TABLES DES MATIERES

Liste des figures

Liste des Algorithmes

Introduction générale

Deux révolutions ont transformer le monde des télécommunications dans les derniers années : L'explosion d'Internet d'une part et la généralisation des communications mobiles sans-fil d'autre part. L'Internet tend depuis une dizaine d'années à absorber tous les types de réseaux, pour les fondre progressivement en un seul réseau de plus d'un milliard d'utilisateurs. Les frontières de l'Internet se sont étendues tout dernièrement aux réseaux sans-fil, qui ont eux aussi récemment connu une croissance phénoménale. La convergence semble maintenant inéluctable entre les technologies Internet et les technologies sans-fil : la perspective est celle d'un réseau ambiant omniprésent, qui déplacera à chaque instant et en temps réel des quantités considérables d'information multimédia, que les éléments du réseau et les destinataires soient fixes ou mobiles.

L'évolution des dispositifs sans fil a permet la manipulation de l'information à travers des unités de calculs portables qui ont des caractéristiques particulières (une faible capacité de stockage, une source d'énergie autonome..) et accèdent au réseau à travers une interface de communication sans fil. Comparant avec l'ancien environnement (l'environnement filaire), le nouvel environnement résultant appelé l'environnement mobile, permet aux unités de calcul, une libre mobilité et il ne pose aucune restriction sur la localisation des usagers. La mobilité et le nouveau mode de communication utilisé, engendrent de nouvelles caractéristiques propres à l'environnement mobile : une fréquente déconnexion, un débit de communication et des ressources modestes, et des sources d'énergie limitées.

Les réseaux mobiles, peuvent être classés en deux grandes classes : les réseaux avec infrastructure qui utilisent généralement le modèle de la communication cellulaire, et les réseaux sans infrastructure ou les réseaux ad hoc. Plusieurs systèmes utilisent déjà le modèle cellulaire et connaissent une très forte expansion à l'heure actuelle (les réseaux GSM par exemple) mais requièrent une importante infrastructure logistique et matérielle fixe.

La contrepartie des réseaux cellulaires sont les réseaux mobiles ad hoc. Dans ce type de réseau, chaque nœud jouant le rôle de l'hôte ainsi que du routeur. Les équipements mobiles dans ces réseaux sont généralement de petites tailles (PC portable, PDA, …) d'où les contraintes de ressources en terme de mémoire et de batteries. Ces réseaux sont caractérisés par la faible bande passante qui diminue également en raison des interférences des signaux ainsi que la déplétion sur le canal (channel fading). Les propriétés des réseaux ad hoc en font des solutions pratiques pour étendre l'accès aux bornes d'une infrastructure fixe (téléphonie

sans-fil, Wifi etc.), en dépassant la portée radio de ces bornes grâce au relais ad hoc que les utilisateurs peuvent se fournir les uns aux autres pour accéder indirectement à l'infrastructure. Cependant, l'aspect le plus novateur des réseaux ad hoc est de pouvoir fournir une couverture réseau mobile de manière automatique et autonome, et ce même sans accès à une infrastructure préexistante, toujours grâce au relais ad hoc que les utilisateurs peuvent se fournir les uns aux autres.

Le routage ou l'acheminement de données, consiste à assurer une stratégie qui garantit, à n'importe quel moment, la connexion entre n'importe quelle paire de noeuds appartenant au réseau. La stratégie de routage doit prendre en considération les changements de la topologie ainsi que les autres caractéristiques du réseau (bande passante, nombre de liens, ressources du réseau…etc.). En outre, la méthode adoptée dans le routage, doit offrir le meilleur acheminement des données en respect des différentes métriques de coûts utilisées.

