoyant en unicast le message d'activation de route « *MACT* ». Ce message cause chaque nœud se trouvant sur le chemin suivi pour qu'il s'ajoute à l'arbre multicast et former en fin une branche multicast se terminant à R. Quoique les nœuds d'un chemin puissent ne pas être des membres multicast, ils doivent résider sur l'arbre multicast en tant que routeur.

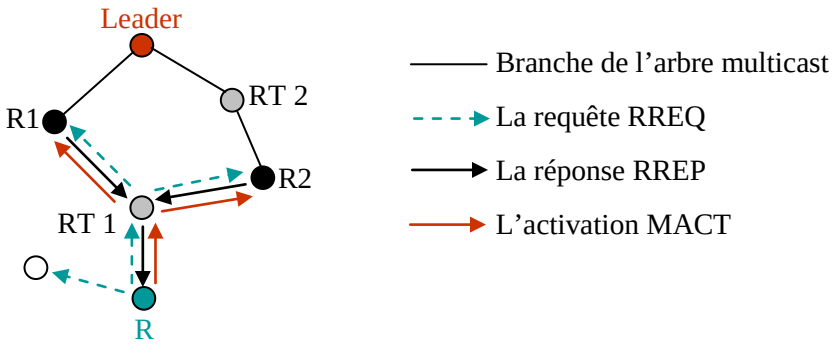


FIG 3.10 – *Procédures de jointure dans MAODV*

Un nœud du groupe multicast peut quitter le groupe à n'importe quel moment. Cependant, si ce noeud n'est pas une feuille de l'arbre multicast alors, il doit rester dans l'arbre pour continuer à agir comme un routeur de distribution. Si par exemple le récepteur multicast R1 de la figure 3.10 désire quitter le groupe, celui-ci doit continuer à agir comme un routeur multicast. Si le récepteur multicast désirant quitter le groupe est une feuille de l'arbre (soit R par exemple) alors, il envoie un message spécial *MACT* vers le groupe. Après avoir reçu ce message, les nœuds se trouvant sur le chemin inverse qui mène au *Leader* et qui ne sont pas des récepteurs multicast (comme le routeur RT1), se détachent aussi de l'arbre multicast, et acheminent de nouveau le message spécial *MACT* vers leur parent jusqu'à ce que ce message atteint un récepteur multicast (comme R1 par exemple).

A cause de la mobilité, les branches de l'arbre peuvent se détruire facilement. Dans ce cas, le premier nœud successeur depuis le pont de cassure devient responsable d'initier la procédure de reconstruction de branche. Il envoie pour cela le message *RREQ* muni de drapeau de jointure validé « J ». Uniquement le nœud de l'arbre multicast possédant un nombre de sauts depuis le *Leader* minimal ou égale à celui du nœud initiateur, et possédant aussi des informations de routage vers le *Leader* les plus récentes, est permis de répondre sur la requête *RREQ*.

Ce protocole sauvegarde beaucoup d'états dans ses tables de routage, ces dernières sont mises à jour périodiquement. Quand un lien tombe en panne, il sera rapidement détecté et certaines procédures de réparation seront utilisées pour le rendre en marche. Une charge de contrôle excessive sera demandée pour maintenir cet arbre, ce qui consomme trop la bande passante dont le reste ne servira pas à une bonne distribution de données. La réparation des branches cassées peut faire inclure de nouveaux nœuds dans l'arbre, augmenter la longueur des chemins, et ainsi, les branches deviennent plus vulnérables aux destructions en cas de fortes mobilités. Par conséquent, MAODV souffre de taux bas de distribution des données.

3.12 DVMRP: Distance Vector Multicast Routing Protocol

DVMRP [66] est un protocole de routage multicast conçu initialement pour les réseaux filaires. Des extensions effectuées sur ce protocole sont présentées dans [84], ces dernières permettent au protocole de fonctionner efficacement dans un environnement mobile ad hoc. Je peux résumé les grandes lignes de ces extensions sous formes des trois points suivants :

- Détection des nœuds feuilles ;
- Elagage et greffage dynamique ;
- Utilisation des mécanismes de vérification des paquets dupliqués ;

DVMRP est un protocole multicast basé source. Pour construire cet arbre, la source diffuse périodiquement le message « *Advertise* » en utilisant l'algorithme d'acheminement sur le chemin inverse (*RPF*), cet algorithme permet de contrôler le processus de diffusion en donnant la main de rediffusion uniquement au nœuds se trouvant sur le chemin le plus court depuis la source.

Quand un nœud feuille, non intéressé à la session multicast, reçoit le message diffusé « *Advertise* », il envoie donc un autre message « *Prune* » vers son prédécesseur. Les nœuds intermédiaires recevant ce message vont marquer le lien correspondant comme étant un lien élagué « *Pruned-off* », et n'acheminent plus de paquets multicast sur ce lien durant toute la session multicast. Si un noeud intermédiaire n'est pas intéressé à cette même session multicast, et si tous ses liens sortant sont élagués alors, lui aussi renvoie vers son prédécesseur le message « *Prune* » et ainsi de suite. Une fois que ce processus d'élagage soit terminé, un arbre multicast basé source sera établi. Le processus de diffusion et d'élagage est illustré dans la figure 3.11.

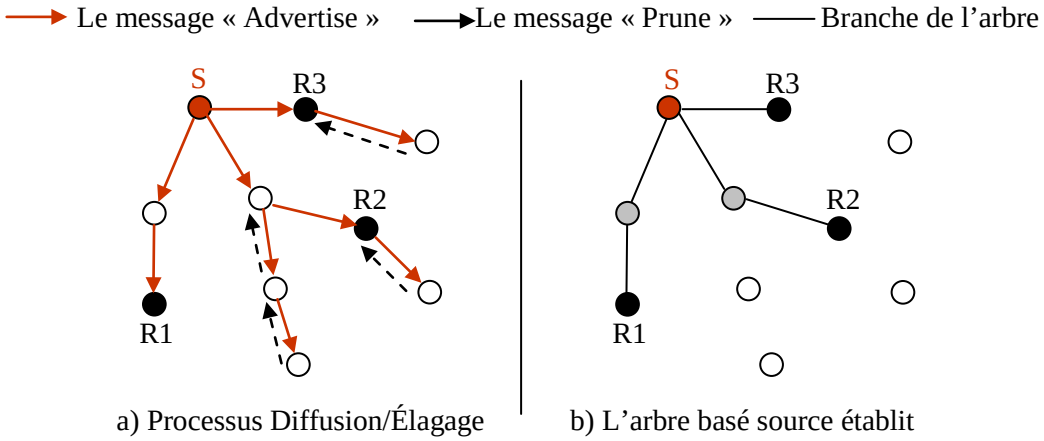


FIG 3.11 – *Méthode de Diffusion/Elagage dans DVMRP*

L'opération de jointure/départ est contrôlée par le fait qu'un message d'élagage « *Prune* » soit propagé en arrière. Périodiquement, le timer associé à chaque branche élaguée expire et les paquets vont être diffusés e nouveau. Si la feuille de la branche ne s'intéresse pas encore à la session multicast, le processus d'élagage se répétera et la branche est élagué une autre fois, sinon, aucun message d'élagage n'est envoyé et la branche devienne membre de l'arbre.

Ce mécanisme encourt le temps de latence lorsque de nouveaux membres désirent joindre le groupe multicast car chacun d'eux doit attendre la diffusion du prochain message « *Advertise* » lancé par la source périodiquement. Pour éliminer ce temps de latence, un nouveau membre peut explicitement envoyer des messages « *Graft* » vers leur successeur en récupérant ainsi l'ancienne branche qui a été purgée précédemment.

En appliquant ce protocole dans les réseaux ad hoc, un problème se présente. Selon DVMRP, le message de purgation est initié par un nœud feuille ne s'intéressant pas à la session multicast, et est propagé vers le nœud prédécesseur. Cependant, dans un réseau mobile ad hoc, il est vraiment difficile à un nœud de connaître s'il est un noeud feuille ou non. Deux systèmes possibles peuvent résoudre ce problème. Le premier consiste à utiliser des messages de reconnaissance et l'autre consiste à échanger les tables de routage entre les voisins.

Le protocole original DVMRP rajoute un mécanisme basé sur le vecteur de distance (*Distance-Vector*) afin de fournir l'information du chemin le plus court menant à la source. Cette information sera utilisée par la suite par l'algorithme de chemin inverse *RPF*. Cependant, dans un réseau très dynamique, les routes menant à la source peuvent changer très rapidement que le protocole de routage en garde trace. Dans ce cas, l'algorithme *RPF* échoue et ne fonctionne pas efficacement. Dans [84], l'algorithme *RPF* est remplacé par un mécanisme de vérification des messages dupliqués pour s'assurer que chaque nœud diffuse uniquement les paquets non dupliqués.

L'opération de Greffage/Purgation dynamique représente une autre extension pour utiliser ce protocole dans les réseaux ad hoc, elle permet d'effectuer des reconfigurations de l'arbre très rapidement. Dans le DVMRP originale, uniquement l'opération de greffage dynamique est fournie afin de permettre aux nouveaux membres de joindre l'arbre multicast. Dans l'opération Greffage/Purgation dynamique, quand un nœud détecte que le chemin le plus court menant à la source a changé (par exemple, quand le trafic multicast est reçu depuis plusieurs nœuds prédécesseurs pendant une période plus grande que la période du seuil), il envoie un message « *Prune* » vers le nœud prédécesseur courant et le message « *Graft* » vers le nouveau prédécesseur. De cette manière, l'arbre multicast DVMRP s'adapte plus rapidement à la mobilité du réseau.

3.13 MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing

MCEDAR [52] est l'extension de l'architecture CEDAR [103], il fournit à la fois la robustesse des protocoles de routage basés maille et l'efficacité des protocoles d'acheminement basés arbre.

Le protocole unicast CEDAR établit dynamiquement un noyau constitué d'un sous ensemble de nœuds du réseau. D'une manière incrémentale, CEDAR propage vers les nœuds du noyau les états des liens caractérisés par une bande passante de grande stabilité. Le noyau de CEDAR représente l'ensemble dominant minimal du réseau, le choix de cet ensemble se repose uniquement sur les informations locales des noeuds. Chaque nœud dans le réseau diffuse périodiquement une balise contenant: le degré, le degré effectif (représente le nombre des noeuds voisins qui l'on choisit comme dominateur) et l'ID de son dominateur, ces informations servent à calculer le noyau. Pour choisir son dominateur, chaque nœud choisit parmi les nœuds de son premier voisinage celui qui possède le plus grand degré et le plus grand degré effectif. Quand un nœud perd sa liaison avec son dominateur, il attend une certaine période pour qu'il puisse recevoir les balises périodiques de ses voisins. Ensuite, ce nœud peut décider soit de se rattacher de nouveau à son dominateur, soit de joindre un autre nœud du noyau dominé par l'un de ses voisins ou de devenir lui même le dominateur. Les informations de topologie ou d'états sont échangées entre les nœuds du noyau à travers un mécanisme « *Core Broadcast Mecanism* », qui coopère entre la couche MAC et la couche Réseau.

L'extension multicast MCEDAR utilise deux composants de CEDAR: l'architecture du noyau et la diffusion dans le noyau. MCEDAR adopte le mécanisme de sélection du noyau de CEDAR, le graphe constitué de ces nœuds s'appelle *Core-Graph*. MCEDAR extrait un sous graphe depuis le *Core-Graph* afin de l'utiliser comme infrastructure pour le routage multicast. Ce sous graphe prend une topologie en maille et s'appelle *M-Graph*, il utilisé dans le but de fournir des liens redondants. Uniquement les nœuds de *M-Graph* ont le droit de distribuer les paquets de données dans tout le groupe multicast.

Quand un nœud n'appartenant pas au noyau désire joindre le groupe multicast, il demande à son dominateur (qui est membre du noyau) d'effectuer cette opération de jointure. Ce dominateur initie l'opération de jointure en diffusant la requête *Join<Multicast-Group-Adresse, Join-ID>* vers tous les nœuds du noyau. Le premier champ de cette requête représente l'adresse du groupe dont le nœud désire joindre, le deuxième champ représente la durée de jointure d'un nœud au groupe multicast associé. Ce champ prend la valeur infinie quand il s'agit d'une nouvelle jointure. Quand un nœud non membre du groupe multicast reçoit le message *Join*, il le ré-achemine vers les nœuds du noyau les plus proches en utilisant le mécanisme de diffusion du le noyau proposé dans CEDAR. Quand ce message atteint un nœud du noyau, il renvoie *Join-*

Ack <Multicast Group Adress, Join-ID> seulement si son *Join-ID* est inférieur à celui de la requête. Ensuite, il continue à acheminer la requête Join vers ses voisins.

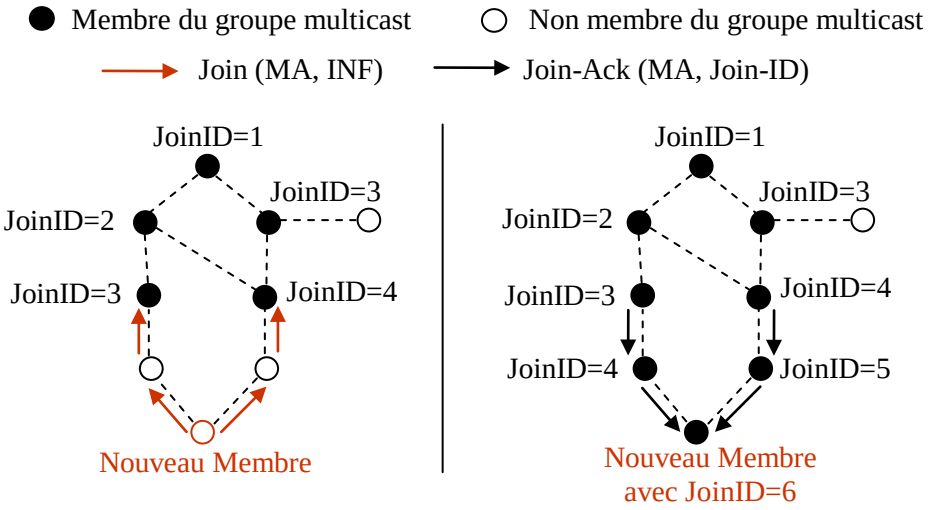


FIG 3.12 – *Processus de jointure dans MCEDAR*

Le champ *JoinID* de *Join-Ack* renvoyé par le membre du noyau prend la valeur de *JoinID* de la requête reçu par ce membre. Lors de réception de cette réponse, un nœud intermédiaire non membre du noyau décide soit d'accepter cette réponse ou non en se basant sur un certain paramètre de robustesse R. Si ce récepteur a déjà accepté cette réponse pour un nombre de fois inférieure à R alors, il peut accepter encore cette réponse. Dans ce cas, il met son champ *JoinID* à la valeur maximale entre son *JoinID* actuel et le *JoinID* reçu depuis la requête, incrémenté par un. Il estompé ensuite le *JoinID* de la réponse *Join-Ack* avec cette nouvelle valeur calculée et envoie cette réponse vers ses successeurs. La figure 3.12 donne un exemple de jointure d'un nouveau nœud avec un facteur de robustesse R=2.

Après avoir extrait le *M-Graph* depuis le noyau, l'acheminement de données peut s'effectuer sur ce *M-Graph* en utilisant le mécanisme de diffusion de noyau proposé dans CEDAR. Les nœuds de *M-Graph* sont chargés d'acheminer les paquets multicast qu'ils reçoivent. Quand un membre de *M-Graph* reçoit un paquet de données, il tente de l'acheminer vers ses voisins qui sont eux aussi membre de même *M-Graph* que lui. Ce mécanisme d'acheminement élimine les transmissions redondantes économisant ainsi la bande passante et améliorant la fiabilité du protocole lorsque les nœuds deviennent très mobiles.

L'inconvénient de ce protocole est le fait que chaque nœud doit diffuser une balise périodique lui permettant de rafraîchir ses informations de voisinage ce qui mène à des collisions. Ceci peut causer en plus des échecs dans la sélection des noeuds du noyau.

3.14 MZR: Multicast protocol based on Zone Routing

MZR [104] est un protocole multicast hybride initié source, et dans lequel un arbre de distribution multicast basé source est crée en utilisant le mécanisme de routage par zone [93]. Ce protocole ne dépend d'aucun protocole de routage unicast dans toutes ses opérations et il restreint l'opération de maintenance de la topologie, en cas de changement, à un sous ensemble de nœuds au lieu de la totalité des nœuds du réseau. Dans ce protocole, chaque nœud du réseau construit une zone autour de lui, et avec un rayon limité, mesuré en nombre de sauts. Pour s'identifier dans le réseau, chaque nœud diffuse périodiquement un paquet « *Advertisement* », le champ *TTL* de ce paquet prend la valeur du rayon de la zone correspondante. Quand un nœud reçoit le paquet « *Advertisement* » d'un autre nœud, ce premier crée une nouvelle entrée dans sa table de routage pour marquer une nouvelle route depuis l'émetteur. L'entrée d'une route dans la table de routage s'expire si le nœud associé n'envoie plus de paquets « *Advertisement* » dans les périodes prévues. En utilisant cette méthode, chaque nœud peut connaître les nœuds se trouvant dans sa zone.

Une source multicast initie la création de l'arbre multicast. En premier lieu, la source forme un arbre multicast dans sa zone à travers l'envoie du message *Tree-Create* vers chaque nœud se trouvant dans sa zone et sur des routes unicast obtenues depuis sa table de routage. Lors de propagation de ce message dans la zone, des entrées des routes inverses sont crées au niveau de chaque nœud traversé. Quand un récepteur multicast se trouvant à l'intérieur de cette zone reçoit *Tree-Create*, il répond par le renvoie du message *Join-Create-Ack*. Les nœuds traversés par cette réponse depuis le récepteur multicast jusqu'à la source deviennent des membres de l'arbre multicast de cette zone. La figure 3.13 illustre un exemple de création d'un arbre multicast initié par la source multicast S et à l'intérieur de sa zone. Les récepteurs appartenant à la zone de S étant R1, R2, et R3. Le nœud non membre multicast I devient membre de cet arbre car il connecte la source S au récepteur multicast R3.

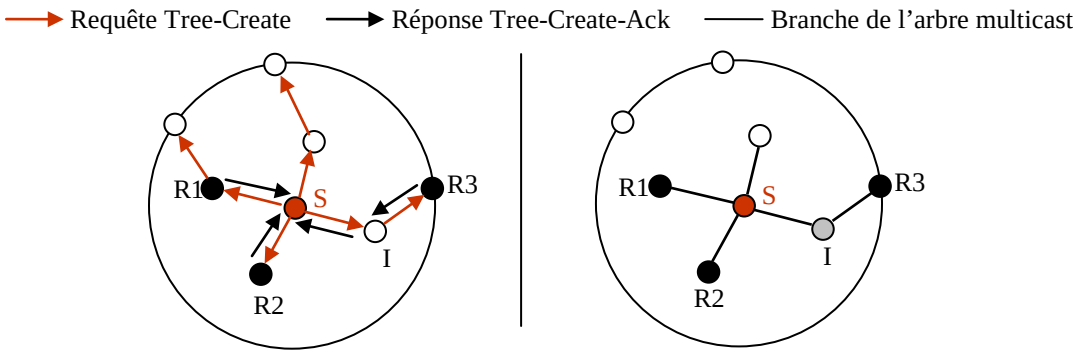


FIG 3.13 –*Création de l'arbre multicast à l'intérieure d'une zone*

Par la suite, la source S essaie de diffuser cet arbre à la totalité du réseau. Grâce à la table de routage par zone, la source peut facilement identifier tous les nœuds bordure de sa zone (ce sont les noeuds distants d'un nombre de sauts égale au rayon de la zone). La source envoie en unicast le paquet *Tree-Propagate* vers chacun de ces nœuds bordures. Quand un de ces derniers reçoit ce paquet (qu'il soit récepteur multicast ou pas), il doit renvoyer le message *Tree-Create* vers tous les nœuds appartenant à sa propre zone. Dans le cas où le récepteur de *Tree-Create* est un récepteur multicast, ce dernier renvoie la répond *Tree-Create-Ack*. Le nœud de bordure initiateur envoie en unicast, à son tour, le même message *Tree-Create-Ack* vers la source multicast. Il refait ensuite la même procédure que la source multicast, en envoyant cette fois ci le message *Tree-Propagate* vers les nœuds bordure appartenant à sa propre zone. Ce processus permet de propager la création de l'arbre en continuant à refaire les mêmes opérations jusqu'à ce que les paquets *Tree-Create* arrivent à tous les nœuds du réseau. De cette façon, l'arbre multicast est établi et la transmission des paquets de données peut commencer.

Dans MZR, les nœuds successeurs sont chargés de détecter et reconstruire les branches cassées. Le nœud successeur initie une recherche globale pour l'arbre multicast en utilisant le mécanisme de routage par zone. Il commence par l'envoie du message *Join* vers tous les nœuds appartenant à sa zone avec une valeur *TTL* égale au rayon de cette zone. Si les nœuds de sa zone recevant ce paquet appartiennent déjà à l'arbre multicast, ces derniers répondent par *Join-Ack* et de nouvelles branches seront établies. Sinon, si aucune réponse de ce genre n'est reçue, le nœud successeur propage la procédure de jointure en envoyant le message *Join-Propagate* vers tous les nœuds bordure de sa propre zone, qui à leur tours envoient leur propres message *Join* vers les nœuds de leur propre zone. Si ces derniers reçoivent une réponse, le message *Join-Ack* est renvoyé vers le nœud initiateur. Sinon, le message *Join-Propagate* est retransmis encore et ainsi de suite jusqu'à la réception d'une réponse.