Le problème de routage, dans les réseaux ad hoc, est compliqué par l'utilisation de communications par radio : la radio est en effet le medium le plus hostile à la propagation de l'information, du fait notamment des interférences entre utilisateurs et de la complexité du traitement du signal. D'autre part, le routage ad hoc est aussi compliqué par la mobilité des éléments susceptibles d'acheminer le trafic (c'est-à-dire les utilisateurs eux-mêmes). N'ayant pas été prévus pour ces dernières complications, les algorithmes de routage classiques ne peuvent donc pas être utilisés tels quels. Ils doivent être optimisés (si ce n'est entièrement revus) pour être efficaces dans les réseaux ad hoc : s'adapter aux communications radio en réduisant au maximum le trafic de contrôle nécessaire au bon fonctionnement du réseau, et en même temps rester en mesure de suivre dynamiquement la mobilité des éléments du réseau. Ce sont ces contraintes qui sont au fondement des algorithmes de routage ad hoc.

Dans ce mémoire nous allons étudier le problème de routage dans les réseaux mobiles ad hoc. Notre étude offre principalement une étude synthétique des travaux de recherche qui ont été fait, et qui se font à l'heure actuelle, dans le but de résoudre le problème d'acheminement de données entre les hôtes mobiles du réseau ad hoc. Dans ce contexte, nous présenterons des nouveaux mécanismes, permettant d'élire des chemins stables et durables entre les entités communicantes, ainsi que des mécanismes de prédiction de rupture de route pour les maintenir soit localement ou de bout en bout (de la source à la destination).

Organisation du mémoire

Le premier chapitre présente des généralités sur les réseaux sans fil, leurs caractéristiques, leurs applications ainsi que leurs classifications.

Le deuxième chapitre décrit les classifications des protocoles de routages les plus connus ainsi que les caractéristiques de chaque classe.

Dans le troisième chapitre nous présenterons un état de l'art sur les protocoles de routage existants.

Le quatrième chapitre est consacré aux concepts de modélisation des réseaux ad hoc (seulement les modèles de mobilités) ainsi la traditionnelle méthode pour évaluer un tel protocole de routage, qui est la simulation.

Le dernier chapitre décrit les détails de notre proposition ainsi que les résultats de simulation que nous avons obtenus.

Chapitre 1

Généralités sur les réseaux mobiles

L'évolution rapide de la technologie dans le domaine de la communication sans fil a permit à des usagers munis d'unités de calcul portables d'accéder à l'information indépendamment de sa localisation géographique. Ces unités, qui se communiquent à travers leurs interfaces sans fil, peuvent être de diverses configurations : avec ou sans disque, de capacités de sauvegarde et de traitement plus ou moins modestes et alimentés par des sources d'énergie autonomes (batteries). L'environnement de calcul résultant est appelé environnement mobile. Cet environnement n'astreint plus l'usager à une localisation fixe, mais il lui permet une libre mobilité tout en assurant sa connexion avec le réseau.

Dans ce chapitre, nous examinons certaines caractéristiques principales de la communication sans fil par rapport aux spécifications et la classification de ces réseaux. Nous passons ensuite à la description des réseaux Ad Hoc, leur définition, leurs applications et leurs contraintes.

Plusieurs technologies de réseaux sans fil existent. Ils se distinguent par la fréquence d'émission utilisée, le débit, la portée des transmissions et même le mode de fonctionnement. Plusieurs classifications sont définies, les plus connues sont les suivantes :

1.1 Classification des réseaux mobiles

Plusieurs types de réseaux sans fil existent et peuvent être classifiés par diverses façons selon les critères de leur classification. Les classifications les plus connues sont **:**

1.1.1 Par formation et architecture de réseau

Les réseaux sans fil peuvent être divisés en deux larges catégories selon la façon dont le réseau est construit :

Avec infrastructure

Dans ce mode de fonctionnement, le réseau est composé de points d'accès appelés stations de bases (SB). Muni d'une interface de communication sans fil pour la communication directe avec les sites ou les unités mobiles (UM), une station de base couvre une zone géographique limitée dite portée (figure 1.1). Une unité mobile n'est rattachée, à un moment donné, qu'à une seule station de base lui offrant tous les services tant que l'UM est à l'intérieure de sa zone de couverture. Cette dernière faite office de pont entre réseau filaire et réseau sans fil, permettant de relier une UM à une unité connectée à un site fixe.

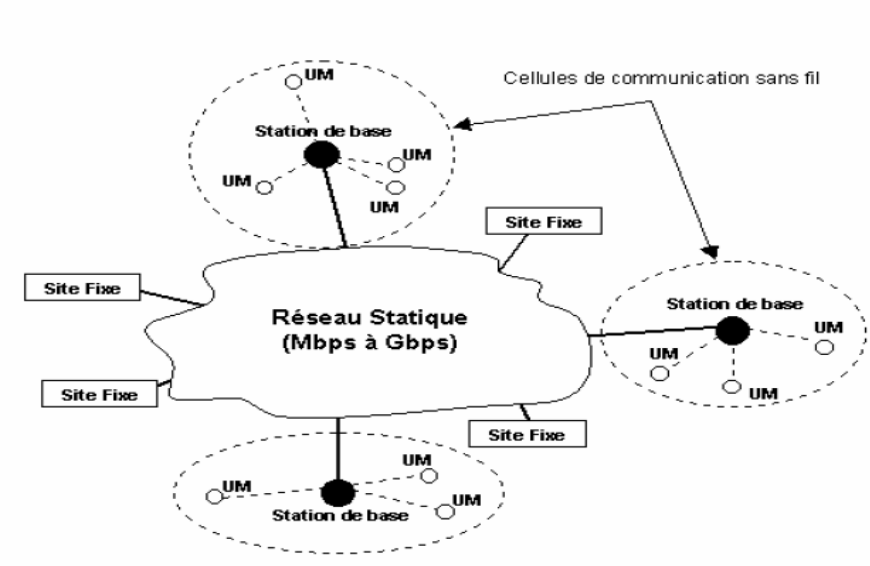


Figure 1.1 : Réseaux mobiles avec infrastructure.

Sans Infrastructure

Dans ce type de réseau, les nœuds sont tous autonomes et capables de se déplacer et de communiquer entre eux librement sans aucun recours à une infrastructure. L'absence d'infrastructure oblige les nœuds à jouer le rôle de routeurs (figure 1.2) et à participer au routage des données au profil des autres nœuds.

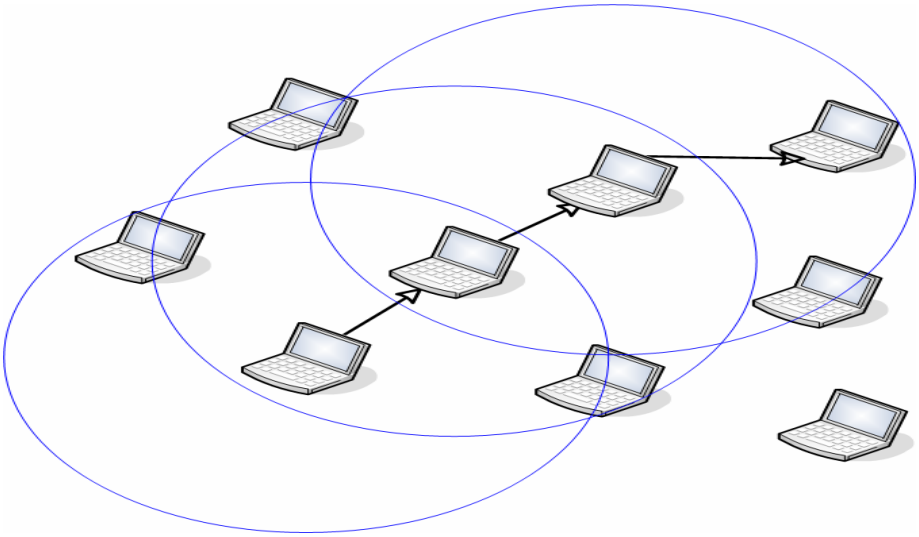


Figure 1.2 : Réseaux mobiles sans infrastructure.