MZR se caractérise par le fait de propager ses requêtes via les noeuds bordure des zones, ce qui permet de faciliter et accélérer efficacement l'établissement de l'arbre multicast. Ceci est basé principalement sur l'exactitude de l'information de routage par zone que collecte chaque nœud du réseau lors de réception des messages de contrôle « *Advertisement* ». Néanmoins, si les nœuds se déplacent très rapidement, l'information concernant une zone peut changer et devenir incorrecte très rapidement aussi, résultant ainsi en une propagation de messages de moindre efficacité. D'autre part, dans le but de garder l'information de routage par zone assez fraîche, la fréquence d'envoie du message « *Advertisement* » doit être grande, ce qui génère un taux de contrôle très élevé. Aussi, avec l'augmentation du nombre de nœuds dans le réseau, et malgré le fait que la diffusion du message « *Advertisement* » soit limitée par le nombre de sauts, la charge de contrôle reste toujours importante. L'arbre multicast établie dans MZR souffre aussi des cassures répétitives et nombreuses de ses branches comme dans tous les protocoles basés arbre, et nécessite donc de nombreux efforts pour réparer toutes les branches cassées.

3.15 MOLSR: Multicast Optimized Link State Routing

MOLSR [105] est une extension du protocole unicast OLSR [37], il est chargé de la construction de la structure multicast afin de router les trafics point à multipoints dans un réseau ad hoc. MOLSR est conçu pour des routeurs mobiles et supporte aussi un environnement hétérogène composé de simple routeurs point à point OLSR, des routeurs MOLSR et des machines sans protocole de routage. MOLSR est conçu pour de façon à tirer profit de ce qui est fait dans le protocole OLSR. Il utilise les différentes tables, de voisinage, de topologie et de routage, afin de minimiser le trafic de contrôle et économiser la bande passante.

Le principe de MOLSR consiste à construire et maintenir un arbre multicast basé source pour chaque tuple (Source, Groupe) d'une façon distribuée sans aucune entité centrale tout en offrant les plus courts chemins de la sources aux membres du groupe. Ces arbres sont constitués uniquement des nœuds pouvant gérer le routage multicast (sauf éventuellement les feuilles). Le protocole unicast OLSR réagit à chaque modification dans le voisinage ou la topologie en recalculant l'ensemble des relais multipoint et la table de routage. MOLSR profite de ces mises à jour pour optimiser les arbres multicast et détruire les branches obsolètes. Dans MOLSR, on distingue trois parties : la construction, la maintenance et la destruction de l'arbre multicast.

❖ ***Construction de l'arbre multicast*** Les routeurs multicast se déclarent dans le réseau en utilisant le message *Mc-Claim* (cette diffusion s'effectue en utilisant la diffusion optimisée d'OLSR par l'intermédiaire des relais multipoints). Cette information est diffusée périodiquement dans le réseau.

Quand une source souhaite émettre des données multicast vers un groupe spécifique G, elle diffuse un message *Source-Claim* afin que les membres détectent sa présence et se rattachent à l'arbre associé à cette source. Cette invitation est diffusée dans le réseau en utilisant les relais multipoints. Les branches de l'arbre sont construites en partant des feuilles.

Plus spécifiquement, lorsqu'un membre reçoit *Source-Claim*, et qu'il ne fait pas partie de cet arbre défini par le tuple (Source, Groupe), alors, il se rattache automatiquement à ce dernier comme suit :

- Le membre regarde dans sa table de routage spécifique au calculs multicast (cette table offre les routes optimales composées de nœuds gérant le multicast), pour choisir le prochain saut qui lui permet d'atteindre la source en un nombre minimale de sauts. Ce noeud devient le parent dans cet arbre multicast.
- Ensuite, il envoie le message *Confirm-Parent* à son parent.
- Le nœud parent qui reçoit un tel message se rattache à son tour à l'arbre en question, s'il n'appartient pas déjà à ce dernier.

La construction et l'activation de la branche se fait d'une façon récursive saut par saut à la réception du message *Confirm-Parent* jusqu'à atteindre la source ou un participant. La figure 3.14 illustre le mécanisme de construction de l'arbre multipoint dans MOLSR.

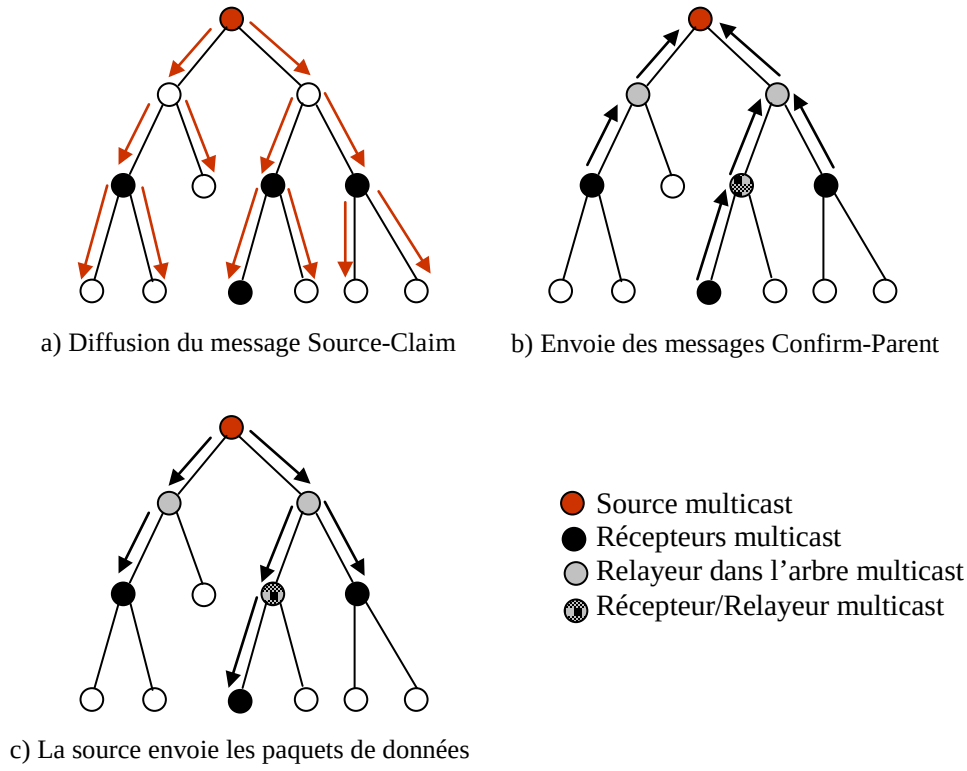


FIG 3.14 – *Mécanisme de construction de l'arbre multipoint*

❖ ***Maintenance de l'arbre multicast*** Les arbres sont rafraîchis périodiquement par l'intermédiaire des messages *Confirm-Parent* et *Source-Claim*. Les modifications de topologie sont détectées par l'échange de messages de contrôle d'OLSR, ce qui entraîne automatiquement les mises à jour nécessaires à répercuter sur les structures multicast gérées par le noeud considéré.

❖ ***Maintenance de l'arbre multicast*** Si un nœud veut quitter l'arbre multicast et qu'il est une feuille, alors, il se détache de l'arbre en envoyant à son parent un message *Leave*. Si le parent devient une feuille à son tour, et qu'il n'est pas membre du groupe multicast, alors, il effectue la même opération de destruction de la branche. Ce message est traité nœud par nœud à chaque saut permettant l'élimination des branches inutilisées dans le réseau et optimisant ainsi la bande passante.

	AMRoute	AMRIS	ODMRP	CAMP	FGMP	MAODV	DVMRP	MEDAR	MZR	MOLSR
Structure utilisée	Maille/ Arbre partagé	Arbre partagé	Maille	Maille	G.A	Arbre basé noyau	Arbre basé source	Ensemble dominant	Arbre basé source	Arbre basé source
Utilisation d'un noyau	(logical Cores)	Sid	Non	Core nodes	G.A	Leader	Non	Non	Non	Non
Utilisation des tunnels	Oui	Non	Non	Non	Non	Non	Non	Non	Non	Non
Système de routage	Proactif	Réactif	Réactif	Proactif	Réactif	Réactif	Proactif	Proactif	Hybrid	Réactif
Dépendance de protocole unicast	Oui	Non	Non	Oui	Non	Non	Non	Oui	Non	Oui
Métrique de routage	N° séquence	Msm-id	Shortest path	Shortest path	Scoped flood	Shortest path	Shortest path	Core broadcast	Zone	Shortest path
Messages périodiques	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Génération des boucles	Oui	Non	Oui	Oui	Non	Non	Non	Non / collision	Non	Non

***TAB 3.1** – Comparaison de quelques protocoles multicast dans les réseaux ad hoc*

Ce tableau m'a permis de constater que la plus part des protocoles utilisent des arbres multicast, quelque soit leur famille. L'utilisation des tunnels pour la transmission des paquets de données n'est pas une approche généralisée. La plus part de ces protocoles sont conçus spécialement pour les communications multicast dans les réseaux ad hoc ; quelques uns sont des extensions multicast à partir de leur version unicast. Pratiquement, la majorité des protocoles se basent sur le principe de plus court chemin, et utilisent des messages de contrôle périodiques pour la maintenance des structures de routage. Je remarque aussi que tous les protocoles essaient d'empêcher le problème de la formation des boucles.

3.16 Conclusion

Dans ce chapitre, j'ai présenté quelques protocoles multicast améliorés ou conçus spécialement pour les environnements mobiles ad hoc. Quinze protocoles en tout, qui différent d'une ou de plusieurs propriétés, ont été décrits en détail et avec des schémas. Ces différences résident dans leurs structures, leurs utilisations ou pas des tunnels, leurs concentrations ou pas sur un noyau, leurs dépendances d'un protocole unicast ou non ...etc. Toutes les propriétés de comparaison ont été mises dans un tableau de comparaison (tableau 3.1) qui regroupe dix (10) protocoles multicast choisis parmi les quinze présentés dans ce chapitre. Il s'agit des protocoles : AMRoute, AMRIS, ODMRP, CAMP, FGMP, MAODV, DVMRP, MCEDAR, MZR, MOLSR.

Selon les différentes propriétés présentées dans ce tableau de comparaison, j'ai choisi un protocole de routage multicast hybride qui s'appelle MZRP (Multicast Zone Routing Protocol) et qui sera décrit en détail dans le prochain chapitre. MZRP est un protocole dont le principe de partition du réseaux en zone est le même que celui de protocole MZR présenté ici (voir la section 3.14), sauf que ce premier utilise en plus certains mécanismes de routage très intéressants pour l'optimisation de la recherche des routes. Mon choix se base essentiellement sur la manière avec laquelle ce protocole essaie de réduire la consommation de la bande passante avec ses différents mécanismes (le protocole BRP), et la minimisation de nombre de diffusions périodiques restreint par les rayons des différentes zones.

Une autre technique de diffusion optimisée existe dans le monde sans fil ; et consiste à minimiser le nombre de transmissions inutiles lors d'une diffusion généralisée (inondation) d'un message. Il s'agit de la technique des relais multipoint MPR (MultiPoint Relay), dont j'ai choisi pour l'amélioration de protocole multicast MZRP. Cette technique sera décrite aussi dans les prochaines sections du chapitre suivant.

Donc, MZRP-MPR est le nom du nouveau protocole que j'ai proposé dans ce mémoire. Le chapitre 4 est consacré spécialement pour présenter ce nouveau protocole de routage multicast. Pour cela, je commencerai par décrire la version unicast ZRP, puis la version multicast du protocole MZRP. Je passerai ensuite à la présentation de la

technique MPR, et j'expliquerai, vers la fin, le fonctionnement du protocole proposé MZRP-MPR.

Chapitre 4

Multicast Zone Routing Protocol (MZRP) et MultiPoint Relay (MPR)

4.1 Introduction

Les réseaux ad hoc sont des réseaux sans fil qui ne demandent aucune infrastructure de communication. Ils sont caractérisés par une topologie dynamique, une bande passante limitée et une faible puissance d'énergie. Dans un environnement ad hoc typique, la communication orientée-groupe est plus populaire que la communication un-à-un. Par conséquent, le multicast a été naturellement considérée comme technique idéale à utiliser pour la communication des groupes. Les protocoles multicast conçus pour les réseaux statiques ne conviennent pas dans les réseaux ad hoc, et ce à cause de la mobilité des noeuds et la limitation de la bande passante. Pour cela, plusieurs nouveaux protocoles multicast ont été exclusivement proposés pour les réseaux ad hoc. Certains d'entre eux sont des protocoles multicast basés sur les arbres, comme le protocole RMB [106], le protocole LAM [107], le protocole AMROUTe [108], le protocole AMRIS [78] et le protocole MAODV [109]. Cependant, d'autres protocoles sont basés sur les mailles, comme le protocole CAMP [110] et le protocole ODMRP [111] [112]. Un système de routage basé sur les mailles peut supporter une route plus robuste à travers l'existence des routes redondantes, cependant, les ressources sont épuisées à cause de l'acheminement inutile des données dupliquées. Dans un système de routage multicast basé sur les arbres, l'utilisation des ressources est optimisée, cependant, la mobilité des nœuds induit une charge et une latence de reconstruction très importantes lors des cassures des branches des arbres.

Ce chapitre décrit en premier lieu le protocole ***ZRP*** (***Z**one **R**outing **P**rotocol*) [113] [93] qui a été conçu pour les communications unicast dans les environnements mobiles ad hoc, ainsi que son extension multicast ***MZRP*** (***M**ulticast **Z**one **R**outing **P**rotocol*) [114]. Ce dernier sera décrit en détail dans les prochaines sections de ce chapitre, et il fera objet de ce mémoire car je vais essayer d'améliorer plus ses performances avec l'intégration d'une certaine technique existante dans l'environnement mobile sans fil ; il s'agit de la technique des relais multipoints (MPR). Cette intégration génère le protocole (MZR-MPR) qui sera présenté en détail dans ce même chapitre.

4.2 Le protocole unicast ZRP

Dans les réseaux ad hoc, on peut supposer sans doute que la plus part des communications se font entre les nœuds proches les uns des autres. Les changements topologiques ont une grande importance à la proximité d'un nœud, par ailleurs, l'ajout ou la suppression d'un nœud dans une autre partie du réseau a un impact limité sur les voisinages locaux.

Le protocole ZRP est un protocole de routage unicast utilisant une approche hybride (proactive/réactive) conçu spécialement pour les réseaux sans fil ad hoc. Ce protocole se caractérise par la limitation de la portée de la procédure proactive en un voisinage local appelé ***zone***. Un nœud du réseau peut être membre de plusieurs zones chevauchées, chaque zone peut avoir une taille différente des autres. La taille d'une zone n'est pas une mesure géographique, mais elle est donnée par un rayon de longueur « **ρ** », où « **ρ** » représente le nombre de sauts depuis le nœud central de cette zone jusqu'au périmètre de celle-ci (voir la figure 4.1).

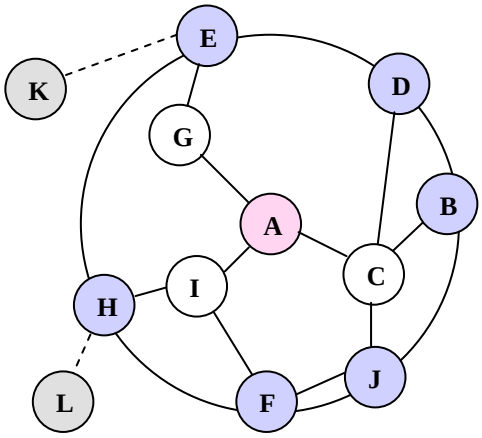


FIG 4.1 *– La zone de routage du nœud A avec* **ρ** *= 2*

En partageant le réseau en plusieurs zones chevauchées et de différentes tailles, le protocole ZRP évite d'avoir une carte hiérarchique totale du réseau, ainsi que le trafic nécessaire pour maintenir une telle carte. Ici, le réseau peut être vu comme un immeuble dans lequel l'optimisation des routes est possible lorsqu'on détecte les zones chevauchées.

Evidement, un nœud a besoin de connaître au départ ses voisins directs avant de construire sa propre zone et déterminer ses nœuds périphériques. Dans le but de connaître ses voisins directs, un nœud peut utiliser directement les protocoles de contrôle d'accès au média (***M**edia **A**ccess **C**ontrol* ou ***MAC***). Alternativement, il peut avoir besoin d'un protocole de découverte de voisins (***N**eighbor **D**iscovery **P**rotocol* ou ***NDP***). En plus, on voie que le protocole ZRP, comme structure, ne spécifie pas strictement le protocole à utiliser mais permet des implémentations indépendantes.

Un protocole NDP compte typiquement sur la transmission des balises « ***Hello*** » par chaque nœud dans le réseau. Si un nœud reçoit une réponse sur un tel message, il peut noter qu'il possède une connexion point-à-point avec ce voisin. Le protocole NDP utilisé est libre de choisir les nœuds selon différents critères, tel que la longueur du signal, de délais et la fréquence des balises, …etc. Une fois que l'information de routage soit collectée, le nœud diffuse périodiquement les messages de découvertes afin de mettre à jour sa carte de voisins [93]. Si la couche MAC des nœuds ne permet pas l'utilisation d'un tel protocole NDP, le protocole de routage ***intrazone*** doit fournir aux nœuds la possibilité de découvrir leurs voisins directs. Ce protocole est responsable de déterminer les routes vers les nœuds périphériques et est généralement un protocole proactif. Le protocole ***Intrazone*** ou ***IARP*** (**I**ntr**A**zone **R**outing **P**rotocol) [115] est décrit en détails dans la section suivante. La communication entre les différentes zones est garantit par le protocole de routage ***Interzone*** ou ***IERP*** (***I**nt**E**rzone **R**outing **P**rotocole*) [116] qui offre les possibilités de routage entre les nœuds périphériques uniquement. Ce qui veut dire que, si un nœud rencontre un paquet destiné à l'extérieure de sa zone, c'est-à-dire qu'il ne connaît pas une route valide pour ce paquet, il l'achemine vers ses nœuds périphériques. Ces derniers maintiennent les informations de routages des zones voisines, ce qui leurs permet de choisir le lieu où le paquet en question doit être acheminé. Pour cela, on utilise un protocole de résolution bordercast (***B**ordercast **R**esolution **P**rotocol* ou ***BRP***) [117] au lieu d'inonder tous les nœuds du réseau avec ces requêtes (les requêtes de recherche de route). Le protocole IERP et le protocole BRP seront présentés dans les sections 4.2.2 et 4.2.3 respectivement.

On peut dire que le protocole ZRP est constitué de plusieurs composants, qui, seulement ensemble, peuvent fournir le bénéfice de routage total à ZRP. Chaque composant travail indépendamment de l'autre, et peut utiliser différentes technologies dans l'ordre de maximiser l'efficacité dans sa propre région. Par exemple, un protocole réactif comme AODV [44] peut être utilisé comme l'***IERP***, alors que l'IARP est généralement un protocole proactif comme OLSR [118].

La figure 4.2 illustre les différents protocoles nécessaires pour ZRP et leurs interactions.

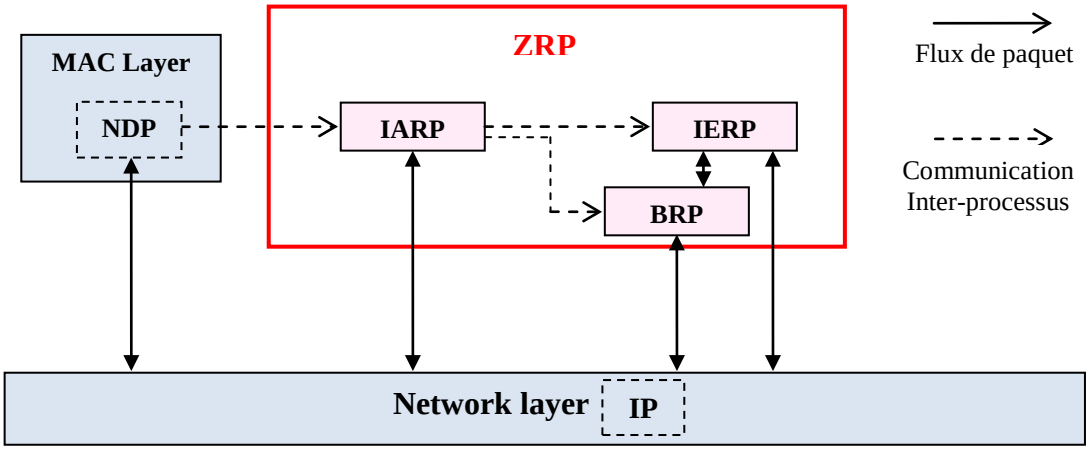


FIG 4.2 *– Les composants du protocole ZRP*

4.2.1 Le protocole IARP

Comme ZRP suppose que la découverte d'un voisin local est implémentée dans la couche ***MAC***, et est fournie par le protocole ***NDP***, le premier protocole qui fait partie de ZRP est le protocole Intrazone ***IARP***. Ce protocole est utilisé par un nœud afin de communiquer avec les autres nœuds internes de sa zone. Un sous ensemble important des nœuds de la zone s'appelle « ***les nœuds périphériques*** », cet ensemble contient tous les nœuds dont la distance qui les sépare du nœud central est exactement égale au rayon de cette zone. Sur la figure 4.1, par exemple, l'ensemble des nœuds périphériques de la zone locale du nœud A est donné par {B, D, E, F, H, J}, avec **ρ** = 2.

Comme le voisinage local d'un nœud peut changer très rapidement, et comme les changements de la topologie locale ont un impact très important sur le comportement du nœud par rapport au changement que subit l'autre partie du réseau, l'IARP doit faire partie de la famille des protocoles proactifs, orientés ou basés table.

Un nœud a besoin de mettre à jour ses informations de routage continuellement afin de déterminer ses nœuds périphériques, et maintenir aussi une carte à partir de laquelle les noeuds locaux peuvent être atteints. IARP impose à chaque nœud de diffuser son information de routage uniquement à l'intérieure de sa propre zone. Cela veut dire que chaque nœud doit maintenir ***une table de routage locale*** qui va lui permettre de retrouver le chemin vers n'importe quel autre nœud de cette même zone. En utilisant IARP, un nœud peut avoir la route recherchée sans aucun délai si et seulement si la

destination se trouve à l'intérieure de sa zone. Il permet en plus d'optimiser une route locale à travers les chemins redondants et aussi lorsque la route ayant le plus petit nombre de sauts soit détectée. En cas de pannes de liaisons, IARP se dévie et parcourt un autre chemin en s'aidant toujours des routes alternatives disponibles dans sa table de routage locale.

Comme c'est déjà mentionné, il est possible qu'un nœud A envoie des messages vers un autre nœud B, mais ce dernier, et à cause des limitations dans sa longueur de signal (causées par une interférence par exemple), ou d'une basse puissance de transmission, ne peut pas atteindre le nœud A. Pour cela, il est important au protocole IARP utilisée de fournir un support pour les liens unidirectionnel entre les nœuds locaux (le problème des liens unidirectionnels).

Dans le but d'adapter un protocole état-de-lien proactif classique pour l'utiliser comme protocole IARP, sa portée doit être limitée à la taille de la zone « ρ ». Ceci peut être implémenté par l'ajout d'un paramètre de durée de vie (**T**ime **T**o **L**ive ou ***TTL***) à chaque requête de recherche de route. Ce paramètre ***TTL*** doit être initialisé à « ρ - 1 », il se décrémente à chaque fois que la requête porteuse soit reçue par un voisin, et ainsi de suite jusqu'à ce que le ***TTL*** devient nul ou que la requête soit refusée [115].

4.2.2 Le protocole IERP

IERP est le composant de routage réactif global de ZRP qui prend l'avantage de la topologie locale déjà connue par les nœuds de la zone à travers IARP, l'utilisation de l'approche réactive permet la communication avec les nœuds des différentes zones. Les requêtes de routes dans IERP sont issues à la demande, le délai de découverte de route (contrairement au protocole IARP, où la route est immédiatement disponible) est minimisé à travers l'utilisation d'un certain type de diffusion, une approche dans laquelle le nœud n'envoie pas la requête vers tous les nœuds du réseau, mais uniquement vers les nœuds périphériques de sa zone. En plus, le nœud n'envoie pas la requête en arrière vers ses prédécesseurs à partir desquels il a reçu la requête, même s'ils sont des nœuds périphériques, et ce afin d'empêcher la formation des boucles.

Dans le but d'adapter un protocole de routage réactif classique pour l'utiliser comme protocole IERP, il est nécessaire d'éliminer les mises à jour proactives des routes locales, car cette fonctionnalité est offerte par le protocole IARP utilisé. En plus, le protocole réactif adapté doit être capable de prendre l'avantage des informations locales fournies par IARP, bien que de changer la manière de découverte de route: au lieu de diffuser la requête vers tous ses voisins, il doit utiliser un protocole de routage ***Bordercast*** (**BRP**) permettant d'initier la recherche de route uniquement avec les noeuds périphériques. Cette approche de diffusion est contrôlée par un mécanisme de contrôle de requête qui sera décrit dans la section 4.2.4.

4.2.3 Le protocole BRP

Le protocole de résolution bordercast (***B**ordercast **R**esolution **P**rotocol ou **BRP***) est utilisé dans ZRP pour diriger les requêtes de recherche de route, initiées par le protocole réactif global IERP, vers les nœuds périphériques de la zone interrogée, tout en éliminant les requêtes redondantes et en maximisant l'efficacité du protocole de routage adapté. En faisant ceci, le BRP utilise la carte fournie par le protocole IARP (les informations de routage déjà enregistrées), ce qui lui permet de construire un arbre bordercast. Contrairement à IARP et IERP, le BRP n'est pas vraiment un protocole de routage mais plutôt un service de distribution des paquets.

La communication bordercast ou le bordercasting utilise les informations offertes par l'IARP, et dirige la requête de recherche de route vers l'extérieure de la zone, à travers une communication multicast qui concerne uniquement les nœuds périphériques entourant cette zone. Ces nœuds périphériques, une fois qu'ils reçoivent cette requête, ils cherchent à l'intérieure de leurs zones l'existence de la destination en question. Si cette dernière n'est pas dans la zone d'un nœud périphérique donné, il reprend la même procédure de recherche avec ces propres noeuds périphériques. Et ainsi de suite jusqu'à ce que tout les nœuds du réseau soient couverts.

Deux approches différentes permettent de réaliser la procédure de la recherche en bordercast. La première s'appelle « ***R**oot **D**irected **B**ordercast* » ou ***RDB***, et qui signifie un bordercast orienté source ou racine. L'autre s'appelle « ***D**istributed **B**ordercast* » ou ***DB*** et qui signifie un bordercast distribué. Le bordercast orienté source a besoin de nœud source et des nœuds périphériques pour la construction des arbres bordercast tout en ajoutant aux requêtes de recherche de route une information supplémentaire. Ceci résulte en une charge additionnelle qui augmente avec l'augmentation du rayon de la zone, cachant ainsi les bénéfices de ZRP. Le bordercast distribué impose à chaque nœud de maintenir une zone de routage étendue, ce qui fait augmenter le nombre des informations de routage échangées à l'intérieure des zones étendues. Cependant, cette deuxième approche réduit les conditions de découverte des routes [119].

Le protocole BRP garde trace des nœuds qui ont déjà reçus la requête de recherche de route, ce qui lui permet d'élaguer les branches de l'arbre bordercast constitué des nœuds qui ont déjà reçus (et/ou relayés) la même requête. Quand un nœud reçoit une requête de recherche de route destinée à un autre nœud qui ne se trouve pas dans sa zone de routage locale, il construit son arbre bordercast, en éliminant les nœuds périphériques déjà couverts, et envoie sur cet arbre la requête de recherche de route vers ses nœuds périphériques. Chacun de ces nœuds périphériques cherche dans sa zone la destination voulue, si cette dernière ne se trouve dans aucune de ces zones, les nœuds périphériques passe alors à la recherche bordercast en commençant par la construction des arbres bordercast de la même manière et avec les mêmes conditions précédentes. Si la destination cherchée est détecté par un certain nœud du réseau, ce dernier renvoie donc une réponse vers la source, en parcourant les chemins inverses déjà enregistrés par

les nœuds intermédiaires lors de la recherche de route. Pour mieux expliquer cette procédure, voici l'exemple illustré dans la figure suivante.

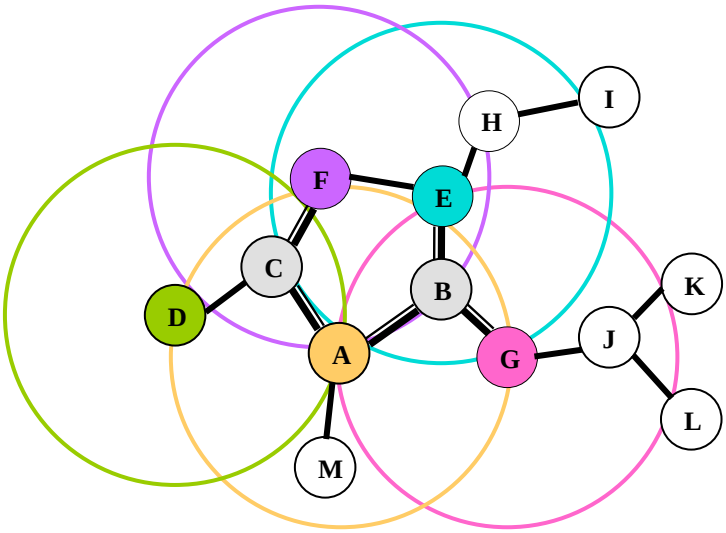


FIG 4.3 *– Exemple du protocole BRP*

La figure 4.3 illustre un exemple de BRP dans lequel le nœud A désire envoyer ses messages vers le nœud L. Mais comme il n'existe pas encore une route valide entre ces deux nœuds, la source A commence par chercher sa destination L dans sa table de routage locale. Comme L n'est pas dans la zone locale de A, ce dernier passe à la construction de son arbre bordercast dont les nœuds feuilles sont les nœuds périphériques du nœud A, et qui sont : D, E, F et G. Chacun de ces noeuds reçoit une requête de recherche de route vers le nœud L, depuis les nœuds intermédiaires B et C. En recevant cette requête, le nœud D cherche dans sa zone de routage la destination L. Mais L n'est pas dans la Zone de D, en plus il ne peut pas construire son arbre bordercast car il n'a pas de nœuds périphériques non parcourus dans sa Zone (A et F ont déjà envoyé et/ou reçu la requête), d'où la procédure de recherche s'arrête à ce niveau par rapport à D. Les nœuds périphériques F et E cherchent dans leurs zones de routage local le nœud L, mais ce dernier ne se trouve dans aucune de ces zones. Ces premiers passent alors à la construction de leurs arbres bordercast pour envoyer la requête à leurs propres nœuds périphériques non encore parcourus. Le tableau 4.1 résume les étapes de la procédure de recherche et de reconstruction de l'arbre bordercast.

Reçu depuis	La source	Relayeur de l'arbre bordercast non parcourus	Nœuds périphériques	
			Couverts	Non Couverts
———	A	B, C, M	——	D, E, F, G
B	E	H	A, C, G	I
B	G	J	A, E	K, **L**
C	F	———	A, B, D, H	———
C	D	———	A, F	———
E	H	I	F, B	———
H	I	———	E	———
H	I	———	E	———

***FIG 4.1** – Etapes de recherche de route en utilisant le BRP*

Eventuellement, la requête de recherche de route envoyée de A vers B est reçue par le nœud G, qui trouve la destination L dans sa zone de routage locale. Le nœud G n'achemine plus la requête, mais il envoie une réponse de découverte de route en arrière vers la source A, tout en indiquant le chemin à parcourir pour l'envoie des messages : A— B — G — J — L.

Cet exemple démontre l'efficacité de la recherche en bordercast en la comparant avec la recherche par diffusion. Si le réseau est constitué de liaisons point-à-point, la recherche en bordercast génère 5 transmissions de requêtes. Par contre, la recherche par inondation va générer 13 transmissions point-à-point [117].

4.2.4 Le mécanisme de contrôle de requêtes

Le ZRP à besoin aussi d'un mécanisme de contrôle de requête efficace dans le but de générer un trafic de contrôle inférieure à celui échangé par un protocole purement proactif lors d'échange d'informations de routes, ou celui généré par un protocole purement réactif lors d'exécution de la procédure de découverte de route. Le mécanisme de contrôle de requête utilisé par ZRP contient les procédures suivantes : (***Q**uery **D**etection* ou ***QD=QD1/QD2***) qui veut dire la détection de requête, (***E**arly **T**ermination* ou ***ET***) qui veut dire la terminaison précoce, (***L**oopback **T**ermination ou **LT***) qui veut dire la terminaison des boucle en arrière et (***R**andom **Q**uery **P**rocessing **D**elay* ou ***RQPD***) qui veut dire le délai de traitement de requête aléatoire.

Pour détecter si une zone de routage a été déjà interrogée par la requête, deux niveaux de détection sont possibles: le premier est le ***QD1*** (Query Detection 1), le second est le ***QD2*** (Query Detection 2). Dans ***QD1***, quand les nœuds relayeurs reçoivent la requête pour l'envoyer vers les nœuds périphériques, ces premiers enregistrent

l'adresse de la zone émettrice. ***QD2*** est utilisée par les réseaux à canal de diffusion unique, le nœud se trouvant dans le rang de transmission d'un nœud relayeur peut entendre la requête, et enregistre ensuite l'adresse de la zone émettrice. Dans la figure 4.4, le nœud A diffuse la requête de recherche de route vers ses nœuds périphériques D et F. Les nœuds relayeurs B et C marquent la zone du noeud A comme zone interrogée ou couverte (***QD1***). Si le réseau est à canal unique, le noeud E, peut entendre le trafic envoyé par A à travers l'envoie de la requête depuis C vers F, et marque la zone de A comme zone interrogée (***QD2***).

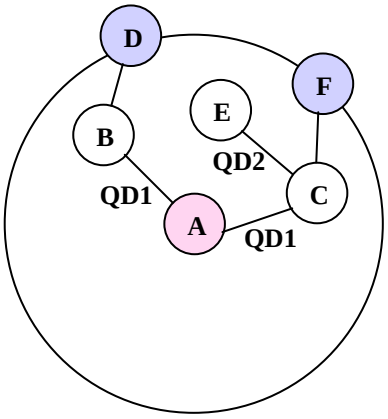


FIG 4.4 *– Les niveaux de détection QD1 et QD2*

Il est simple de détecter qu'un nœud donné est déjà couvert, cependant, ce n'est pas tout. Le protocole a besoin d'élaguer tous les paquets qui vont être envoyés vers un nœud déjà interrogé par la même requête. Ceci est réalisé par l'utilisation d'une procédure qui s'appelle « ***E****arly* ***T****ermination ou* ***ET*** », et compte évidemment sur la procédure ***QD***. Le principe de ***ET*** consiste à élaguer n'importe quelle branche qui débute depuis le nœud relayeur et qui va, soit vers un nœud périphérique déjà couvert, ou vers un nœud périphérique qui a déjà relayé cette même requête.

Dans la procédure (***L****oopback* ***T****ermination* ou ***LT***), les routes qui bouclent en arrière, c'est à dire vers la zone du nœud demandeur, sont éliminés du fait qu'un tel nœud peut être atteint localement en utilisant le protocole IARP.

Le delai (***R****andom* ***Q****uery* ***P****rocessing* ***D****elay* ou ***RQPD***) donne au nœuds relayeurs une autre chance avant d'élaguer les branches descendantes. Il est considéré avant la construction de l'arbre bordercast et l'exécution de la procédure ***ET*** [114].

Dans l'ordre d'éliminer en plus les diffusions non nécessaires, ***le BRP*** peut implémenter la procédure (***S**elective **B**ordercasting ou **SB***). Dans cette approche, un nœud a besoin de connaître l'information topologique du réseau à l'intérieure d'une zone étendue de rayon (**2ρ – 1**). Donnant cette information, un nœud peut éliminer certains nœuds périphériques de sa zone et qui sont déjà enregistrés dans sa liste des récepteurs bordercast, et ce dans le cas où les nœuds périphériques externes (de la zone étendue) les couvrent. La figure 4.5 montre la manière dont le nœud A est capable de supprimer le nœud C depuis son arbre bordercast, puisque les nœuds G et H peuvent être atteints depuis les nœuds B et D respectivement [119].

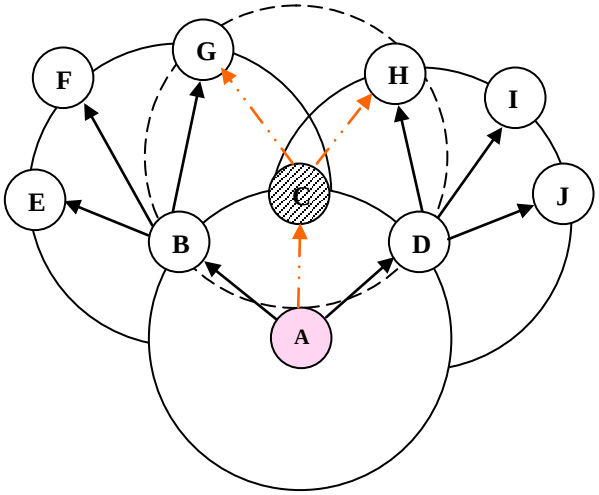


***FIG 4.5** – Selective Broadcasting*

Dans un autre contexte de ZRP, le protocole BRP peut être vue comme une jointure qui lie les deux protocoles IARP et IERP ensemble, son but est de prendre tout avantage des composants proactifs et réactifs, et dans une bonne utilisation. Le protocole BRP est décrit en détail dans [117].

4.3 Le protocole multicast MZRP

Le MZRP est un protocole de routage multicast à arbre partagé. Les messages de « Leader ou chef » du groupe multicast sont diffusés dans la totalité du réseau pour informer les nœuds de l'existence d'un groupe multicast et de son leader. En adaptant IARP à MIARP, le MZRP devient capable de garder trace des informations des groupes multicast dans la zone de routage locale de chaque noeud. En adaptant IERP à MIERP, le MZRP devient aussi capable de construire un arbre partagé couvrant le groupe multicast. Un tunnel IP est utilisé pour distribuer les paquets des données alors que les branches cassées sont réparées par MZRP, en bénéficiant de l'existence des routes redondantes.

4.3.1 Le leader du groupe

Le premier membre du groupe multicast devient le leader du groupe jusqu'à ce qu'il décide de quitter ce groupe, ou jusqu'à ce que deux partitions de l'arbre multicast se fusionnent. Le leader du groupe multicast se charge de diffuser périodiquement des messages pour informer tous les nœuds du réseau de son existence, ces messages s'appellent ***les messages de leader***.

4.3.2 MIARP

Dans l'arbre multicast, il existe deux sortes de nœuds : les nœuds d'acheminement multicast, et les membres du groupe multicast. La fonction des nœuds d'acheminement multicast consiste à interconnecter les membres du groupe multicast et acheminer les paquets de données sur cet arbre. ***Les messages d'adhésion*** à l'arbre multicast sont diffués à l'intérieure de la zone de routage locale par chaque nœud adhérant. Ces messages peuvent être envoyés séparément ou en cascade sur un paquet IARP original, périodiquement ou après déclanchement d'événement (Réveil ou Alarme).

Chaque nœud du réseau garde trace des leaders des groupes ainsi que des membres de ces groupes se trouvant à l'intérieure de sa zone de routage locale. Ceci permet aux nœuds du réseau de joindre le groupe multicast désiré avec une charge de recherche de route minimale s'il y a au moins un membre du groupe en question dans la zone locale de ce nœud. Sinon, la procédure de recherche de route à l'extérieure de la zone sera exécutée par le protocole MIERP.

4.3.3 MIERP

4.3.3.1 La procédure de recherche de route multicast

Un nœud qui désire joindre le groupe multicast et qui est déjà un nœud d'acheminement multicast à ce même groupe, commute tout simplement son état de nœud d'acheminement au membre de groupe multicast. Un nœud qui désire envoyer ses données vers le groupe multicast (on appelle ce nœud : « source »), envoie une requête de recherche de route (MRREQ). Deux types de requête (MRREQ) existent : les requêtes (MRREQ) unicast et les requêtes MRREQ bordercast, tout dépend de la position de ce nœud source par rapport à l'arbre multicast.

Si le nœud possède une route valide vers l'arbre multicast (vers n'importe quel nœud de l'arbre), il envoie une requête unicast MRREQ sur la route qui mène vers l'arbre multicast et attend la réception d'une réponse (MRREP). Les nœuds intermédiaires acheminent la requête unicast (MRREQ), et les chemins inverses sont enregistrés dans leurs tables de routage multicast. Quand la destination (un membre de l'arbre multicast) reçoit cette requête, elle répond par le message (MRREP). Si la requête unicast échoue ou si le nœud source ne trouve pas une route valide vers l'arbre multicast, il passe à l'envoie d'une requête bordercast (MRREQ). Cette requête passe d'abord sur l'arbre bordercast du nœud source, quand elle atteint les nœuds périphériques, chacun d'entre eux vérifie dans sa propre zone de routage l'existence d'une route valide vers l'arbre multicast. Si c'est le cas, la requête (MRREQ) est envoyée en unicast vers le membre en question, et la source attend une réponse (MRREP) sur cette requête. Sinon, chacun des nœuds périphériques envoie en bordercast la requête (MRREQ) vers les nœuds périphériques qui cherchent à leurs tours l'existence d'une route valide vers un membre de l'arbre multicast, et ainsi de suite jusqu'à ce que la destination soit détectée.

Les chemins inverses sont établis entre les nœuds intermédiaires. Si la source ne reçoit pas une réponse (MRREP) durant un délai limité, elle refais la même recherche en envoyant encore la requête (MRREQ) en bordercast, sans dépasser (***mrreq-retries***) tentatives de recherche. Après avoir dépasser ce nombre sans recevoir aucune réponse depuis la destination, la source confirme que ce groupe multicast n'existe pas dans tout le réseau. Cette source devient alors le leader du groupe en question et commence à diffuser ses ***messages de leader du groupe*** périodiquement.

4.3.3.2 La procédure de réponse de route multicast

Quand la destination reçoit la requête (MRREQ) pour un groupe multicast donné (car la destination peut être membre de plusieurs groupes multicast), si elle est membre de l'arbre multicast de ce même groupe multicast, elle envoie la réponse (MRREP) vers la source et attend un message d'activation de route multicast (MRACT) depuis le nœud source, et ce afin d'activer cette nouvelle branche de l'arbre multicast. Rappelons que la réponse (MRREP) envoyée par la destination passe par le chemin inverse qui mène à la source et qui a été déjà enregistré par les nœuds intermédiaires.

4.3.3.3 La procédure d'activation de route multicast

Après avoir envoyé sa requête de recherche multicast (MRREQ), la source peut recevoir plusieurs réponses multicast (MRREP) depuis différents chemins, pour cela, la source attend un délai égale à (***wait-for-more-reply***) milli seconde, après la réception de la première réponse. La source choisit donc la meilleure réponse reçue selon certaines métriques, et envoie ensuite le message d'activation (MRACT) vers la destination choisie et sur le chemin parcouru par la réponse. Les autres nœuds intermédiaires attendent la réception d'un telle message d'activation un certain délai, et élaguent enfin les routes inactives tracées par les réponses refusées, en les éliminant depuis leurs tables de routage (ce sont les chemins inverses déjà enregistrés par ces nœuds intermédiaires).