1.1.2 Par champ de couverture du réseau

1.1.2.1 Réseaux Étendus Sans fil (WWAN: Wireless Wide Area Networks)

Les WWAN sont des réseaux avec infrastructure, fondé sur des infrastructures de gestion de réseau et des stations de base. Ce type de réseau permet aux utilisateurs mobiles d'établir des connexions sans fils qui excèdent les réseaux publics ou les réseaux privés. Ces connexions peuvent être établies au-dessus de grands secteurs géographiques, au-delà des villes ou même des pays, en utilisant des antennes ou des systèmes satellites maintenus par les fournisseurs de service sans fil. Les réseaux cellulaires (comme les réseaux GSM ou les réseaux CDMA) et les réseaux satellitaires sont des bons exemples des réseaux WAN sans fil.

1.1.2.2 Réseau métropolitain sans fil (WMAN : Wireless Metropolitan Area Network)

Les réseaux sans fil WMAN désignés parfois sous le nom de radio fixée. Ce sont des réseaux avec infrastructure qui permettent à des utilisateurs d'établir des connexions sans fil à une large bande s'étalant à une zone métropolitaine (par exemple, la connexion entre les bureaux des établissements d'une ville ou dans un campus universitaire). En outre, les WMAN peuvent servir comme moyen de communication de secours pour les réseaux câblés en cas d'épuisement des lignes spécialisées primaires. IEEE a installé un groupe de travail

802.16 pour le développement des normes sans fil à large bande d'accès de zone métropolitain [2].

1.1.2.3 Réseaux locaux sans fil (WLAN : Wireless Local Area Network)

Les réseaux locaux sans fils permettent à des utilisateurs d'établir des connexions sans fils dans un secteur limité géographiquement, typiquement à l'intérieur d'un établissement. WLAN's fournissent des systèmes de communications de données flexibles pouvant être employés dans des bureaux provisoires ou dans des espaces où l'installation du câblage étendu serait prohibitive. Ils peuvent aussi compléter un LAN existant de sorte que les utilisateurs puissent travailler dans différents endroits en se déplaçant. Plusieurs normes de WLAN ont été développées, nous citons les deux principales : IEEE 802.11b [3] et Hiperlan.

Wi-Fi (Wireless Fidelity)

Connu aussi sous la norme "IEEE 802.11b" [3]. Il constitue une solution de connexion réseau pratique et intéressante offrant mobilité, flexibilité, et faible coût de déploiement et d'utilisation. Le Wi-Fi promu par l'alliance WECA3 (Wireless Ethernet Compatibility Alliance) entérine des transmissions à 5.5 et 11 Mbits/s, sur un rayon de 50 à 100 mètres. Des évolutions sont d'ores et déjà à l'ordre du jour : 802.11g [3] offrant déjà 54 Mbp/s sur la bande de fréquences des 2,4 GHz ainsi la norme 802.11a [4] mais sur des fréquences de 5 GHz. Toutes ces normes sont basées sur la norme 802.11 [5] de l'IEEE. La norme IEEE 802.11 définit plusieurs couches physiques et une couche d'accès au médium. La norme IEEE 802.11 et ses déclinaisons ont été conçues à l'origine afin de constituer des réseaux locaux sans fil administrés par une ou plusieurs stations de base. Grâce à son succès commercial, il est très rapidement devenu incontournable dans le monde des réseaux ad hoc.

Hiperlan (High Performance Radio LAN)

HiperLAN [6] a bien connu des efforts de standardisations de WLAN par l'institut européen de télécommunications ETSI (European Telecommunications Standards Institute) ; plus précisement, il a été développé par le projet BRAN (Broadband Radio Access Networks) de l'ETSI. HiperLAN 2 [7] est la seconde version du standard, fournit une large bande tout en assurant une interopérabilité avec les réseaux sans fil de la troisième génération (par exemple l'UMTS).

HiperLAN 1 est défini pour fonctionner sur la bande de 5.2 GHz, fournissant un débit théorique de 20 Mbp/s. Malheureusement, HiperLAN1 n'a été pris par aucune compagnie commerciale et il est devenu rapidement désuet. l'ETSI-BRAN a proposé HiperLAN2, tout en souhaitant une meilleure acceptation dans le marché.