4.3.3.4 Exemple sur les procédures de MIERP

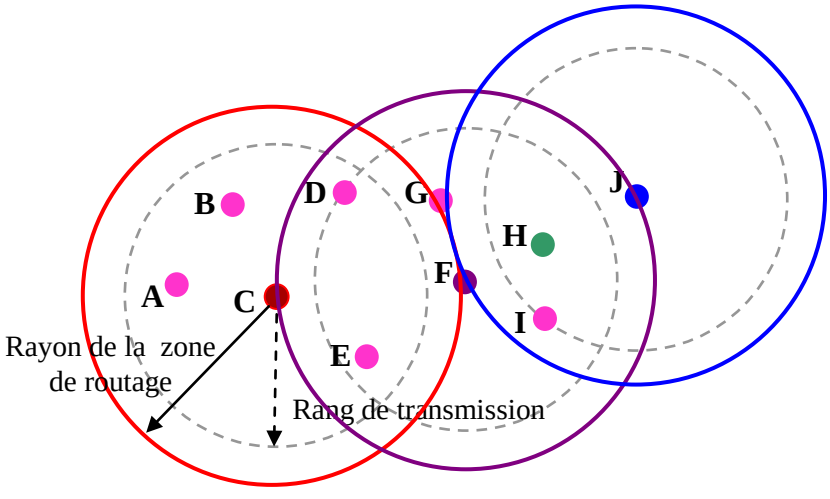


FIG 4.6 *– Avant la construction de l'arbre multicast*

Dans cet exemple (figure 4.6), le rayon de la zone est 2. Supposant que le nœud C désire joindre le groupe multicast (MG1), et supposant aussi que ce groupe ne possède pas encore des membres multicast. Donc, et après avoir cherché l'existence d'un membre du groupe en question dans sa zone locale, le nœud C envoie en bordercast la requête (MRREQ) vers ses nœuds périphériques sur son arbre bordercast. Quand les nœuds périphériques F et G reçoivent la requête, ils cherchent dans leurs zones un nœud membre de groupe multicast MG1, comme ce membre n'existe pas dans leurs zones, ils renvoient la requête vers leurs nœuds périphériques et sur leurs arbres bordercast, et ainsi de suite. Comme le groupe MG1 est supposé vide, le nœud C ne recevra aucune réponse sur sa requête. Après avoir faire « ***mrreq-retries*** » tentatives d'envoie de requête sans aucune réponse, le nœud C élit soi-même comme étant leader du groupe multicast MG1, et commence à diffuser ***les messages de leader du groupe*** dans tout le réseau. Il commence à diffuser aussi ***les messages d'adhésion à l'arbre multicast***, et ce à l'intérieure de sa zone de routage locale. Par conséquent, tous les nœuds du réseau sachent que le leader du groupe MG1 est le nœud C. Les nœuds A, B, D, E, F et G (les nœuds qui se trouvent dans la zone de routage de C) sachent qu'il y a un membre de l'arbre multicast du groupe multicast MG1 après la réception ***des messages d'adhésion*** à l'arbre multicast depuis C. Supposant maintenant que le nœud J désire joindre le groupe multicast MG1, comme il n'a pas de membre du groupe multicast dans sa zone locale, il envoie en bordercast la requête (MRREQ) vers ses nœuds périphériques F et G sur son arbre bordercast. Chacun de ces nœuds possède une route valide vers le leader C, quand ils reçoivent la requête depuis J, ils envoient en unicast la requête MRREQ vers le leader C. Le leader répond donc sur les deux requêtes reçues depuis F et G. Ainsi, le nœud J reçoit deux réponses (MRREP) sur sa même requête. Il choisit une parmi les deux et envoie le message d'activation de route (MRACT) vers la destination choisie. Dans cet exemple, le nœud J choisit la réponse reçue depuis G. Une nouvelle branche de l'arbre multicast du groupe MG1 est crée, et les nœuds intermédiaires H, G et D deviennent des nœuds d'acheminement multicast du groupe MG1, ils commencent donc la diffusion ***des messages d'adhésion*** à l'arbre multicast du groupe multicast MG1 à l'intérieure de leurs zones de routage locales. Si le nœud F décide de joindre le groupe MG1 par exemple, il peut avoir plusieurs chemins différents. Il peut choisir un ou plus d'entre eux pour envoyer sa requête unicast (MRREQ). Dans cet exemple, F choisit le nœud membre de l'arbre multicast G, pour envoyer sa requête unicast. Après les procédures de réponse et d'activation, une autre nouvelle branche est ajoutée à l'arbre multicast. Si H désire joindre le groupe multicast maintenant, il suffit juste de changer son état de nœud d'acheminement multicast au nœud membre du groupe multicast, car il est déjà dans l'arbre multicast de ce même groupe. La figure 4.7 illustre l'arbre de routage multicast du groupe multicast MG1 construit.

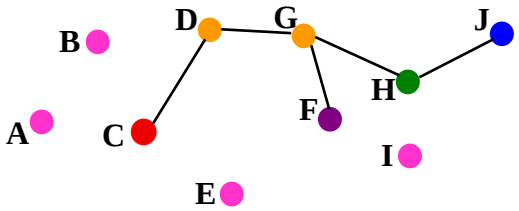


***FIG 4.7** – Après la construction de l'arbre multicast*

4.3.4 La maintenance de l'arbre multicast

4.3.4.1 La procédure d'élagage

Supposant qu'un membre du groupe multicast décide de quitter son groupe. Si c'est un nœud feuille de l'arbre multicast, il peut facilement élaguer soi-même de l'arbre multicast en envoyant un message d'élagage (***MPRUNE***) vers son prédécesseur dans cet arbre, en mettant le flag « ***prune*** » à « ***vrai*** ». Le nœud prédécesseur élague soi-même à son tour s'il n'est pas membre de groupe multicast, et s'il devient un nœud feuille de l'arbre après l'élagage de son successeur, et ainsi de suite. Dans le cas contraire, la procédure d'élagage doit s'arrêter. Supposant que le nœud qui désire quitter le groupe multicast n'est pas un nœud feuille de l'arbre multicast, il suffit simplement qu'il change son état de nœud membre de groupe multicast au nœud d'acheminement. Supposant maintenant que le leader du groupe multicast désire quitter son groupe, il envoie donc le message d'élagage (MPRUNE) vers l'un de ses nœuds successeurs, en positionnant le flag « ***new-leader*** » à « ***vrai*** », et si nécessaire, le flag « ***prune*** » aussi. Cette procédure d'élagage continue jusqu'à ce qu'il n'y aura plus de nœuds d'acheminement à élaguer. Un nœud membre du groupe multicast devient le nouveau leader du groupe. Le nouveau leader du groupe diffuse ***les messages de leader du groupe***, sur lesquels le flag « ***update_leader*** » est mis à « ***vrai*** », pour indiquer que le nouveau leader du groupe a été élu. Les autres ***messages de leader de groupe*** multicast seront diffusés simplement et périodiquement.

4.3.4.2 La réparation des cassures

La cassure d'un lien est détectée lorsque aucun paquet n'est reçu depuis le voisin et après avoir passé un délai égale à [« ***hello-interval »****(1+ « ***allowed-hello-loss »***)]. Ils existent plusieurs autres mécanismes de détection de voisins qui peuvent accomplir cette même tâche. Quand on détecte un lien cassé, le nœud successeur se charge de réparer de ce lien. Ce qui empêche la formation des boucles durant la procédure de réparation. Quand le nœud successeur possède une route valide vers un membre du groupe multicast à l'intérieure de sa zone, une réparation locale sera effectuée en envoyant le message unicast (MRREQ) dont le flag « ***repair*** » est mis à « ***vrai*** ». Dans le cas où le nœud successeur ne trouve pas une route valide vers les membres du groupe multicast depuis sa zone, la requête (MRREQ) sera diffusée dans la totalité du réseau. Chaque membre de l'arbre multicast dont le nombre de sauts qui le sépare du leader est inférieur ou égale au nombre de sauts qui sépare le nœud successeur (réparateur de lien cassé) du leader, peut répondre sur la requête avec le message (MRREP), et ce afin d'empêcher la formation des boucles durant le processus de réparation. Si le nœud successeur ne reçoit aucune réponse (MRREP), il répète cette procédure de recherche un certain nombre de fois égale à « ***mrreq-retries »***, il comprend vers la fin du délai que le réseau en question a été partitionné. Dans ce cas, un nouveau leader du groupe sera élu de la même manière décrite dans la section 4.3.4.1. D'autre part, le nœud prédécesseur du lien cassé peut élaguer soi-même s'il devient bien sûr un nœud feuille et d'acheminement en même temps, ceci après avoir passé un délai égale à « ***route-expiration*** » (milli seconde).

4.3.4.3 La reconnection des partitions

Les ***messages de leader du groupe*** peuvent être utilisées pour détecter la reconnection des partitions dans le réseau. Quand un membre de l'arbre multicast reçoit ***un message de leader*** du même groupe mais depuis un leader différent à celui de départ, il dénonce ceci à son premier leader si l'adresse IP de ce nouveau leader est supérieure à celle de l'ancien leader. Sinon, ce membre de l'arbre ignore les messages de ce nouveau leader. Revenant au cas où le nouveau leader possède une adresse IP supérieure à celle de l'ancien, donc le leader d'adresse IP minimale peut recevoir plusieurs réclamations depuis plusieurs membres de l'arbre multicast en même temps. Ce leader d'adresse minimale interroge un seul nœud membre de l'arbre multicast pour effectuer la reconnection, ceci afin d'éviter la formation des boucles. Le nœud interrogé, et après avoir achever la procédure de reconnection, informe le leader d'adresse minimale que l'opération de reconnection est terminée avec succès. En recevant cette information, le leader du groupe de plus petite adresse IP diffuse des ***messages de leader de groupe*** mais cette fois-ci avec le flag « ***update-leader*** » mis à « ***vrai*** ».

4.3.5 La transmission des paquets de données sur un tunnel IP

Les paquets de données peuvent être laissés tomber dans la couche MAC soit à cause d'une cassure de lien, ou à cause de l'échec RTS-CTS lors d'une contention d'accès aux media. Un lien cassé est détecté si aucun paquet n'est reçu depuis le voisin et ce durant une période égale à [« ***hello-interval*** »*(1+ « ***allowed-hello-loss*** »)]. Dans les scénarios à grande mobilité, les cassures des liens peuvent ne pas être détectées à cause des durées minimales relatives durant lesquelles le lien reste cassé. A cause de l'absence des routes redondantes dans les protocoles basés arbre, une sévère dégradation de performances peut avoir lieu sous les conditions mentionnées précédemment : l'échec RTS-CTS, les cassures des liens fréquentes et de durée minimales, et relativement les longues durées de réparations des liens cassés effectuées par le protocole de routage multicast. L'utilisation d'un tunnel IP sous ces conditions peut améliorer significativement la performance du protocole. Avec une zone de routage de rayon supérieur ou égal à deux (>=2), il peut y avoir un ou plusieurs nœuds dans la région d'intersection des deux zones de routage des deux nœuds voulant échanger des paquets de données entre eux. Un tunnel IP peut être crée temporairement entre ces deux nœuds en passant par un nœud intermédiaires se trouvant dans la région d'intersection de leurs zones. Examinant maintenant la faisabilité d'utiliser un tunnel IP en se référant à la figure 4.8 [114].

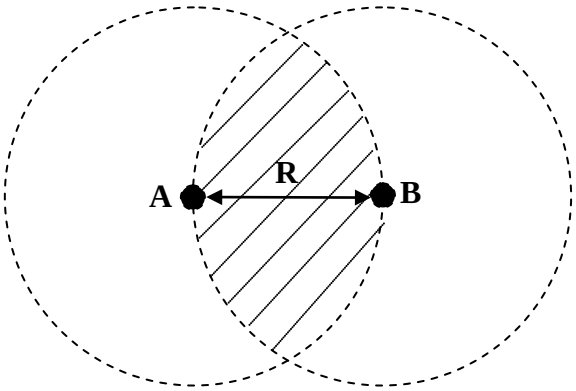


FIG 4.8 *– Illustration de la faisabilité d'un tunnel IP*

Dans cet exemple, la région d'intersection est d'aire égale à $2\left(\frac{\pi}{3}-\frac{\sqrt{3}}{4}\right)R^2$. Supposant que l'aire du réseau est égale à S, le nombre de nœuds mobiles dans tout le réseau est égal à N, et le rang de transmission est égal à R. Supposant encore que le lien entre les deux nœuds A et B vient de se casser. S'il existe au moins un nœud dans la région d'intersection (la partie hachurée) alors, les nœuds A et B peuvent échanger leurs

données sur un tunnel IP qui passe par le nœud qui se trouve dans la région d'intersection. La probabilité qu'il ai au moins un nœud dans cette région d'intersection est approximativement égale à $1-\left(1-\frac{2\left(\frac{\pi}{3}-\frac{\sqrt{3}}{4}\right)R^2}{S}\right)^{N-2}$, où on a approximé la distance entre A et B à R aussi. Pour un calcul numérique, si R=250m, S=1000x1000m^2, et N=50 alors, la probabilité prend la valeur 97,84 %. Ce qui signifie que l'utilisation d'un tunnel IP dans de tels cas augmente les performances du protocole d'une manière significative.

Une simulation de protocole MZRP a été faite par « **Xiaofeng Zhang** et **Lillykutty Jacab** »[❖] en utilisant le simulateur NS-2 [120] [121] [122]. Quatre métriques de performances ont été utilisées pour évaluer l'efficacité du protocole, il s'agit du taux de livraison des paquets, de la charge de routage multicast normalisée, du rapport nombre de paquets de données transmis par nombre de paquets de données délivrés et enfin du délai de découverte des routes multicast. Le protocole MZRP a été comparé avec le protocole multicast basé maille ODMRP [111] [123] afin d'avoir une bonne évaluation de performances. D'après les auteurs, on peut conclure que ce n'est pas seulement les protocoles basés maille peuvent fournir une meilleur performance de routage dans les environnements mobiles, mais les protocoles basés arbre peuvent le faire aussi. Le protocole MZRP est un protocole scalable (qui marche avec différentes échelles) pour un nombre important de groupes multicast et d'émetteurs aussi, et possède une charge de routage multicast normalisée inférieure à celle d'ODMRP.

Dans ce qui suit, je vais proposé une extension ou amélioration à ce protocole de routage multicast (MZRP), qui consiste à introduire une certaine technique appelée ***M**ulti**P**oint **R**elay* ou ***MPR*** [48] [96]. Cette technique permet de minimiser la charge de contrôle, d'obtenir une grande scalabilité et résout efficacement le problème des liens unidirectionnels dans les communications sans fil [96]. La section suivante commence par présenter la simulation de la technique MPR. Je passerai ensuite à l'analyse du protocole proposé MZRP-MPR.

❖ *Centre for Internet Research School of Computing — National University of Singapore — 3 Sciences Drive 2, Singapore 117543.*

4.4 Multipoint Relay et l'algorithme de sélection

Multipoint relay (MPR) ou la technique des relais multipoints vise à réduire le nombre de retransmissions inutiles, lors d'une diffusion généralisée d'un message donné. La transmission radio est par défauts une inondation à tous les voisins. Les voisins du second niveau peuvent être couverts par un ou plusieurs nœuds du premier niveau. L'idée se résume alors à choisir le nombre de répéteurs nécessaires pour atteindre tous les nœuds du second niveau d'un nœud particulier. Cet ensemble forme un arbre recouvrant et s'appel l'ensemble des relais multipoints. Cet ensemble permet d'économiser la bande passante et réduit le nombre de messages reçus en plusieurs copies par un nœud.

Pour diffuser un message dans tout le réseau, chaque nœud désigne ses voisins relais multipoints qui joueront le rôle des premiers répéteurs de ce nœud. Récursivement et en répétant ce processus, le message diffusé atteint la totalité des nœuds. [124] Démontre que la connaissance de deux niveaux de voisinage permet la diffusion des messages dans tout le réseau.

La particularité de cette technique est qu'elle fonctionne d'une manière totalement indépendante et distribuée ; chaque nœud calcule ses relais multipoints en se basant sur la connaissance de son voisinage à deux sauts. Le nœud doit informer ses voisins de leur nouveau rôle. Dans un environnement mobile avec une topologie changeante d'une manière imprévisible, l'ensemble des relais multipoint doit être recalculé à chaque fois qu'on détecte une modification dans le voisinage à deux sauts. C'est pour cette raison que le statut de relais multipoint est conditionné par le voisinage pour une durée limitée.

La figure 4.9 montre un exemple du nombre d'émissions nécessaires pour atteindre tous les nœuds du second saut en utilisant l'inondation pure. L'inondation pure stipule que chaque nœud émet une fois le message reçu ; ce qui en résulte en six retransmissions du message envoyé par la source. On remarque aussi que la source va recevoir son propre message six fois de suite s'il n'y a pas de collision dans le réseau.

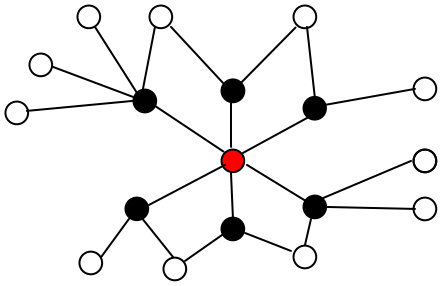


***FIG 4.9** – Graphe de voisinage (6 retransmissions)*

Dans la deuxième figure 4.10, on applique la technique des relais multipoint. Le nœud central choisit ses répéteurs, en occurrence les nœuds colorés en noir. Dans ce cas de figure ce nombre est quatre. Donc, le relayage du message demande quatre transmissions au lieu des six utilisées pour une inondation classique.

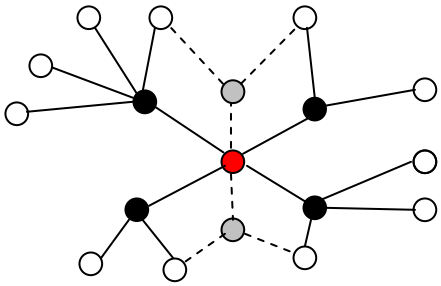


***FIG 4.10** – Graphe de voisinage (4 retransmissions)*

L'élection des relais multipoint permet donc de réduire le nombre de retransmissions inutiles. Choisir cet ensemble peut sembler facile à première vue. Mais en réalité, ce genre de problème revient à trouver un ensemble dominant qui appartient à la famille des problèmes NP complet [125] [126] [127]. Vu le contexte de réseau mobile ad hoc, l'élection de relais multipoint doit se faire rapidement et avec le minimum de calculs possible. La solution doit combiner rapidité et efficacité. L'idée de trouver une solution intermédiaire avec un algorithme simple à implémenter et qui donne un résultat proche de l'optimal dans la majorité des cas. Une heuristique pour l'élection des relais multipoint a été proposée dans [128] et analysée dans [129] [130]. Cette heuristique donne une solution optimale à *log(n)* prés avec *n* le nombre de voisin du nœud calculant l'ensemble des relais multipoint.

4.4.1 Heuristique de choix des relais multipoint

La connaissance des voisins du second niveau d'un nœud permet de procéder à l'élection de ses nœuds multipoint relais. Notons que, cette opération d'élection est faite au niveau de tous les nœuds du réseau. Son but est de trouver un ensemble de nœuds voisins qui couvrent tous les nœuds du second niveau. Généralement, on a plusieurs solutions 4.11. Ce nombre dépend essentiellement de la façon dont sont interconnectés les nœuds.

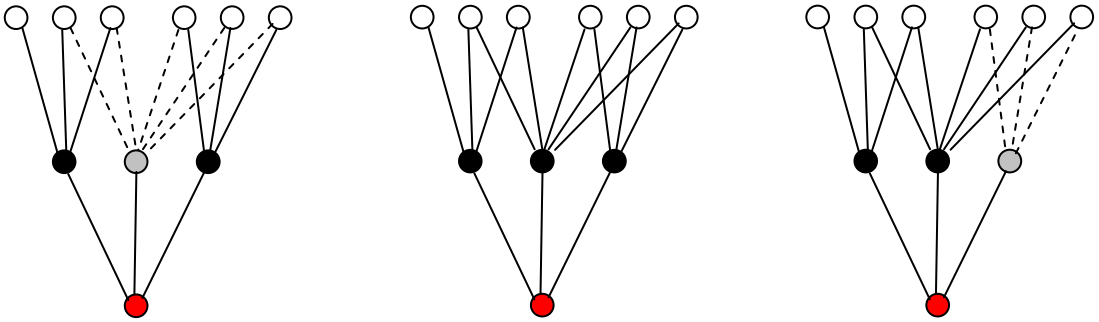


FIG 4.11 *– Les trois solutions possibles de l'ensemble des relais multipoint (nœuds noirs)*

4.4.2 Exemple de déroulement de l'algorithme de sélection

Prenons un exemple pour mieux illustrer le principe de cet algorithme (le même que précédemment). On se propose de chercher l'ensemble des relais multipoint du nœud N (figure 4.12)

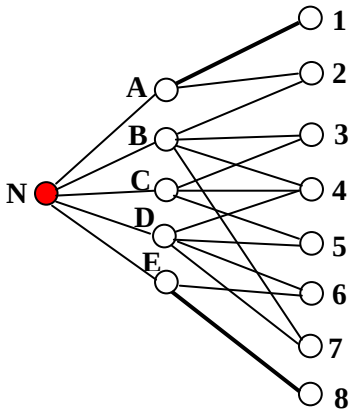


FIG 4.12 *– Sélection des voisins possédant un seul lien avec un nœud du second niveau*

La première étape consiste à trouver les nœuds du premier niveau possédant des liens uniques avec un nœud du second niveau. Ces noeuds feront partie nécessairement de l'ensemble des relais multipoints M afin que les voisins à deux sauts soient totalement couverts. Selon l'exemple que l'on a pris, A et E sont dans ce cas (couvrant 1 et 8). On les insère dans M, et on élimine tous les nœuds du second niveau couverts par A et E (figure 4.13) : les nœuds 1,2 ainsi que 6 et 8.

La deuxième étape est une boucle : à chaque itération, on cherche le nœud du premier niveau qui couvre le maximum de nœuds du second niveau. On l'ajout dans l'ensemble M et élimine ses nœuds du second niveau. La boucle prend fin naturellement lorsqu'il n'y a plus de nœuds du second niveau (N2= {}). La figure 4.13 illustre les différentes étapes du déroulement de cet algorithme.

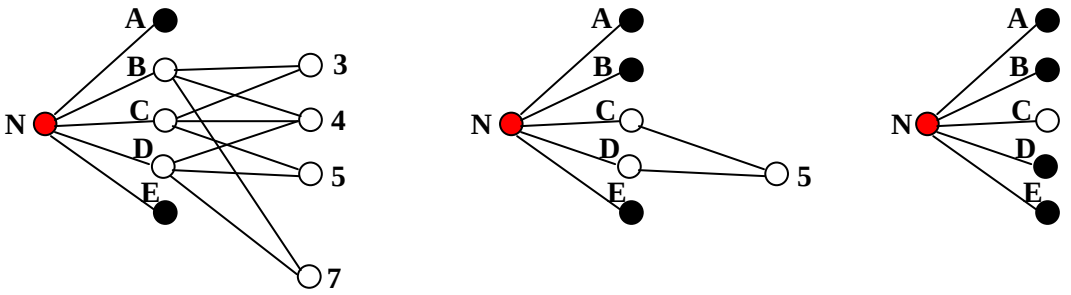


***FIG 4.13** – Illustration de l'algorithme de sélection des relais multipoint*

Plus formellement l'algorithme de sélection des relais multipoint se présente comme suit :
On désigne par *N(x)* l'ensemble des voisins directs de *x*, par *N2(x)* l'ensemble des voisins du second niveau, et *MPR(x)* étant l'ensemble des relais multipoint de *x*.

1. Commencer par un ensemble de relais multipoint vide *MPR(x)* = { }.

2. Choisir les nœuds de l'ensemble des voisins *N(x)* qui sont les seuls ayant un lien avec un voisin du second niveau. Ajouter ces nœuds sélectionnés de *N(x)* à l'ensemble *MPR(x)*, et éliminer tous les nœuds du second niveau couverts par ces derniers de l'ensemble *N2(x)*.

3. Tant que $N2(x) \neq \{\ \}$ refaire

   (a) Calculer le degré de chaque nœud dans *N(x)*. Le degré pour un nœud est le nombre des voisins du second niveau couverts par celui-ci présent dans *N2(x)*.

(b) Ajouter le nœud de *N(x)*, ayant le degré maximal à l'ensemble des relais multipoint *MPR(x)*, et enlever tous les nœuds du second niveau couverts par celui-ci de *N2(x)*.

Comme indiqué plus haut, cette heuristique ne fournit pas la solution optimale et en voici un contre-exemple sur la figure 4.14. La solution donnée par l'algorithme est de trois relais multipoint et la solution optimale est de deux relais multipoint.

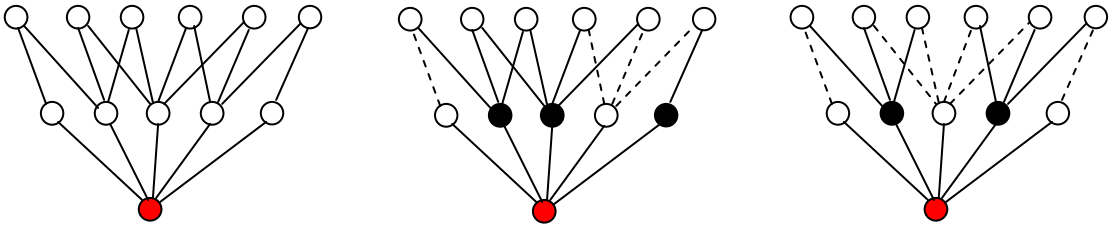


***FIG 4.14** – un contre exemple de l'heuristique du choix des relais multipoint*

Dans [96] une simulation de la technique des relais multipoint a été réalisée afin d'évaluer les performances de cette technique à celle de la diffusion par inondation. Cette étude a été effectuée dans le cas des réseaux larges et denses (1024 nœuds) en considérant le fading des liens entre les voisins [129].

D'après les résultats de cette simulation, l'auteur constate que la diffusion se termine plus rapidement en utilisant cette technique de diffusion optimisée. Ceci est dû au nombre plus réduit de nœuds relayeurs. Les nœuds sont moins submergés par les copies multiples du même message, ce qui fait, que l'on ne gaspille plus la bande passante. Notons que dans le cas de l'inondation, le nombre de réceptions est égal au nombre de retransmissions.

Par contre, la technique des relais multipoint n'est pas fiable et donne des résultats médiocres lorsque la probabilité d'erreur de réception augmente et dépasse 50%. Dans cette zone-là, la technique de l'inondation se comporte bien et donne une meilleure couverture du réseau avec une consommation de bande passante plus importante.

En conclusion, je peux dire que la technique de l'inondation est puissante, mais elle génère un trafic important avec beaucoup de duplications inutiles. La technique des relais multipoint se présente comme une meilleure alternative pour éviter le gaspillage de la bande passante.

4.5 Le protocole multicast MZRP-MPR

Le protocole MZRP-MPR proposé est un protocole multicast à arbre partagé. Il est conforme pour des topologies dynamiques avec des nœuds mobiles, et utilise la technique des relais multipoint afin de réduire la charge de contrôle et minimiser le délai des transmissions. MZRP-MPR conserve le même fonctionnement interne que MZRP sauf qu'il utilise la technique MPR à la place de l'inondation avec certain messages de contrôle. Dans ce qui va suivre, pour toute diffusion optimisée par MPR, j'utiliserai le verbe MPR-diffuse.

4.5.1 Le leader du groupe multicast dans MZRP-MPR

Pour informer tous les nœuds de son existence, le premier membre du groupe multicast qui devient le leader de ce même groupe MPR-Diffuse les ***messages de leader du groupe*** dans la totalité du réseau. De cette manière, la consommation de la bande passante sera minimisée par rapport à la diffusion par inondation. Et les ***messages de leader du groupe*** submergent tous les nœuds du réseau plus rapidement, ce qui convient plus avec la nature dynamique de la topologie des réseaux ad hoc.

4.5.2 Le protocole MIARP-MPR

De la même manière précédente, tous les paquets qui ont été diffusés par inondation dans le composant MIARP du protocole de base MZRP seront diffusés par la technique MPR (ou MPR-Diffusés) dans ce nouveau protocole MZRP-MPR.

D'après la section 4.3.2 de ce chapitre, la participation d'un nœud du réseau au routage multicast doit être propagée à l'intérieure de sa zone locale, il s'agit donc de MPR-diffuser ses ***messages d'adhésion*** à l'arbre multicast à l'intérieure de la zone locale du nœud en question. En s'aidant bien sûr des informations fournies par le protocole de base MIARP (la table de routage local), le nœud peut rapidement choisir son ensemble des relais multipoint (MPR), en exécutant l'algorithme de sélection MPR adéquat. Le bénéfice d'utilisation de ce type de diffusion ici s'avère plus important lorsque la taille de la zone est grande (>2); car le nombre de transmissions par inondation sera réduit d'une manière remarquable en utilisant MPR.

Afin de minimiser les délais de recherche d'un membre du groupe multicast, chaque nœud dans le réseau maintient une certaine structure de données dans laquelle il enregistre les groupes multicast, les membres multicast correspondant se trouvant à l'intérieure de sa zone locale, ainsi que les autres membres adhérents à l'arbre multicast (à travers les messages d'adhésion à l'arbre multicast précédents) ; sachant, bien sûr, qu'un nœud donné peut participer à plusieurs groupes multicast en même temps.

Le but de maintenir une telle structure est de joindre rapidement un groupe multicast donné s'il existe au moins un seul membre de l'arbre multicast correspondant dans sa zone locale. Car sinon, le nœud procède à une autre alternative de jointure à l'arbre multicast, en se basant sur les MPR-diffusions d'une certaine requête de recherche de route (***MRREQ***), à l'extérieure de sa zone locale.

4.5.3 Le protocole MIERP-MPR

Ce protocole se compose essentiellement des opérations de recherche, de réponse et d'activation des routes. Ici, un nœud qui désire joindre un groupe multicast et qui ne trouve pas de chemin valide directement depuis sa zone, il procède à la recherche d'une route valide à l'extérieure de sa zone. Pour cela, le nœud doit construire son arbre bordercast couvrant tous ses noeuds périphériques non parcourus par cette même requête, en respectant ici le principe de la technique MPR, et qui consiste à MPR-diffuser les messages qui ont été diffusés par inondation dans MIERP. La construction de l'arbre bordercast fait appelle donc au protocole BRP (voir la section 4.2.3).

Le nœud source (celui qui désire envoyer la requête) envoie sur son arbre bordercast la requête ***MRREQ*** vers ses nœuds périphériques. Ces derniers recommencent la même opération de recherche interne, puis externe s'ils ne trouvent pas de chemins valides depuis leurs zones.

Si, à un moment donné, un noeud (un nœud périphérique d'une certaine zone) trouve une route valide vers l'arbre multicast, il renvoie une réponse ***MRREP*** sur le chemin inverse accumulé par les nœuds intermédiaires dans la requête envoyée. Cette opération est réalisée par une certaines procédure d'accumulation des chemins [131].

Le nœud qui a renvoyé la réponse vers la source attend, un certain délai, la réception d'un message d'activation de route ***MRACT*** que doit envoyer cette source. Mais, comme cette dernière peut recevoir plusieurs réponses depuis différents nœuds et sur différents chemins, elle doit choisir une seule réponse parmi plusieurs, en se basant dans son choix sur certaines métriques concernant les nouds et les routes. Les autres répondeurs de cette requête attendent, pour un délai limité, la réception d'un message d'activation avant de confirmer que leurs réponses ont été refusées. Ils passent ensuite à l'élagage des chemins correspondants depuis leurs tables de routage.

Si la source ne reçoit aucune réponse pendant une durée bien limitée, elle confirme qu'il n'y a plus de membres du groupe en question, et qu'elle est le premier membre participant à ce groupe. De ce fait, elle élit soi même en tant que leader de ce groupe multicast et commence directement à MPR-diffuser périodiquement de ***ses messages de leader du groupe***.

4.5.4 La maintenance de l'arbre multicast

4.5.4.1 La procédure d'élagage

Dans cette procédure, j'applique le même principe du protocole de base MZRP, et ce lorsque un membre du groupe multicast décide de quitter son groupe (voir la section 4.3.4.1).

Ici, je traite le cas où le leader du groupe multicast décide de quitter son groupe. Pour cela, le leader envoie vers l'un de ses successeurs un message d'élagage ***MPRUNE***. Si le successeur en question est un nœud d'acheminement dans cet arbre multicast, il élague soi même aussi de cet arbre et envoie le message d'élagage vers l'un de ses successeurs. Cette opération d'élagage s'arrête lorsque le message d'élagage soit reçu par un nœud membre du groupe multicast sur cet arbre, qui devient par la suite le leader de ce groupe multicast. Par conséquent, il commence à MPR-diffuser périodiquement ses ***messages de leader du groupe***.

4.5.4.2 La réparation des cassures

Certains mécanismes de détection de voisins servent à contrôler les branches de l'arbre multicast. Si un nœud ne reçoit plus de messages depuis son voisin, et après un délai d'attente borné, il constate qu'une cassure du lien a eu lieu avec son voisin. Par conséquent, une réparation de ce lien doit être effectuée et c'est le nœud successeur de cette branche qui doit l'effectuer. Si ce nœud successeur possède une route valide vers l'arbre multicast directement depuis sa zone de routage local, et que le nombre de sauts séparant ce membre de leader est inférieure à celui qui sépare le nœud successeur de leader ; alors, une réparation locale est immédiatement effectuée par l'envoie unicast de la requête ***MRREQ*** vers ce membre de l'arbre multicast.

Dans le cas contraire, où il n'existe aucun chemin valide vers l'arbre depuis la zone de routage local, le nœud successeur MPR-diffuse cette requête ***MRREQ*** dans la totalité du réseau et attend une réponse. Une fois recevoir cette requête, les membres de l'arbre multicast dont le nombre de sauts qui les séparent de leader du groupe est inférieure à celui qui sépare le successeur de leader lui même, peuvent répondre sur cette requête par les messages ***MRREP***. Dans le cas où le successeur ne reçoit aucune réponse sur sa requête durant une certaine période, il refait cette même opération de recherche un nombre limité de fois. S'il ne reçoit toujours plus de réponses, il confirme que le réseau a été partitionné, par conséquent, un nouveau leader du groupe doit être élu de la même manière précédente (voir la section 4.5.4.1).

4.5.4.3 La reconnexion des partitions

Ce sont les messages de leader du groupe, circulant périodiquement dans le réseau, qui servent à détecter la reconnexion des différentes partitions du réseau. Ici, je procède de la même manière décrite dans la section 4.3.4.3 sauf que le leader du groupe de plus petite adresse IP n'inonde pas le réseau avec les messages de leader du groupe, mais il MPR-Diffuse ses ***messages de leader du groupe***, en mettant à jour certains flags de ces messages, pour annoncer le changement de leader du groupe.

4.5.4.4 La transmission des paquets de données sur un tunnel IP

La transmission des paquets de données sur un tunnel IP s'effectue exactement de la même manière décrite dans la section 4.3.5, et sans aucun changement.

4.6 Conclusion

Dans ce chapitre j'ai décrit tous les pas nécessaires pour la compréhension de nouveau protocole proposé MZRP-MPR. J'ai commencé tout d'abord par la version unicast ZRP, où les composants de ce protocole qui ont servi dans toutes les versions qui viennent après, ont été présentés en détail.

Une extension multicast de ce protocole à été présenté en deuxième lieu, il s'agit de protocole multicast MZRP. MZRP est le protocole multicast que j'ai choisit pour ma nouvelle proposition MZRP-MPR. Il est aussi décrit vers la fin de ce chapitre.

Le protocole MZRP-MPR fonctionne exactement comme le protocole MZRP, sauf que, dans ce premier j'ai introduit une certaine technique de diffusion optimisée qu'est la technique des relais multipoint (ou MPR). L'introduction de cette technique dans le protocole MZRP vise à minimiser la charge de contrôle et de réduire aussi les délais de retransmissions effectuées par les nœuds relayeurs.

Le chapitre suivant se compose de deux parties essentielles ; la première partie est consacrée à l'évaluation du la technique MPR à travers une simulation effectuée dans [96]. La deuxième partie présente une analyse qualitative de nouveau protocole MZRP-MPR en se basant sur les résultats de simulation de la technique MPR.

Chapitre 5

Analyse du protocole MZRP-MPR

5.1 Performance de la technique MPR vs inondation

Dans ce sous chapitre, je vais présenté les techniques de diffusion dans les réseaux ad hoc sans fil. Pour diffuser les messages de contrôle dans le réseau, les protocoles réactifs tels que AODV, DSR et ODMRP utilisent l'inondation pure, ce qui résulte en une consommation de bande passante considérable et une perturbation de tous les nœuds, qui peut être très répétitive. Une évaluation de performances de la diffusion par les relais multipoint sera décrite par la suite. Ces performances seront comparées à celles de la diffusion par inondation. Cette étude a été effectuée dans le cas des réseaux larges et denses (1024 nœuds) en considérant le fading des liens entre les voisins [96] [129].

5.1.1 Les techniques de diffusion dans les réseaux

Plusieurs solutions ont été proposées dans la littérature pour gérer et optimiser la diffusion dans les réseaux sans fil. Les broadcast dans les réseaux sans fil sont très différents du cas des réseaux filaires où un routeur connaît généralement les routeurs qui lui sont reliés. Une transmission radio est une diffusion par nature, et tous les voisins à la portée sont des récepteurs potentiels ; quant aux nœuds hors portée, ils ont besoin d'un ou plusieurs relayages, pour recevoir cette information. Comment faut-il procéder, et comment faire ceci le plus efficacement possible (i.e. avec un minimum de consommation de bande passante radio) ?

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

- La solution la plus classique consiste en une inondation totale. Chaque participant de réseau, répète le message reçu s'il le reçoit pour la première fois. Cette méthode est la plus facile à implémenter et ne nécessite aucun trafic de contrôle supplémentaire. Sont inconvénient est qu'elle utilise excessivement la bande passante et provoque des tempêtes d'émissions perturbant ainsi le fonctionnement du réseau.

- Une deuxième technique consiste à diviser le réseau en clusters, avec un chef pour chacun d'entre eux. Ce chef doit avoir connaissance de chaque membre de son groupe. La diffusion s'effectue via les chefs des clusters de proche en proche. Cette méthode pose le problème de la division du réseau en clusters et l'élection du chef. Les membres

d'un même cluster doivent être d'accord sur la méthode d'élection, ceci est délicat et peut mener à des situations d'instabilité à cause de la nature changeante de la topologie du réseau ad hoc.
[132] [133] [134] ont étudié le problème de choix des chefs de clusters et défini des méthodes afin de converger rapidement à une situation stable. Cette méthode est adoptée par le protocole CBRP [51] par exemple.

- Construction d'un dominating set : cette famille de technique se propose de construire une sorte de backbone dans le réseau ou une sorte de grille permettant d'atteindre tous les nœuds à un saut. Le routage s'effectue par l'intermédiaire de cet ensemble. Voir [135] qui décrit ce genre de méthode et la formation distribuée de tels ensembles.

- Algorithmes se basant sur la localisation par GPS. Par exemple, un nœud N réémet le message reçu si et seulement si, la distance le séparant de son voisin qui a émis le message est supérieure à une certaine valeur D [136].

- Méthodes centralisées : il s'agit de la construction d'arbre multicast centralisé autour d'un noyau reliant tout le réseau par un arbre couvrant. Tout message à diffuser est envoyé vers le noyau qui le distribue à son tour.

- La technique des relais multipoint décrite précédemment (voir le sous chapitre 4.4) permet de réduire le nombre des relayeurs. En effet, c'est à chaque nœud de définir son ensemble de relayeurs qui lui permet d'acheminer les informations vers ses voisins à deux sauts. Ce processus est récursif et permet de cette façon de couvrir tout le réseau.

[137] décrit plus en détail les techniques de diffusion décrites ci-dessus. Dans la suite, je vais présenter une comparaison de performances de la technique des relais multipoint par rapport à la technique de référence : l'inondation.

5.1.2 Simulation de technique des relais multipoint

Ici je vais présenté une simulation de la technique des relais multipoint réalisée par le docteur A.LAOUITI dans sa thèse de doctorat [96]. L'auteur a simuler deux algorithmes de diffusions dans les réseaux sans fil. Cette simulation vise à évaluer le comportement de l'algorithme des relais multipoint par rapport à la technique d'inondation. En considérant des erreurs de réception au niveau des nœuds récepteurs. On va regarder l'impact de ces erreurs sur tout le réseau. Cette simulation va permettre aussi de voir à quel point, l'algorithme des relais multipoint est capable d'assurer la diffusion et garantir de bons résultats.

Ces résultats ont été confrontés aux résultats d'une deuxième simulation qui va porter sur une diffusion elle aussi, mais, en utilisant le protocole classique de l'inondation ; l'inondation présente une meilleure fiabilité et robustesse, et consomme une large bande passante en contre partie.

Cette étude porte sur un grand réseau en terme de nombre de nœuds. Les nœuds ont un nombre de liens important avec leur voisinage. Afin de garantir un chemin d'un nœud quelconque vers un autre, un graphe connexe est considéré.

La simulation consiste à faire varier la probabilité d'erreur de réception P, allant de 0 à 1 par pas de 1/100. Pour chacune des valeurs, la diffusion d'un message d'un nœud vers le reste du réseau est appliquée. Cette étape est répétée pour tous les nœuds du graphe dans le but de calculer une moyenne pour chaque valeur de probabilité d'erreur de réception.

Pour cette simulation, certaines hypothèses ont été introduit [96], dans le but de mieux cadrer le champ de cette étude sur le problème de l'impact de l'erreur de réception au cours du déroulement de la diffusion.

- **Hypothèse 1 :**

Les mécanismes de réémission ne sont pas considérés lorsqu'il y a une erreur de réception (synchronisation, ...). Il s'agit d'un broadcast.

- **Hypothèse 2 :**

On suppose que les liens sont parfaitement symétriques entre chaque paire de nœuds du réseau, la matrice de topologie résultante est une matrice composée de 0 et 1 parfaitement symétrique.

- **Hypothèse 3 :**

A chaque fois qu'un nœud émet un paquet, ses voisins à un saut le reçoivent avec une probabilité p comprise entre o et P. P étant une variable de simulation qui varie entre 0 et 1 par pas de 1/100.

- **Hypothèse 4 :**

Pas de routage à considérer dans cette étude. On ne s'intéresse qu'au cas de la diffusion.

- **Hypothèse 5 :**

Il n'y a pas de trafic autre que le paquet à diffuser à chaque fois.

- **Hypothèse 6 :**

Un nœud ne répète un paquet (dans le cas où il est relayeur d'un nœud voisin) que s'il le reçoit pour la première fois. Sinon, il l'ignore.

- **Hypothèse 7 :**

Tout nœud est candidat potentiel pour être un relayeur d'un nœud voisin.

5.1.2.1 Topologie

Dans cette étude [96], l'auteur s'est focalisé sur des réseaux larges et très denses. La topologie utilisée est constituée de 1024 nœuds disposés sur une grille. Chaque ligne de la grille est composée de 32 nœuds posés sur 32 colonnes. Chaque nœud peut disposer de 48 voisins au maximum. Les voisins sont choisis aléatoirement dans un rayon de trois sauts. La probabilité d'être un voisin diminue avec la distance. Chaque nœud aura en général un ensemble de voisins compris entre 24 et 40.

5.1.2.2 Accès au canal

Le but de cette simulation est d'étudier l'effet des erreurs de réception dues à d'autres phénomènes que les collisions. Pour ne pas avoir de collisions, l'auteur a utilisé un modèle de blocage à deux niveaux ; i.e pour qu'un nœud puisse émettre, il ne faut pas que l'un de ses voisins à deux niveaux ait quelque chose à diffuser. Si le cas se présente, alors, l'une des émissions est bloquée. Ceci a été utilisé dans le but d'éliminer le problème d'interférence lorsqu'un nœud reçoit deux transmissions radio au même moment. Un blocage de deux niveaux est suffisant pour assurer cela, car récursivement, ceci est généralisé sur tout le réseau [96]. Comme le montre la figure 5.1, bien que le mécanisme de contrôle d'accès au canal permet aux deux nœuds, A et B, d'émettre en même temps, ceci génère une interférence au niveau du nœud C.

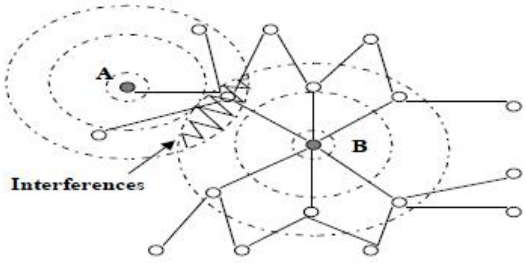


FIG 5.1*– Interférences des émissions de A et B* [96]

5.1.2.3 Les erreurs de réception

Le but de cette simulation est de prendre en compte l'erreur de réception lors des diffusions. La modélisation de l'erreur revient à faire un tirage au sort lors d'une émission d'un nœud vers ses voisins à un saut. On tire donc un nombre p, compris entre 0 et 1, si P < p, alors la réception est réussit ; sinon, ce voisin ne reçoit rien (voir la

figure 5.2). rappelons que, P est un paramètre de simulation qui varie entre 0 et 1 par pas de 1/100.