HiperLAN 2 est une seconde version développée par l'H2GF (HyperLan 2 Global Forum) fondé en 1999 par Bosch, Dell, Ericsson, Nokia, Telia et Texas Instruments. Ils ont été rejoints un an après par d'autres industriels, tel que Canon, Motorola ou encore Samsung.

Hiperlan2 offre un débit de 54 Mbp/s, et exploite la gamme de fréquence de 5 GHz, sur une portée de 100m.

1.1.2.4 Réseaux personnels sans fil ou WPAN (Wireless Personal Area Network)

Les technologies sans fil WPAN permettent à des utilisateurs d'établir des communications sans fil en utilisant des dispositifs sans fil personnels tels que les PDAs ou les téléphones portables. Ces derniers sont utilisés dans un espace de fonctionnement restreins, typiquement jusqu'à une gamme de 10 mètres. Deux grandes technologies sans fil principales du WPAN sont le Bluetooth et l'infrarouge. Le Bluetooth [8, 9] est une technologie qui remplace les câble et emploie les ondes radio pour transmettre des données à une distance jusqu'à 9 à 10 m. Les WPAN sans fil se développent rapidement en raison de ses basses complexités, ses basses puissances d'énergie, et leur interopérabilités avec les réseaux 802.11.

1.1.3 Par technologie d'accès

Les réseaux sans fil peuvent être classifiés selon la norme, la fréquence, et l'utilisation spécifique de spectre :

- Réseaux GSM
- Réseaux TDMA
- Réseaux CDMA
- Réseaux Satellite
- Réseaux Wi-Fi (802.11)
- Réseaux Hiperlan2
- Réseaux Bluetooth
- Réseaux infrarouges

1.1.4 Par types d'applications de réseau

Les réseaux sans fil peuvent être classifiés selon leurs applications et leurs utilisations.

- Réseau d'entreprise
- Réseau domestique
- Réseau de capteur
- …,

1.2 Réseau mobile Ad Hoc (MaNet)

1.2.1 Histoire des réseaux ad hoc mobiles

Au début des années 70, le réseau ad hoc mobile (MANET) s'appelé le réseau de radio en paquet qui a été patronné par le DARPA (Defense Avance Research Projects Agency). Ce dernier a fondé le projet appelé PRNets (Packet Radio Networks), radio en paquet, qui est constitué de plusieurs terminaux sans fil s'entrent communiquant sur des champs de batailles. Ces réseaux disposent d'une architecture de système distribuée, et partagent le canal de diffusion par une combinaison des protocoles Aloha et CSMA. Par la suite, en 1983, les SURAN (Survivable Radio Networks) furent développes par le DARPA. L'objectif était de passer outre les principales limitations des PRNet (en particulier permettre le passage à des réseaux comportant énormément de nœuds, gérant la sécurité, gérant l'énergie, et offrant des capacités de calcul suffisantes pour supporter des protocoles évolués). Les recherches sur ces types de réseaux restaient exclusivement militaires. Les applications civiles ou généralisées de ces réseaux appelés ad hoc n'apparues que beaucoup plus tard, vers la fin des années 1990 [10, 11].

1.2.2 Définition

Les réseaux ad hoc auxquels nous nous sommes intéressés sont ceux décrits et étudiés par le groupe de travail MANET (Mobile Ad hoc NETworks) de l'IETF (l'Internet Engineering Task Force). Une définition de ces réseaux est donnée formellement dans la RFC 2501 [12].

Un réseau mobile ad hoc est formé dynamiquement par un ensemble de nœuds mobiles qui sont reliés par l'intermédiaire des liens sans fil, sans recours à une infrastructure préexistante ou à une administration centralisée. Les nœuds sont libres de se déplacer aléatoirement et de s'organiser arbitrairement, ainsi la topologie du réseau peut changer rapidement et de manière imprévisible.