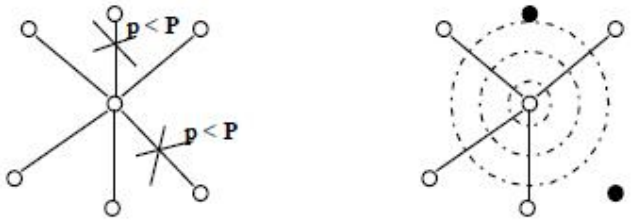


***FIG 5.2**– Les voisins ne reçoivent pas la diffusion* [96]

5.1.2.4 Résultats de simulation

Plusieurs simulations ont été effectuées par l'auteur [96], pour présenter, comparer et enfin évaluer les performances de la technique MPR à ceux de la diffusion par inondation. Voici alors les courbes résultantes de la simulation des deux techniques.

- **Date de fin d'activité dans le réseau**

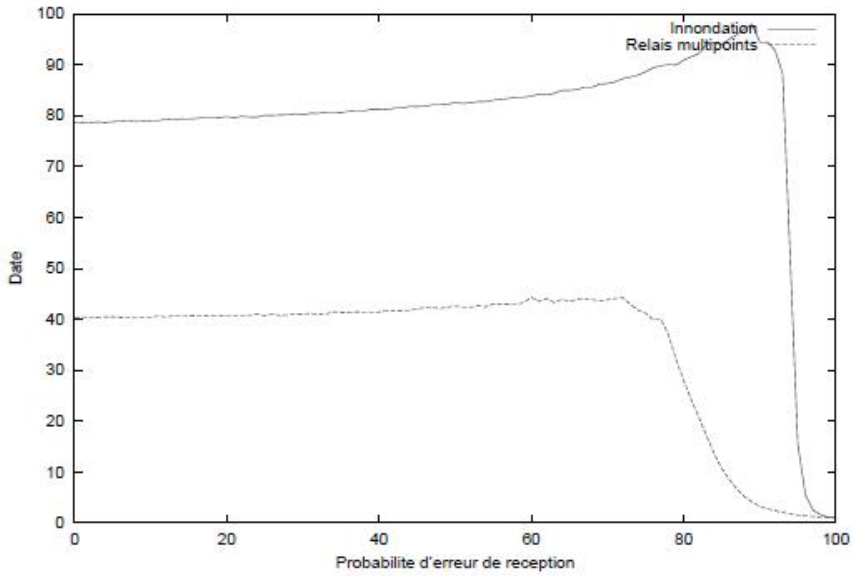


***FIG 5.3**– Date de fin d'activité dans le réseau* [96]

Les courbes de la figure 5.3 montrent que, dans le cas de l'inondation (la courbe en ligne continue), cette activité augmente parallèlement avec la probabilité d'erreur de réception. Elle s'accélère après la valeur de 80% pour atteindre son pic à 90% puis, elle

chute brusquement pour arriver à une valeur de 1 correspondant à la valeur de 100% de probabilité d'erreur de réception.
Dans le cas de la technique des relais multipoint (ligne en pointillé), cette activité garde une valeur quasi constante lorsque la probabilité d'erreur de réception augmente, puis, elle chute après les 80% pour atteindre une valeur de 1 pour 100% de probabilité d'erreur de réception.

On conclut donc qu'en utilisant la techniques des relais multipoints, la diffusion se termine beaucoup plus rapidement dans le réseau. Ceci nous fait gagner de la bande passante radio en plus de gain de temps. La chute de la courbe est beaucoup plus raide dans le cas de l'inondation.

- **Nombre de messages reçus par nœud**

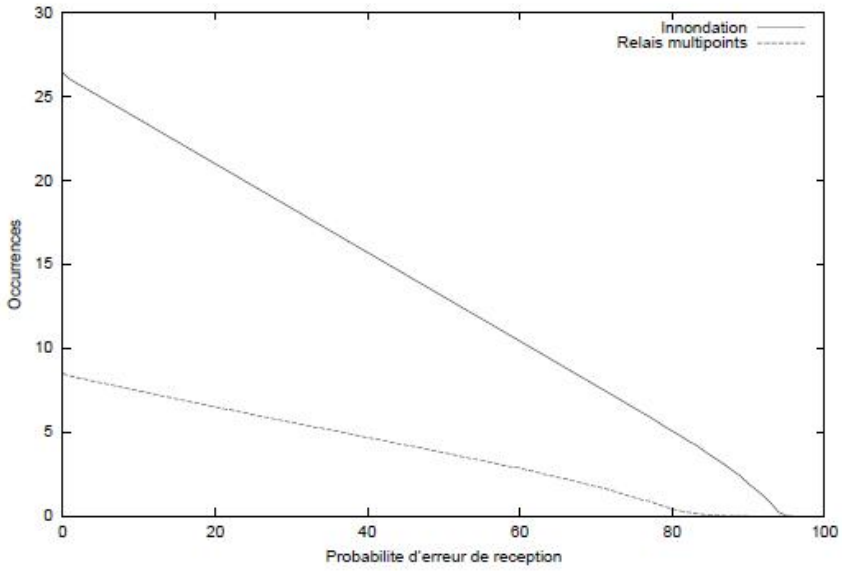


***FIG 5.4** –Nombre de messages reçus par nœud* [96]

Dans le cas de l'inondation, les nœuds reçoivent beaucoup d'occurrence du même paquet. Ce nombre est beaucoup plus faible dans le cas des relais multipoint. Pour une probabilité d'erreur égale à 0, on a un rapport de trois entre les deux cas de figure. Evidement, ce nombre diminue avec l'augmentation de la probabilité d'échec de réception.

- **Nombre de retransmissions dans le réseau**

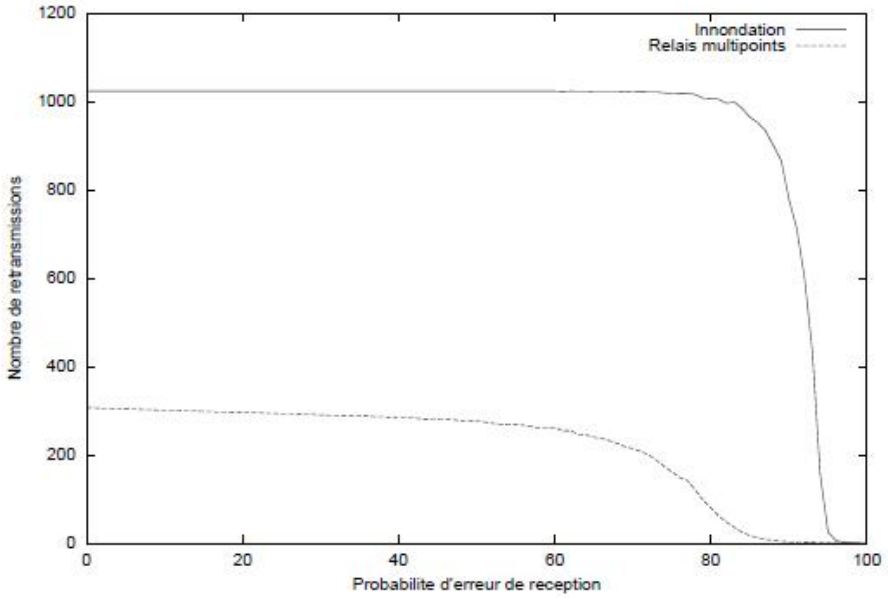


***FIG 5.5** –Nombre de retransmissions dans le réseau* [96]

Ici, le nombre de retransmissions est divisé par trois pour la diffusion utilisant les relais multipoint par rapport à celle de l'inondation. Ceci est très intéressant de fait que cela nous fait gagner des retransmissions inutiles avec un résultat satisfaisant pour l'ensemble du réseau.

- **Nombre de nœuds qui reçoivent la diffusion**

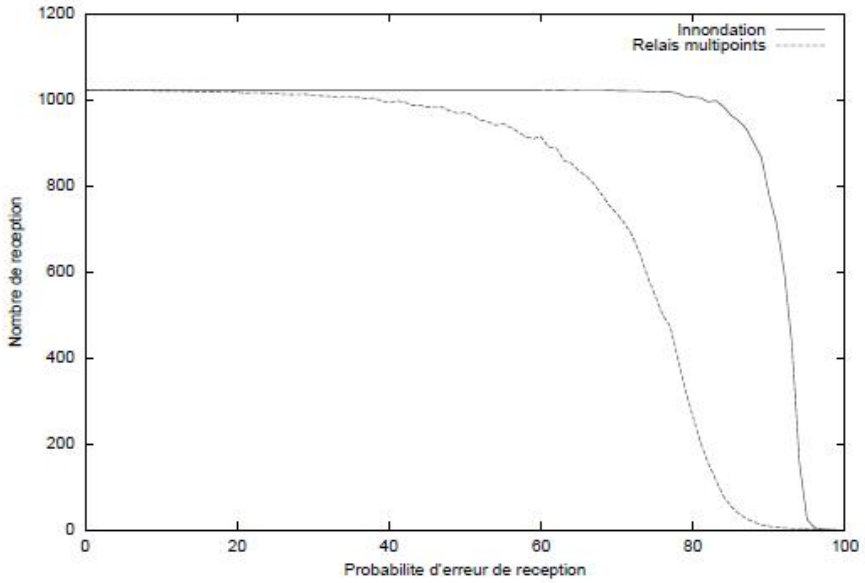


***FIG 5.6** –Nombre de nœuds qui reçoivent la diffusion* [96]

La figure 5.6 montre que la technique des relais multipoint entraine des pertes sensibles dès que, l'on dépasse une probabilité d'erreur de réception de 40%. Dans la théorie, la probabilité d'échec de réception acceptable est d'au plus de 10% [96]. Par conséquent on peut affirmer que cette technique nous fournit de très bons résultats dans le cas des réseaux denses.

- **Date de la première réception par le dernier nœud**

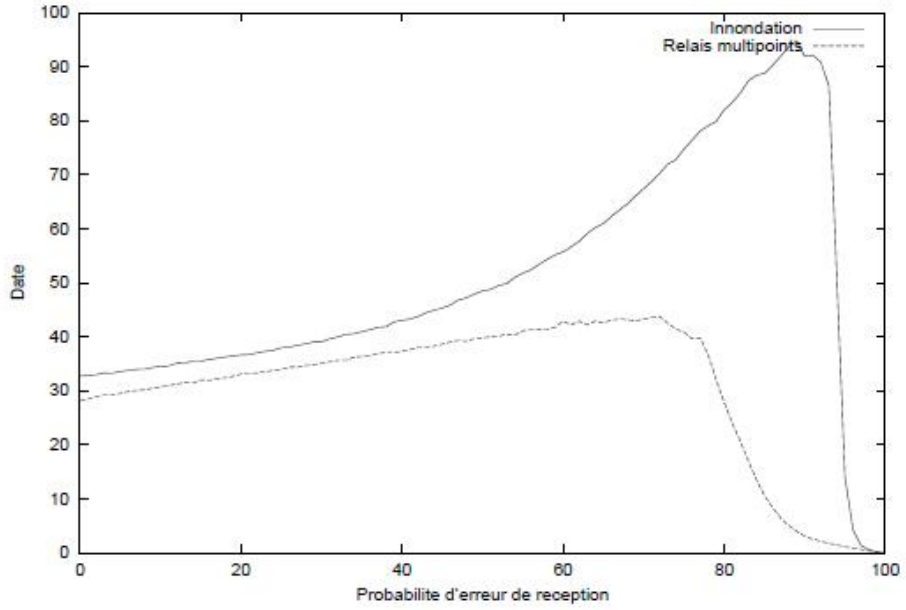


***FIG 5.7** – Date de la dernière première réception* [96]

Cette figure 5.4, montre que le grand pic dans le cas de l'inondation s'explique par le fait qu'avec cette probabilité, d'échec il y a encore des nœuds qui reçoivent le message diffusé pour la première fois avec un peu de retard . ce qui n'est pas le cas pour la technique des relais multipoint, qui avec cette probabilité d'échec, ne réussit pas à couvrir la majorité du réseau.

- **Nombre de réceptions et de retransmissions dans le réseau**

Les deux courbes de la figure 5.8 en trait continu confondues retracent le nombre de nœuds qui ont reçu le paquet diffusé et le nombre de retransmissions dans le réseau en fonction de la probabilité d'erreur de réception. Le fait que les deux courbes soient confondues est très normale, car avec la technique de l'inondation, chaque fois qu'un nœud reçoit un paquet il le diffuse à son tour une seule fois. On souligne ici le très bon comportement de cette technique vu que le nombre de réception total ne se dégrade qu'après un taux d'erreur de réception supérieur à 80%. Ceci confirme bien la bonne maintenabilité de cette technique.

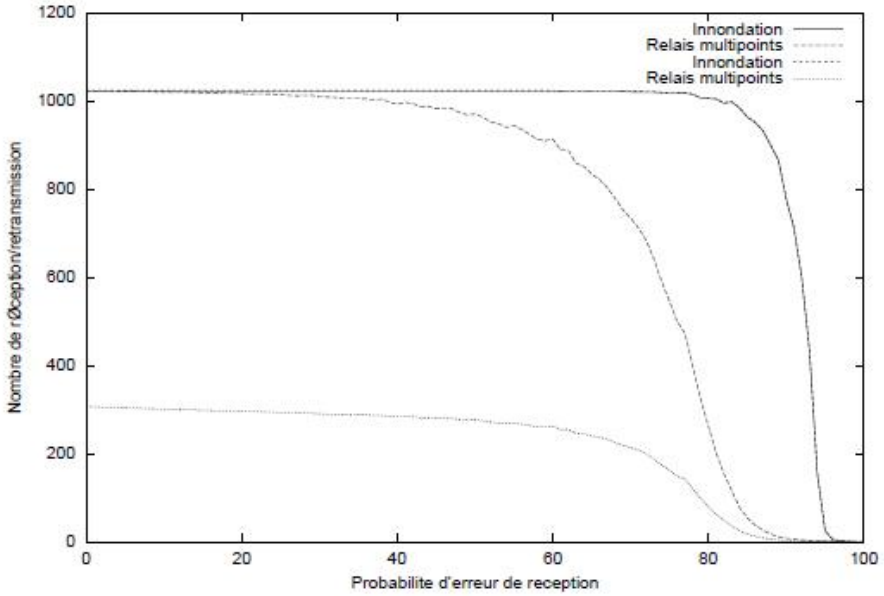


FIG 5.8 *– Nombre de réceptions et retransmissions dans le réseau* [96]

- **Date de fin d'activité et la première réception par le dernier nœud**

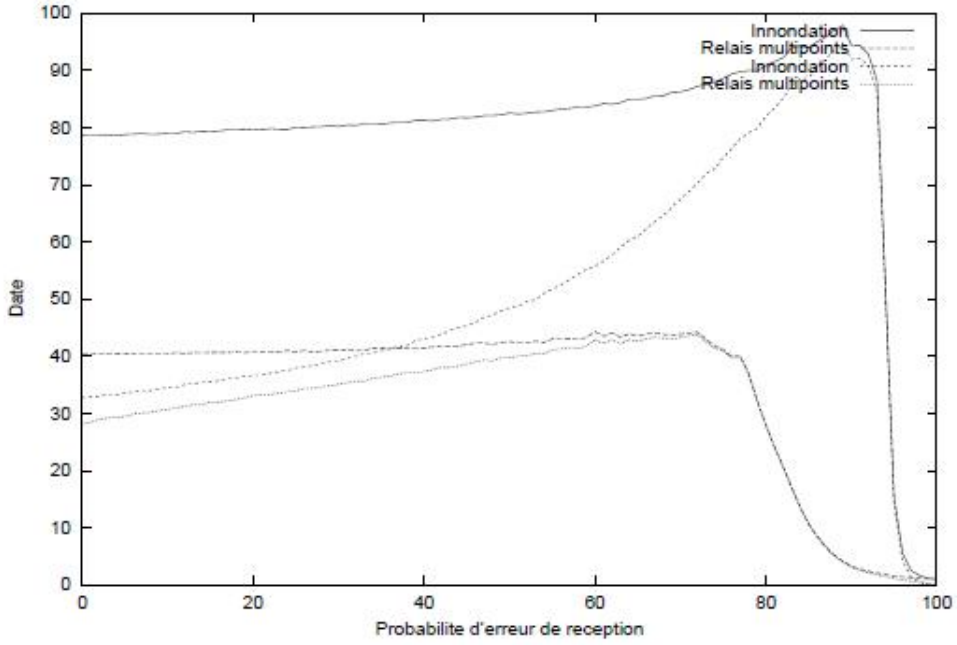


FIG 5.9 *–Date de fin d'activité et la dernière première réception dans le réseau* [96]

Pour atteindre les quelques 1024 nœuds, on a eu besoin seulement de 300 retransmissions selon cette technique. La courbe en ligne continue supérieure (figure 5.9), représente la date de la première réception par le dernier nœud et la courbe en pointillés supérieure représente la date de la fin d'activité dans tout le réseau dans le cas de l'inondation.

D'après les résultats de cette simulation réalisée dans [96], l'auteur a peut conclure que la technique des relais multipoint améliore légèrement le temps nécessaire pour que la diffusion d'un message atteigne au moins une fois tous les nœuds. Il a constaté aussi que la diffusion se termine plus rapidement en utilisant cette technique de diffusion optimisée. Ceci est dû au nombre plus réduit de nœuds relayeurs. Les nœuds sont moins submergés par les copies multiples du même message, ce qui fait, que l'on ne gaspille plus la bande passante. Notons que dans le cas de l'inondation, le nombre de réceptions est égale au nombre de retransmissions.

Par contre, la technique des relais multipoint n'est pas fiable et donne des résultats médiocre lorsque la probabilité d'erreur de réception augmente et dépasse les 50%. Dans cette zone là, la technique de l'inondation se comporte bien et donne une meilleure couverture du réseau avec une consommation de bande passante plus importante.

5.2 Analyse qualitative du protocole multicast MZRP-MPR

MZRP-MPR hérite tout le fonctionnement du protocole MZRP et conserve la plus part de ses métriques. MZRP-MPR construit et maintient un arbre partagé pour chaque groupe multicast. MZRP-MPR ne produit pas de charge suplémentaire dans les réseaux à grande mobilité, car ici je ne propose pas l'ajout d'un paquet de contrôle ou d'une structure d'enregistrement supplémentaire, ni même la construction d'un autre l'arbre multicast lors de détection et réparation des liens cassés. MZRP-MPR est aussi un protocole qui construit son arbre multicast à la demande et qui conserve la propriété d'utilisation de tunnel IP temporaires pour la distribution des paquets de données en cas de cassure des liens.

5.2.1 Les caractéristiques de MZRP-MPR

Les principales caractéristiques de ce nouveau MZRP-MPR protocole sont :

1. La charge de routage est réduite à l'intérieure et à l'extérieur des zones du réseau
2. Les messages de contrôle restent les mêmes que ceux de MZRP.
3. La construction de l'arbre multicast suit les mêmes étapes que celles de MZRP.
4. Maintenance et utilisation de l'arbre sont optimisées à travers la technique MPR.

MZRP-MPR utilise la technique MPR pour réduire la charge de routage, et minimiser le temps des retransmissions des différents paquets de contrôle. Tout paquet de contrôle qui a été diffusé par inondation dans le protocole MZRP sera diffusé par la technique des relais multipoints dans MZRP-MPR.

5.2.2 Les différences entre MZRP et MZRP-MPR

Les différences principales entre les deux protocoles sont :

1. Dans MZRP, si je suppose le cas de plusieurs nouveaux groupes multicast, le premier membre de chacun de ces groupes commence à diffuser périodiquement ***les messages de leader du groupe*** dans la totalité du réseau. Imaginant donc le grand nombre de messages des leader qui vont submerger le réseau périodiquement.
   MZRP-MPR utilise dans ce cas la diffusion MPR dans le but de réduire la charge de ces messages périodiques, en minimisant le nombre de répéteurs de ces messges.

2. Dans MZRP, lorsqu'un nœud joint un groupe multicast, il diffuse à l'intérieur de sa zone locale ***les messages d'adhésion*** à l'arbre multicast. Ici, lorsque le rayon de la zone est grand, cette diffusion entraine une charge de routage remarquable à l'intérieure de la zone en question.
   MZRP-MPR, par contre, minimise cette charge en MPR-diffusant ces messages d'adhésion quelques soit la taille de la zone ; il bénéficie ici cela des informations fournies par le protocole MIARP (informations des voisins locaux).

3. Lorsqu'un nœud désire joindre un groupe multicast qui n'existe pas, il élit soit même en tant que leader de ce groupe ; et ceci après des tentatives d'envoie de requêtes **MRREQ** sans recevoir aucune réponse. Dans MZRP, ce nœud qui devient leader commence par les diffusion périodiques ***des messages de leader de groupe***. Alors que dans MZRP, ces mêmes messages sont MPR-diffusées périodiquement dans le réseau.

4. Lorsque le leader d'un groupe multicast décide de quitter son groupe, il commence par envoyer un message d'élagage MPRUNE vers son successeur ; si ce successeur n'est pas un membre de groupe multicast, il refait cette même opération avec son successeur aussi. Et ainsi de suite jusqu'à ce que le message d'élagage arrive à un nœud membre du groupe multicast, qui devient par défaut chef du groupe multicast. Dans MZRP, ce nouveau chef inonde périodiquement le réseau par ses ***messages de leader du groupe***. Dans MZRP-MPR, par contre, ces messages sont MPR-diffusés, minimisant ainsi la consommation de la bande passante.

5. La mobilité des nœuds peut entrainer l'apparition ou la disparition des nœuds dans des zones différentes de leurs zones initiales. Cette mobilité peut causer des cassures de liens pouvant être détectées grâce à certains mécanismes de détection de voisins. Pour réparer ces liens cassés, un nœud commence par chercher un membre de l'arbre multicast à l'intérieur de sa zone locale , avant de passer à sa recherche externe. Le MZRP diffuse dans ce denier cas la requête **MRREQ** dans la totalité du réseau, alors que dans MZRP-MPR, cette même requête est MPR-diffusée sans inonder et surcharger le réseau à chaque détection de cassure.

6. A travers l'envoie périodique ***des messages de leader*** des différents groupes multicast, je peux détecter et reconnecter les différentes partitions du réseaux. Dans MZRP, lorsque deux ou plusieurs leader diffusent les messages de leader d'un même groupe multicast, le leader possédant la plus petite adresse IP (le plus ancien) inonde le réseau avec les messages de leader du groupe, en ignorant celle de leader possédant la grande adresse IP.
   MZRP-MPR applique le même principe sauf qu'il MPR-diffuse périodiquement ***les messages de leader*** de plus petite adresse IP, au lieu de les diffuser par inondation.

7. MZRP-MPR évite les liens unidirectionnels se trouvant sur les routes d'acheminement lorsque les différents nœuds répondent par MRREP, alors que MZRP ne résout pas ce genre de problème.

5.2.3 Le coût de protocole MZRP-MPR

Dans MZRP, les leaders des différents groupes multicast diffusent périodiquement ***les messages de leader de groupe***, ce qui représente une charge supplémentaire à ce protocole. Pensant juste à l'occupation de canal et au temps durant lequel toutes les transmissions sont effectuées. MZRP-MPR optimise l'utilisation de la bande passante et le temps nécessaire pour les différentes diffusions périodiques à travers sa technique MPR ; cette dernière à été étudier et simuler [96] dans le but de prouver son importance et son rôle d'optimisation remarquable dans les réseaux sans fil.

Dans mon analyse, je suppose qu'il y a M groupes multicast dans un réseau constitué de N nœuds. Par conséquent, le nombre de leader de groupes multicast est [(m> =M) <= N], car un même groupe multicast peut avoir, à moment donné, plus d'un seul leader dans le cas où le groupe est partitionné . Lorsque ces différents leaders de groupe multicast MPR-diffusent leurs ***messages de leader de groupe*** dans la totalité du réseau, chaque MPR-diffusion à deux sauts utilise un nombre de relayeurs r plus réduit par rapport à la diffusion totale qui diffuse ses messages vers tous ses relayeurs (R >=r). Par conséquent, le nombre de transmissions t des messages de leader de groupe avec MZRP- MPR est réduit dès la première MPR-diffusion. Alors que dans le MZRP, le nombre de transmissions T est supérieur ou égale à t à cause d'utilisation de la technique d'inondation (voir les figures 4.9 et 4.10).

Si par exemple le nœud F est élu pour la première fois comme étant chef ou leader de groupe multicast, il commence par envoyer périodiquement ses messages de leader de groupe dans la totalité du réseau. Dans MZRP, il diffuse ces messages par inondation (voir la figure 5.10) .

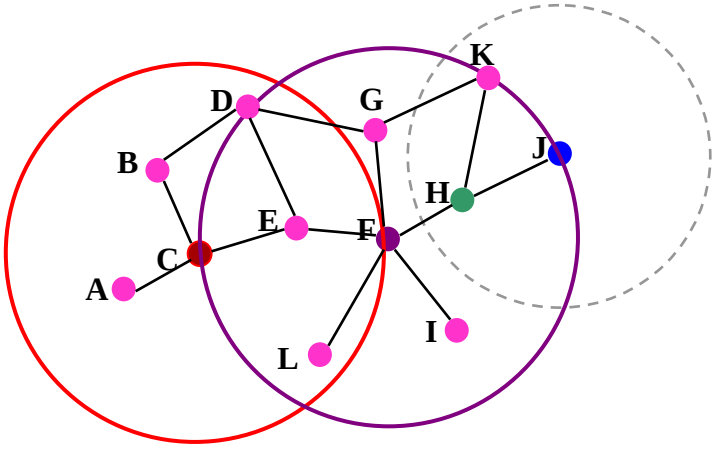


FIG 5.10 *– Diffusion par inondation des messages de leader de groupe*

Cette figure montre bien que la diffusion par inondation engendre 5 transmissions dans le premier niveau (voisinage locale de F), 6 transmissions dans le second et 3 transmissions à l'extérieure de la zone de F. Le totale donne 14 transmissions pour diffuser une messages de leader de groupe dans une telle topologie.

Lorsque j'utilise la technique MPR proposée dans MZRP-MPR, le nombre de transmission est réduit comme le montre la figure 5.11. Ici, le nombre de transmissions dans le premier niveau est 2, dans le second est 4, et à l'extérieure de la zone est 2. Le nombre totale de transmissions est 8.

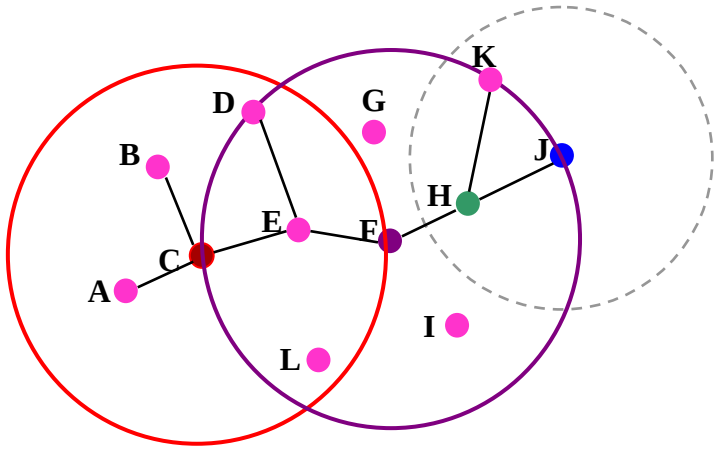


FIG 5.11 *– Diffusion optimisée (MPR) des messages de leader de groupe*

Dans cet exemple, le nombre de transmissions dans la diffusion par inondation est réduit d'un rapport de (8/14) càd de 57% en utilisant la technique MPR. Imaginant donc le cas des diffusions périodiques dans un réseau dense et possédant plusieurs groupes multicast.

Si je suppose maintenant que la longueur de chaque message de leader du groupe est égale à L alors le nombre de transmissions totale TT d'un message de leader de groupe est dans le MZRP (TT=T*L), et dans MZRP-MPR (TT=t*L). Comme t<=T ; alors t*L <= T*L ; par conséquent le nombre de transmissions d'un message de leader de groupe multicast peut être réduit avec l'utilisation de la techniques MPR. Je peux dire aussi que si tous les leaders des groupes multicast diffusent leurs messages de leader de groupe en même temps, le nombre de transmission globale GT est, [GT (MZRP)=M*T*L] > = [GT(MZRP-MPR)=M*t*L].

Si je considère le rapport de réduction précédent (57%), le nombre de transmissions globale en utilisant la diffusion par inondation (GT (MZRP)=M*T*L) = (GT(MZRP-MPR)=M*t*L)*0,57.

Cette opération de MPR-diffusion périodique ***des messages de leader de groupe*** multicast est exécutée dans plusieurs cas :

- Auto-élection de leader d'un nouveau groupe multicast ;
- Détection d'un membre de groupe multicast sur l'arbre multicast, qui devient leader de ce groupe, et ce après l'exécution de la procédure d'élagage de l'ancien leader de ce groupe.
- Détection d'une autre partition d'un même groupe multicast possédant son propre leader multicast. Le leader d'adresse IP minimale commence les MPR-diffusion périodiques des messages de leader.

Si je considère maintenant le cas de cassure des branches, MZRP-MPR essaie de réparer ceci en cherchant un membre de l'arbre multicast interne à sa zone locale. S'il ne réussit pas, il MPR-diffuse la requête **MRREQ** pour la recherche externe de ce membre. Cette MPR-diffusion n'est pas périodique comme celle des messages de leader de groupe, mais peut arriver fréquemment à cause de la mobilité des nœuds. Donc, l'application de la technique MPR dans ce cas améliore aussi les performance de ce nouveau protocole.

Revenant maintenant au cas des messages d'adhésion qui sont MPR-diffusés à l'intérieure des zones locales. Lorsqu'un nœud joint un groupe multicast, il MPR-diffuse à l'intérieur de sa zone locale ***les messages d'adhésion*** à l'arbre multicast l'importance de cette technique MPR réside ici dans le cas où la taille de la zone est grande. La diffusion par inondation va engendrer une charge de routage remarquable à l'intérieure de la zone par rapport à la diffusion optimisée par MPR.

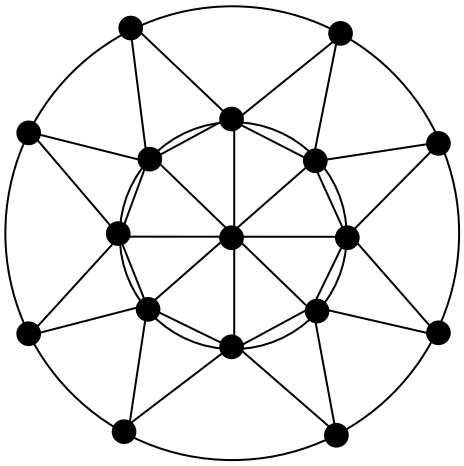


FIG 5.12 *– Diffusion des messages d'adhésion à l'arbre multicast*

Dans cette figure 5.12, la diffusion d'un message d'adhésion à l'arbre multicast à l'intérieure d'une zone locale génère 32 transmissions. Dans la figure 5.13, par contre, la MPR-diffusion d'un tel message à l'intérieure de sa zone locale va générer 12 transmissions ; et qui représente exactement le (3/8 = 0,37%) par rapport à la diffusion.

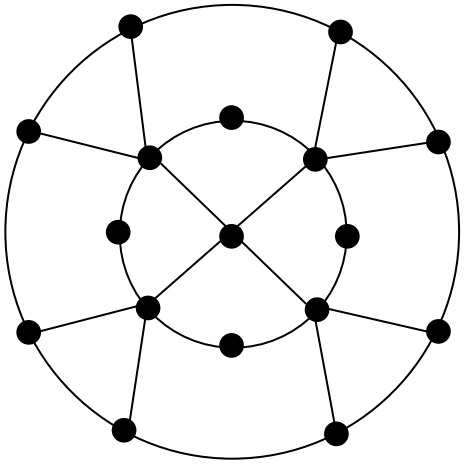


***FIG 5.13** – MPR-Diffusion des messages d'adhésion à l'arbre multicast*

La réduction de nombre de transmissions que garantit MZRP-MPR engendre automatiquement la réduction de temps de transmission. Par conséquent, la bande passante est moins consommée et est libérée rapidement.

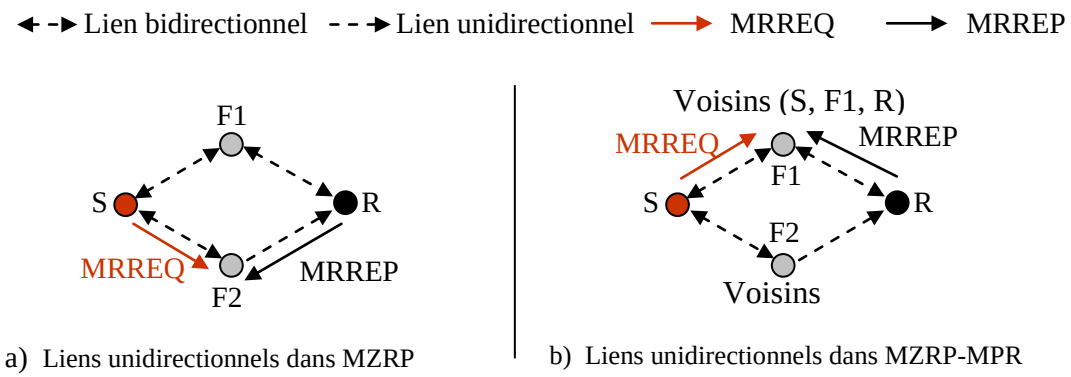
5.2.4 La redondance des messages dans MZRP-MPR

Le but de la MPR-diffusion est de minimiser la charge qu'engendre la diffusion par inondation. A cause de la non fiabilité des communications sans fil, spécialement dans les environnements à grande mobilité, ou de la lourde charge ; j'ai besoin de la redondance au niveau de certains messages permettant de construire et de maintenir l'arbre multicast. Pour cela, l'algorithme de sélection des nœuds MPR sélectionne d'autres nœuds supplémentaires à l'ensemble de nœuds MPR minimal, en choisissant pour cela des nœuds convenables tout en les mettant à jours pour l'environnement mobile, et en s'aidant des informations fraîches que garantit le protocole proactif MIARP.

5.2.5 Résolution de problème de liens unidirectionnels

A travers les informations de voisinage collectées par le protocole MIARP, les nœuds peuvent connaître leurs voisins unidirectionnels et bidirectionnels. Par conséquent, ils peuvent éviter au maximum la sélection des liens unidirectionnels pour la transmissions des différents messages dans le réseau, et envoyer leurs données

multicast. Une chose qui n'a pas été prise en considération dans MZRP (voir la figure 5.14) .



a) Liens unidirectionnels dans MZRP

b) Liens unidirectionnels dans MZRP-MPR

FIG 5.14- *Résolution de problème des liens unidirectionnels dans MZRP-MPR*

5.2.6 Conclusion

Dans ce chapitre, j'ai commencé par présenter l'avantage d'utiliser la technique MPR à la place de la diffusion par inondation dans les réseaux ad hoc sans fil. Cet avantage a été exprimé sous forme d'une simulation réalisé par le Dr Anis LAOUITI dans sa thèse de doctorat [96]. Les résultats de simulations nous ont permis de conclure que la technique de l'inondation est puissante, mais elle engendre un trafic important avec beaucoup de duplications inutiles. La technique des relais multipoints se présente comme une meilleure alternative pour éviter le gaspillage de la bande passante.

Plusieurs auteurs ont amélioré certains protocoles de routage par l'introduction de la technique MPR, comme par exemple OLSR [96], MMDV [138] dans l'unicast ; et le protocole ODMRP-MPR [100] dans le multicast. Tous les auteurs ont conclus que les performances de leurs protocoles de base ont été améliorées à travers l'ajout de la technique MPR ; tout en garantissant le bon fonctionnement, la grande scalabilité et l'optimisation de temps et de la bande passante.

De ma part j'ai décidé d'améliorer le protocole de routage multicast hybride MZRP par l'ajout de cette technique MPR ; l'originalité de mon travail consiste en le choix d'un protocole hybride, composé de plusieurs autres mécanismes visant à minimiser la consommation de la bande passante. Ma proposition a engendrer le nouveau protocole MZRP-MPR dont le fonctionnement a été décrit dans le chapitre précédent ; et dont une analyse qualitative a été présentée dans ce chapitre. Cette analyse, en se référant bien sûr des différents résultats des travaux précédents sur la technique MPR, a permis de montrer l'impact positif de l'ajout de la technique MPR dans le protocole MZRP.

CONCLUSION ET PERSPECTIVES

Il est certain que les réseaux ad hoc sans fil ont un bel avenir devant eux, avec une grande flexibilité de déploiement. Ces réseaux offrent une solution pour des applications demandant une mobilité fréquente dans un environnement dynamique, comme des pompier organisant les secours après une catastrophe naturelle.

Ce mémoire s'est focalisée sur ces réseaux ad hoc sans fil. il a présenté une sélection des protocoles de routage unicast et multicast, et a étudié en particulier le protocole multicast hybride MZRP, ainsi que la technique de diffusion optimisée MPR.

Dans ce mémoire, j'ai proposé une nouvelle amélioration pour le protocole multicast MZRP, par l'introduction de la technique de diffusion optimisée MPR qui a permet de générer le protocole MZRP-MPR.

J'ai présenté aussi dans ce manuscrit une étude par simulation de la technique MPR, réalisé par le Dr. Anis LAOUITI dans sa thèse de doctorat. Le gain de la bande passante de cette technique par rapport à la technique de diffusion par inondation a été mis en évidence.

Une analyse qualitative a été faite vers la fin de ce mémoire pour montrer l'impact positif d'introduire une telle technique d'optimisation dans un protocole multicast hybride.

Autre point ; l'étude par simulation est indispensable pour évaluer les performances des différents protocoles de routage. Malheureusement, dans mon cas, je n'ai pas réussit à le faire à cause de plusieurs difficultés comme : le manque de code source du protocole de base MZRP. En pensant à implémenter un tel protocole, en se trouve devant un autre travail complexe qui consiste à implémenter et adapter plusieurs autres mécanismes et protocoles comme (AODV, MAODV, OLSR, MOLSR, IARP, MIARP, IERP, MIERP, BRP avec ses mécanismes de contrôle QD, ET,…), et la technique MPR en plus. Devant tout ce travail qui nécessite lui aussi d'autres codes sources et un travail exacte, et beaucoup d'aide, je me suis dévié vers une analyse qualitative en s'aidant des résultats de simulation de la technique qui nous intéresse, MPR.

Le monde des réseaux ad hoc sans fil est encore dans une phase de gestation. Plusieurs domaines et axes de recherche intéressants liés à ce type de réseau reste à explorer et auxquels il faudra trouver des réponses adéquates. je propose dans les futurs travaux,

d’implémenter les différents composants de MZRP pour pouvoir le simuler. D’implémenter MZRP-MPR afin de le comparer avec la version d’origine MZRP. je propose aussi l’amélioration des autres protocoles avec l’ajout de la technique relais multipoints dynamique (MPR-Dynamique), l’accumulation des chemins et routes multiples.

Bibliographie

[1] **J.Cartigny** « *Contribution à la diffusion dans les réseaux ad hoc* » thèse soutenue le 19 décembre 2003, pour l'obtention de titre de Docteur en informatique. Université des Sciences et technologies de Lille, LIFL – UMR 8022 – Cité Scientifique, Bât.M3 – 59566 Villeneuve d'Ascq Cedex.

[2] **N. Bansal and Z. Liu.** "*Capacity, delay and mobility in wireless ad-hoc networks*". In Proc. IEEE INFOCOM 2003, San Francisco, USA, 2003.

[3] **B. Leiner, D. Nielson, and F. Tobagi**, editors. Proc. IEEE "*Special Issue, Packet Radio Networks*", volume 25, January 1987.

[4] **G. Zussman and A. Segall**. "*Energy efficient routing in ad hoc disaster recovery networks*". In Proc. IEEE INFOCOM 2003, San Francisco, USA, 2003.

[5] **R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D.S.J.** "*De Couto. CarNet: A scalable ad hoc wireless network system*". In Proc. 9th ACM SIGOPS European workshop: Beyond the PC: New Challenges for the Operating System, Kolding, Denmark, September 2000.

[6] **D. Estrin, R. Govindan, J.S. Heidemann, and S. Kumar**. "*Next century challenges: Scalable coordination in sensor networks. In Mobile Computing and Networking*", pages 263–270, 1999.

[7] **P. Fraignaude and E. Lazard.** "*Methods and problems of communication in usual networks*" Discrete applied mathematics, 53:79 – 133, 1994. (special issue on broadcasting).

[8] **G. Chelius et E. Fleury** "*Critères d'évaluation d'arbres multicast Ad Hoc*" CITI INSA de Lyon – ARES INRIA Rhôme-Alpes.

[9] **C. Berge**. "*Graphes et Hypergraphes*" Dunod, 1973. 2eme ed.

[10] **N. Abramson.** "*The Aloha system, Another alternative for computer communications*". In Proc. Fall Joint Computer Conference, pages 281–285, 1970.

[11] **L. Kleinrock and F.A. Tobagi**. "*Packet switching in radio channels: carrier sense multiple- access modes and their througput-delay characteristics*". IEEE Trans. Commun., 12 :1400–1416, 1075.

[12] **IEEE Computer Society LANMAN Standards Committee**. "*Wireless lan medium access Control (mac) and physical layer (phy)*". IEEE Std. 802.11-1197.

[13] **C-K Toh** "*Ad Hoc Mobile Wireless Networks: Protocols and Systems*", Age of Pervasive Mobile Networking and Computing, 2002 by Prentice Hall PTR. Prentice - Hall, Inc. Upper Saddle River, NJ 07458.

[14] **F. Talucci and M. Gerla. Maca-bi** (maca by invitation). "A *wireless MAC protocol for high speed ad hoc networking*". In IEEE ICUPC'97. IEEE, 1997.

[15] **F. Talucci, M. Gerla, and L. Fratta. Maca-bi**(maca by invitation)-"*A receiver oriented access protocol for wireless multihop network*". In IEEE PIMRC'97. IEEE, 1997.

[16] **P. Karn. Maca** – "*A new channel access method for packet radio*". In ARRL/CRRL Amateur Radio 9th Computer Networking Conference, pages 134–140, 1990.

[17] **V. Bharghavan, A.J. Demers, S. Shenker, and L. Zhang**. "*MACAW: A media access protocol for wireless LAN's. In Comput*". Commun. Rev, pages 212–225, October 1994.

[18] **C.R. Lin and M. Gerla. Maca/pr**: "*An asynchronous multimedia multihop wireless network*". In IEEE INFOCOM'97. IEEE, 1997.

[19] **Y. Chen, E.G. Sirer, and S.B. Wicker**. "*On selection of optimal transmission power for ad hoc networks*". In Proc. 36th Hawaii International Conference on System Sciences (ICSS'2003), Hawaii, January 2003.

[20] **C.C. Chiang, H.K. Wu, W. Liu, and M. Gerla**. "*Routing in clustered multihop mobile wireless networks with fading channel*". In Proc. IEEE Singapore International Conference on Networks, 1997.

[21] **Z.J. Haas and J. Deng.** "*Dual busy tone multiple access (dbtma) - performance evaluation*". In VTC'99, Houston, TX, May 1999.

[22] **J. Deng and Z. Haas**. "*Dual busy tone multiple access (dbtma): A new medium access control for packet radio networks*". In Proc. IEEE ICUPC'98, Florence, Italy, October 1998.

[23] **Suresh Singh and C.S. Raghavandra**, "*PAMAS – Power Aware Multi-Access Protocol with Signalling for ad Hoc networks*", in ACM Computing Communications Review, July 1998.

[24] **S. Singh and C. Raghavendra.** "*Pamas: Power aware multi-access protocol with signalling for ad hoc networks*". In ACM Computer Communications Review, 1999.

[25] **C.K. Toh, Vasos Vassiliou, Guillirmo Guichal, and C.H. Shih**, "*MARCH: A Medium Access Control Protocol For Multihop wireless Ad Hoc Networks*", in Proceedings of IEEE Military Communications, October 2000.