Les réseaux mobiles ad hoc sont des réseaux sans infrastructure puisqu'ils n'exigent aucune infrastructure fixe telle qu'une station de base pour leur fonctionnement. Les itinéraires dans un réseau ad hoc peuvent inclure des sauts multiples pour atteindre une cible demandée, et par conséquent, il est approprié d'appeler de tels réseaux « les réseaux ad hoc sans fil de multi-Sauts ». La figure 1. 3, montre un exemple de réseau mobile ad hoc et sa topologie de communication.

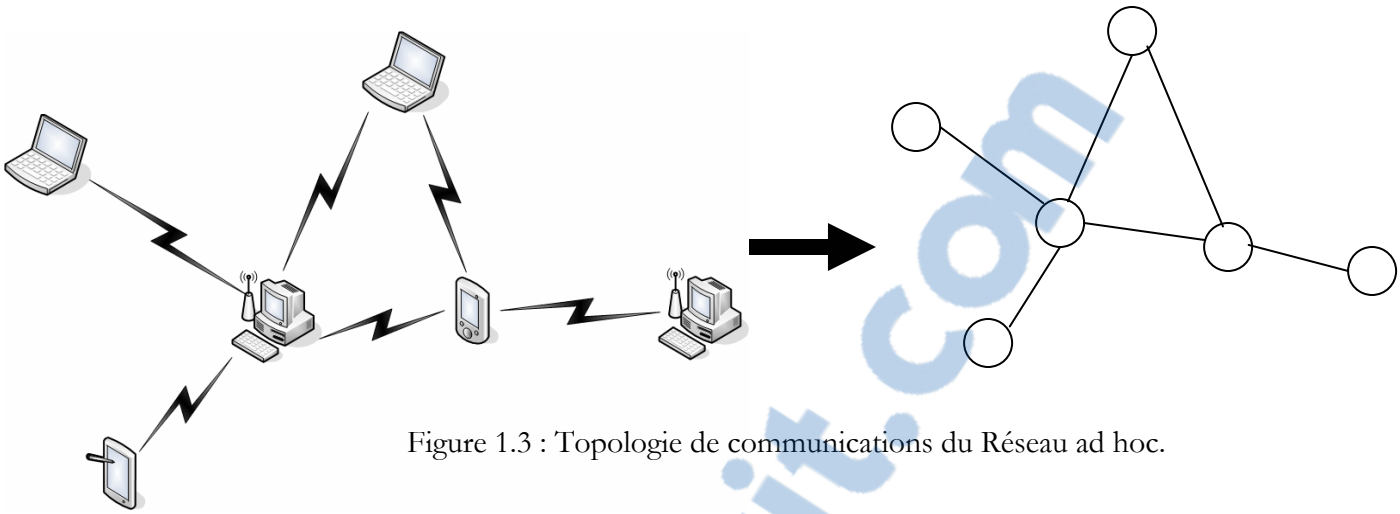


Figure 1.3 : Topologie de communications du Réseau ad hoc.

1.2.3 Applications des réseaux ad hoc

Les réseaux ad hoc présentent un intérêt important dans des environnements sans infrastructure. Les militaires les utilisent sur des champs de batailles pour établir des connexions entre les différentes unités mobiles. Ils peuvent également être utilisés par les organisations des secours après une catastrophe naturelle où toutes les infrastructures de communication ont été détruites, etc. Cependant, les réseaux Ad Hoc commencent à être déployés pour des applications plus classiques comme les conférences, ainsi l'établissement de petits réseaux sans fil de bon marché... . On peut conclure que plus les performances des réseaux ad hoc s'améliorent, plus leur utilisation pourrait se développer et les applications de ce type de réseau se diversifier.