[26] **D. Johnson.** "*Routing in ad hoc networks of mobile hosts. In Workshop on Mobile Computing Systems and Applications*", Santa Cruz, CA, U.S., 1994.

[27] **L.M. Feeney**. "*A taxonomy for routing protocols in mobile ad hoc networks*". Technical Report T99/07, SICS (Swedish Institute of Computer Science), Sweden, October 1999.

[28] **S. Ramanathan and M. Steenstrup.** "*A survey of routing techniques for mobile communications networks*". ACM/Baltzer Mobile Networks and Applications, 1(2) :89–104, 1996.

[29] **A. Jain.** "*Routing protocols for mobile ad-hoc networks*". Technical report, Departement of computer science and engineering, Indian institute of Technology, Kanpur, 2000.

[30] **L.R. Ford Jr. and D.R. Fulkerson**. "*Improving the routing and addressing of IP*". IEEE Journal Network Magazine, 7 :10–15, May 1993.

[31] **C.E. Perkins and P. Bhagwat**. "*Highly dynamic destination-sequenced distance vector (DSDV) for mobile computers*". ACM SIGCOMM '94 Computer Communications Review, 24(4) :234–244, October 1994.

[32] **C.P.. P. Bhagwat**, « *highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers* », in Proceedings of ACM SIGCOMM' 94, pp. 234-244, September 1994.

[33] **S. Murthy and J.J. Garcia-Luna-Aceves**. "*A routing protocol for packet radio networks*". In Proc. ACM First International Conference on Mobile Computing & Networking (MOBICOM'95), November 1995.

[34] **Jyoti Raju and J.J. Garcia-Luna-Aceves**, "*A Comparison of On-demand and Table-driven Routing for Ad Hoc Wireless Networks*," in Proceedings of IEEE ICC, June 2000.

[35] **C.C. Chiang, H.-K. Wu, W. Liu, and M. Gerla**, « *Routing in Clustered Multihop Mobile Wireless Networks with fading Channel,* » in Proceedings of IEEE Singapore International Conference on Networks, 1997.

[36] **T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot**. "*Optimized link state routing protocol*". In Proc. IEEE INMIC, Pakistan, 2001.

[37] **T. Clausen, P. Jacquet, A. Laouiti and P. Minet, P. Muhlethaler, A. Qayum, and L.Viennot.** "*Optimized link state routing protocol*". IETF Internet Draft, http://www.ietf.org/ internet-drafts/ draft-ietf-manet-olsr-07.txt, December 2002.

[38] **Bhagrav Bellur, Richard G. Ogier, Fred L. Templin**, "*Topology Broadcast based on Reverse-Path Forwarding (TBRPF)*", draft-ietf-manet-tbrpf-05.txt, Draft IETF, March 2002.

[39] **M. Gerla, X. Hong, G.Pei**, "*Fisheye State Routing Protocol (FSR) for Ad hoc Networks*", draft-ietf-manet-fsr-02.txt, Draft IETF, December 2001.

[40] **G. Pei, M. Gerla, T. W. Chen,** "*Fisheye State Routing: A Routing Scheme for Ad hoc Wireless Networks*", Proceedings of the IEEE International Conference on Communications (ICC), pages 70-74, New Orleans, LA, June 2000.

[41] **M. Gerla, X. Hong, L.MA, G. Pei,** "*Landmark Routing Protocol (LANMAR) for Large Scale Ad Hoc Networks*", draft-ietf-manet-lanmar-03.txt, Draft IETF, December 2001.

[42] **M. Gerla, X. Hong, G. Pei**, "*Landmark Routing for Large Ad Hoc Wireless Networks*", Proceedings of IEEE GLOBECOM 2000, San Francisco, CA, November 2000.

[43] **C.E. Perkins and E.M. Royer**. "*Ad-hoc on-demand distance vector routing*". In Proc. Second An-nual IEEE Workshop on Mobile Computing Systems and Apllications, pages 90–100, February 1999.

[44] **C.E. Perkins, E.M. Royer, and S.R. Das.** "*Ad hoc On-Demand Distance Vector (AODV) Routing*". IETF Internet Draft, http ://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-12.txt, Novem-ber 2002.

[45] **D.B. Johnson, D.A. Maltz, Y. Hu, and J.G. Jetcheva**. "*The dynamic source routing protocol for mobile ad hoc networks*". IETF Internet Draft, http ://www.ietf.org/internet-drafts/draft- ietf-manet- dsr-07.txt, February 2002.

[46] **David B. Johnson and David A. Maltz,** "*Mobile Computing. Kluwer Academic Publishers*", 1996.

[47] **V.D. Park and M.S. Corson.** "*A highly adaptive distributed routing algorithm for mobile wireless networks*". In Proc. IEEE INFOCOM '97, Kobe, Japan, April 1997.

[48] **R. Dube, C. Rais, K. Wang, and S. Tripathi**. "*Signal stability based adaptive routing (ssa) for ad hoc mobile networks*". IEEE Personal Communication, February 1997.

[49] **C.-K. Toh.** « *Ad Hoc Mobile Wireless Networks: Protocols and Systems* », chapter 6. Prentice Hall, 2002.

[50] **Z.J. Haas and M.R. Pearlman**. "*The performance of query control schemes for the zone routing protocol*". In ACM SIGCOMM'98, 1998.

[51] **Mingliang Jiang, Jinyang Li, Y. C. Tay**, « *Cluster Based Routing Protocol (CBRP)* », draft-ietf-manet-cbrp-spec-01.txt, draft IETF, August 1999.

[52] **P. Sinha, R. Sivacumar, and V. Bharghavan**, "*MCEDAR: Multicast core extraction distributed ad-hoc routing*", In Proc. Of the Wireless Communications Networking Conference, November 1999.

[53] **K. Grace**, "*Mobile Mesh Link Discovery Protocol*", draft-grace-manet-mmldp-00.txt, Draft IETF, September 2000.

[54] **K. Grace**, "*Mobile Mesh Routing Protocol*", draft-grace-manet-mmrp-00.txt, draft IETF, September 2000.

[55] **K. Grace,** "*Mobile Mesh Border Discovery Protocol*", draft-grace-manet-mmbdp-00.txt, Draft IETF, September 2000.

[56] **Y.–B. Ko, N. H. Vaidya**. "*Location- Aided Routing (LAR) in mobile Ad Hoc Networks*", Proceeding of the 4th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pages 66-75, Dallas, Texas, October 1998.

[57] **J. Gomez, A. Campbell,** "*Power Aware Routing Optimization for Wireless Ad Hoc Networks*", High Speed Networks Workshop (HSN 2001), June 2001.

[58] **L. Zhou and Z.J. Haas**. "*Securing ad hoc networks*". IEEE Network Magazine, 13(6), 1999.

[59] **M. Hauspie, J. Carle, and D**. "*Simplot. Partition detection in mobile ad-hoc networks*". In Proc. 2nd IFIP Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET'2003), Mahdia, Tu-nisia, 2003.

[60] **M. Hauspie, J. Carle, and D. Simplot**. "*Partition detection in mobile ad-hoc networks using multiple disjoint path set*". In 1st International Workshop on Objects models and Multimedia technologies (OMMT), Genova, Switzerland, 2003.

[61] **M. Hauspie, J. Carle, and D. Simplot**, "*Partition detection in mobile ad hoc networks using multiple disjoint path set*". In 1st International Workshop on Objects models and Multimedia technologies (OMMT), Genova, Switzerland, 2003. To appear.

[62] **J. Moy,** "*Multicast Ectensions to OSPF*", IETF RFC 1584, March 1994

[63] **S. Deering**, "*Host Extensions for IP Multicasting*", IETF RFC 1112, May 1988.

[64] **S. Deering, D. Farinacci, V. Jacobson, C.Lui, and L. Wei,** "*An Architecture for Wide Area Multicast Routing*", in Proceedings of ACM SIGCOMM'94, September 1994.

[65] **H. Eriksson,** "*MBone: The Multicast Backbone*," in Communications of the ACM, August 1994.

[66] **D. Waitzman, C. Pratidge, and S.Deering,** "*Distance Vector Multicast Routing Protocol (DVMRP)*," in Internet Work Group RFC 1584, March 1994.

[67] **Y. K. Dalal, R. M. Metcalfe**, "*Reverse path forwarding of broacast packets*", Communications of the ACM, 21(12): 1040-1048, December 1978.

[68] **S. Deering,** "*Multicast Routing in a datagram internetwork*", Phd Thesis, Standford University, California, USA 1991.

[69] **S. Deering,** "*Multicast Routing in Internetworks and Extended LANs*", in SIGCOMM Summer 1998 Proceedings, August 1998.

[70] **A. Adams, J. Nicholas**, "*Protocol Independent Multicast – Dense Mode (PIM-DM)*", draft-ietf-pim-dm-new-v2-01.txt, IETF draft, February 2002.

[71] **A. Ballardie,** "*Core Based Trees (CBT) Multicast Routing Architecture*", IETF RFC 2201, September 1997.

[72] **A. Ballardie**, "*Core Based Trees (CBT version 2) Multicast Routing*", IETF RFC 2189, September 1997.

[73] **B. Fenner, M Handley, H. Holbrook, I. Kouvelas**, "*Protocol Independent Multicast Sparse Mode (PIM-SM): Protocol Specification (Revised)*", ddraft-ietf-pim-sm-v2-new-05.txt, IETF draft, March 2002.

[74] **S. Deering, and al**, "*Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*", IETF RFC 2362, June 1998.

[75] **W. Fenner,** "*Internet Group Management Protocol, Version 2*", RFC 2236, November 1997.

[76] **S-Y. Ni, Y-C Tseng, Y-S Chen, and J-P Sheu**, « *The Broadcast Storm Problem in a Mobile Ad Hoc Network*, » in Proceedings of ACM / IEEE MOBICOM, August 1999.

[77] **Mingyan Liu, Rajesh R. Talpade, and Anthony McAuley**, "*AMRoute: Ad hoc Multicast Routing Protocol*", in University of Maryland Technical Research Report (CSHCN T.R. 99-1), 1999.

[78] **C. W. Wu, and Y. C. Tay**, "*AMRIS: A Multicast Protocol for Addd hoc Wireless Networks*," in Proceedings of IEEE MILCOM, November 1999.

[79] **C. Perkins and E. Royer**, "*Ad-Hoc On-Demand Distance Vector Routing*", in Proceedings of 2nd IEEE Workshop on Mobile Computing System and Applications, February 1999.

[80] **J.J Garcia-Luna-Aceves and E. L. Madruga**, "*The Core Assisted Mesh Protocol*", in IEEE Journal on Selected Areas in Communications, August 1999.

[81] **M. Gerla, Guangyu Pei, et al**, "*On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks*," in IETF Internet draft, November 1998.

[82] **Y. B. Ko and N. Vaidya**, "*Location-based Multicast in Mobile Ad Hoc Networks*," in Technical Report, Texas A&M, October 1998.

[83] **C. GUI and P. MOHAPATRA**, "*Efficient Overlay Multicast for Mobile Ad hoc Networks*," in IEEE WCNC' 03, 2003

[84] **M. Gerla, C. Chiang, and L. Zhang**, "*Tree multicast strategies in mobile, multihop wireless networks"*, in ACM Mobile Networks and Applications, vol. 4, 1999, pp. 193-207.

[85] **S. Yang, and J. Wu,** "*New Technologies of Multicasting in MANET*", Departement of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431.

[86] **L. JI, and M.S.Corson**, "*Differentiel Destination Multicast- a MANET multicast routing protocol for small groups*", in Proc. Of INFOCOM 2001, 2001, pp. 1192-1202.

[87] **K. Chen and K. Nahrstedt**, "*Efficient Location- Guided Tree Construction Algorithms for small group multicast in MANET*", in Proc. Of INFOCOM 2002, 2002.

[88] **R. Sivakumar, P.Sinha, and V. Bharghavan**, "*Core Extraction Distributed Ad hoc Routing (CEDAR) specification*," Internet Draft draft-ietf-manet-cedar-spec-00.txt, Sept. 1998.

[89] **S. Das, B. Manoj, and C.Murthy**, "*A dynamic Core Based Multicast Routing Protocol for Add hoc Wireless Networks*," in ACM MOBIHOC, June 2002.

[90] **K. Tang, K. Obraczka, S. Lee, and M. Gerla**, "*Reliable Adaptive Lightweight Multicast Protocol*", in IEEE International Conference on Communications (ICC 2003), 2003.

[91] **J. Luo, P. Eugster, and J. Hubaux**, "*Route Driven Gossip: Probabilistic reliable multicast in ad hoc networks"*. In Proc. Of INFOCOM 2003, 2003.

[92] **S-J.Lee, M.Gerla, and C-C.Chiang**, "*On-Demand Multicast Routing Protocol*". Wireless Adaptive Mobility Laboratory, Computer Science Department. University of California Los Angeles, CA 90095-1596. LA Sep 1999.

[93] **Z.Haas and M.Pearlman**, "*The Zone Routing Protocol (ZRP) for Ad hoc Networks*," Inetrnet Draft, draft-ietf-manet-zone-zrp-04.txt, Aug.2002.

[94] **V.Park, S.Corson** "*Temporally-Ordered Routing Algorithm (TORA) routing protocol*" version 1, Internet draft, draft-ietf-manet-tora-spec- 03.txt, work in progress, June 2001.

[95] **C-K Toh,** "*Associativity-Based-Long-Livedd Routing*", chapitre 6 du livre "Ad Hoc Mobile Wireless Networks", Age of Pervasive Mobile Networking and Computing.

[96] **A. Laouiti**, « *Unicast et Multicast dans les réseaux Ad hoc sans fil* », thèse de doctorat présenté à l'université de Versailles. Saint-Quentin. Juillet 2002.

[97] **M. Lee, Y.K. Kim**, "*PatchODMRP: an ad-hoc multicast routing protocol*", Information Networking, 2001. Proceedings. 15th International Conference on, 2001, Page(s): 537-543.

[98] **IEEE Computer Socitey LAN MAN Standards Committee**, "*Wireless LAN Medium Access Protocol (MAC) and physical layer (PHY) Specification*", IEEE Std 802.11- 1997. IEEE New York, 1997.

[99] **S. Byounghan, H.Jeon, and J.Lee**, "*PEODMRP: Performance Enhanced On-Demand Multicast Routing Protocol in Ad-Hoc Networks*", Department of EEE, Yonsei University. Institute of Information Technology Assessment under (2003-2-0989).

[100] **Y. Zhao, L.Xu, M.Shi**, "*On-Demand Multicast Routing Protocol with Multipoint Relay (ODMRP-MPR) in Mobile Ad-Hoc Network*", Proceedings of ICCT 2003, Page (s): 1295-1300.

[101] **C.Chiang, M.Gerla, and L.Zhang**, "*Forwarding Group Multicast Protocol (FGMP) for multihop wireless networks*", cluster computing, 1.No.2 (118), pp. 187-196.

[102] **E.M.Royer and C.E.Perkins**, "*Multicast ad hoc On-Demand Distance Vector (MAODV) Routing", Internet-Ddraft,"* draft-ietf-manet-maodv-00.txt", July 2000.

[103] **R.Sivakumar, P.Sinha and V.Bharghavan**, " *CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm*", IEEE Journal on Selected Areas in communication, vol 17, No 8, August 1999.

[104] **V.Devarapalli and D.Sidhu**, "*MZR: A Multicast Protocol for Mobile Ad Hoc Networks*", in IEEE International Conference on Communications (ICC), (Helsinki, Finland), Jun 2001.

[105] **A.Laouiti, P.Jacquet, P.Minet, L.Viennot, T.Clausen and C.Adjih**, "*Multicast Optimized Link State Routing*", Institut National de Recherche en Informatique et en automatique. ISSN 0249-6399, ISRN/RR- 4721—FR+ENG. N° 4721, Février 2003.

[106] **M. S. Corson and S. G. Batsell,** "*A reservation-based multicast (RBM) routing protocol for mobile networks: initial route construction phase*," ACM/Baltzer Wireless Networks, Vol. 1, 1995, pp.427-450.

[107] **L. Ji and M. S. Corson, "***Alightweight adaptive multicast algorithm*," in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '98), 1998, pp. 1036-1042.

[108] **E. Bommaiah, M. Liu, A. Mc Auley, and R. Talpade,** "*AM-route: ad-hoc multicast routing,"* Draft-talpade-manet-amroute-00.txt, Internet-Draft, IEEE, 1998.

[109] **E. M. Royer and C. E. Perkings,** "*Multicast operation of the ad-hoc on-demand distance vector routing protocol*", in Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99), 1999, pp.207-218.

[110] **E. L. Madruga and J. J. Garcia-Luna-Aceves, "***Scalable multicasting: the core assisted mesh protocol*," ACM/Baltzer Mobile Networks and Applications, Vol. 6, 200, pp. 151-165.

[111] **S. J. Lee, M. Gerla, and C. C. Chiang, "***On-demand multicast routing protocol,"* in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC' 99), 1999, pp. 1298-1304.

[112] **S. J. Lee, W. Su, and M. Gerla, "***On-demande multicast routing protocol in multicast wireless mobile networks,"* ACM/Baltzer Mobile Networks and Applications, Vol. 7, 2002, pp. 441-453.

[113] **Z. J. Haas, "***A new routing protocol for the reconfigurable wireless networks*", in Proceedings of IEEE International Conference on Universal Communications (ICUPC' 97), 1997, pp. 562-566.

[114] **Xiaofeng Zhang and LillyKutty Jacob,** "*MZRP: An Extension of the Zone Routing Protocol for Multicasting in MANETs",* Journal of Information Science and Engineering 20, 535-551 (2004).

[115] **Z. J. Haas, M. R. Pearlman, and P. Samar,** "*the intrazone routing protocol (IARP) for ad hoc networks"*, Draft-ietf-manet-zone-iarp-02.txt, Internet-Draft, IETF, 2002.

[116] **Z. J. Haas, M. R. Pearlman, and P. Samar, "***The Interzone routing protocol (IERP) for ad hoc networks,"* Draft-ietf-manet-zone-ierp-02.txt, Internet-Draft, IETF, 2002.

[117] **Z. J. Haas, M. R. Pearlman, and P. Samar, "***The bordercast resolution protocol (BRP) for ad hoc networks",* Draft-ietf-manet-zone-brp-02.txt, Internet-Draft, IETF, 2002.

[118] **T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, «***Optimised Link State Routing Protocol »* Draft-ietf-manet-olsr-07.txt, Internet-Draft, IETF.

[119] *"Analysis of the Zone Routing Protocol"*

[120] **Eitan Altman and Tania Jiménez**, "*NS Simulator for beginners"* Lecture notes, 2003-2004. University de Los Andes, Mérida, Venezuela and ESSI, Sophia-Antipolis, France. December 4, 2003.

[121] **P. Anelli and E. Horlait,** *"Principe de conception et d'utilisation »* Version 1.3. UPMC-LIP6 : Laboratoire d'Informatique de Paris VI, Réseaux et Performances. URL://www.lip6.fr/rp/~pan.

[122] **Kevin. Fall and Kannan. Varadhan,** *"The ns Manual; Formerly ns Notes and Documentation"* The VINT Project, A collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. June 9, 2006.

[123] **S. J. Lee, W. Su, J. Hsu, and M. Gerla,** *"A performance Comparaison study of ad hoc wireless multicast protocols",* in proceedings of IEEE INFOCOM' 00, 2000, pp. 565-574.

[124] **P. Jacquet, P. Minet, P. Mühlethaler, N. Rivierre,** *« Increasing Reliability in cable-free radio LANs- Low level forwarding in HIPERLAN, »* Wireless Personal Communication, Vol4, No 1, 1997.

[125] **M. R. Gary and D. S Johnson. Computers and intractability,** "a guide to the theory of NP-completeness" W. H. Freeman, 1979.

[126] **T. Cormen, C. Leiserson, and R. Rivest,** "*Introduction to Algorithmes"* MIT Press, 1990.

[127] **C. H. Papadimitriou and K. Steiglitz.** "*Combinatorial Optimization: Algorithms and Complexity"* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[128] **Amir Qayyyum,** "*Analysis and evaluation of chanel access schemes and routing protocols for wireless networks",* Thèse de doctorat en Informatique, Université PARIS SUD-PARIS XI, 2000.

[129] **Anis Laouiti, Amir Qayyum, Laurent Viennot "***Multipoint Relaying: An efficient Technique for flooding in mobile wireless networks"*, rapport de recherché RR-3898, Mars 2000.

[130] **Laurent Viennot,** *"Complexity Results on Election of Multipoint Relays in Wireless Networks"*, rapport de recherché INRIA RR-3584 Décembre 1998.

[131] **Nicklas Beijar,** *"Zone Routing Protocol (ZRP)"*, Networking Laboratory, Helsinki University of Technology, Finland.

[132] **C.R. Lin, M. Gerla**, "*Adaptive clustering for mobile wireless networks*" IEEE J. Selected Areas in communications,15,7,1997.

[133] **M. Gerla, J.T.C. Tsai**, "*Multicluster, mobile, multimedia radio network, Wireless networks*" VoI 1, 1995.

[134] **M. Gerla, T.J kwon, G. pei**, " *on demand routing in large ad hoc wireless networks with passive clustering*", Proc. IEEE WCNC, September 2000.

[135] **J. Wu, H. Li**, "*On calculating connected dominating set for efficient routing in ad hoc wireless networks*" Proceeding Dial M, seattle, August 1999.

[136] S.Y. Ni, Tseng, Y.S. Chen, J.P. Sheu, « The broadcast strom problem in a mobile ad hoc network" Proceeding Mobicom seattle, August 1999.

[137] **I. Stojmenovic, M. Seddigh, J. Zunic**, "*Inetrnal nodes based broadcasting algorithms in wireless networks*", Proceeding of the 34th Annual Hawaii International Conference on System Sciences (HICSS 2001)

[138] **Abderrahmen Mtibaa**, "*Etude des performances du protocol MMDV: Multipath and MPR based AODV*", Thomson Lab, Paris, France

[a] http://www.geod.rncan.gc.ca/index_f/geodesy_f/gps_f.html

Table des Figures

Liste des tableaux