1.2.4 Conception et contraintes

Les nœuds dans un réseau ad hoc se déplacent librement et aléatoirement et s'organisent arbitrairement, ainsi, la topologie du réseau peut changer rapidement et d'une manière imprévisible. La différence fondamentale entre les réseaux fixes et le MANET est que les nœuds (utilisateurs) dans un MANET sont mobiles. En raison de la mobilité de ces nœuds, certaines caractéristiques sont seulement applicables à MANET. On cite dans ce qui suit quelques caractéristiques de conception des réseaux MANET ainsi les contraintes relatives à ce type de réseaux :

- Pas d'infrastructures : Un réseau ad hoc ne possède pas d'infrastructures, qu'elles soient fixes ou mobiles. Les fonctionnalités réseaux sont donc intégralement prises en charge par des terminaux mobiles, ce qui explique le terme de nœud routeur pour tout terminal. Chaque nœud doit être capable de maintenir une vue partielle ou totale du réseau qui lui permettra de participer dans la fonction de routage.

- Changement dynamique de la topologie du réseau : Les nœuds sont libres de se déplacer arbitrairement, ce qui signifie que la topologie de réseau peut changer aléatoirement, rapidement et imprévisiblement.
- Limitation niveau physique : Les réseaux ad hoc utilisent un médium radio partagé pour les communications entre les mobiles. Il existe de nombreux problèmes propres aux réseaux radio tels que le problème du nœud caché, le problème du nœud exposé, ainsi que le problème de collision qui est naturel dans un environnement radio.
- Largeur et qualité de bande limitée : Les liens sans fil ont la capacité en largeur de bande sensiblement inférieure à leurs contreparties câblées. Ils sont également moins fiables dus à la nature de la propagation du signal.
- Diversité de liens et de nœuds : Chaque nœud peut être équipé d'une ou plusieurs interfaces radio, qui ont des possibilités variables de transmission/réception et fonctionnent par différentes fréquences de bandes. Cette hétérogénéité dans les capacités radio de nœud peut avoir comme conséquence des liens asymétriques. En outre, chaque noeud mobile pourrait avoir une configuration différente de software/hardware, et des capacités de traitement variables.
- Contrainte d'énergie : Les dispositifs dans un réseau mobile se fondent sur des batteries ou d'autres moyens épuisables en tant que leur source d'énergie. La conservation et l'utilisation efficace de l'énergie peuvent être le critère le plus important dans la conception de système pour ce type de réseau.
- Sécurité du réseau : Il est facile d'espionner un canal radio de manière passive. Les protections ne sont pas faites de manière physique, par exemple il est difficile d'empêcher quelqu'un de placer une antenne réceptrice très sensible dans le voisinage. Les mesures de protection doivent être mises en place de manière logique, utilisant les techniques de cryptographie ou éventuellement l'utilisation des antennes directionnelles. Mais le canal radio restera quoiqu'il en soit vulnérable à un brouillage massif (attaque de type dénie de service).
- Passage à l'échelle : Beaucoup d'applications de réseau ad hoc mobiles impliquent de grands réseaux formés de dizaines de milliers de nœuds, comme trouvé par exemple, dans les réseaux de capteurs et les réseaux tactiques. L'évolution vers un large réseau constitué par des nœuds avec des ressources limitées n'est pas évident et présente beaucoup de défis qui doivent toujours être résolus tels que le problème d'adressage, de routage, d'interopérabilité, de sécurité, etc.
- La qualité de service : De nombreuses applications et surtout les applications multimédias exigent certaines garanties et qualité de services pour leur bon fonctionnement tel que le débit de transmission, les délais (délai d'installation d'itinéraire, etc.), le nombre de paquets détruits, etc. Dans les réseaux ad hoc, ces exigences sont très difficiles à obtenir. Ceci est dû à la nature du canal radio d'une part (interférences et taux d'erreur élevés) et au fait que les mobiles se partagent le médium radio.

1.3 Conclusion

Dans ce chapitre, nous avons présenté un panorama des environnements mobiles sans fil et ses classifications, ainsi qu'un aperçu des applications qui s'y déploient et des technologies utilisées. Nous nous sommes ainsi penché sur les réseaux ad hoc, leur définition, leurs applications, et leurs contraintes.

Maintenir des communications stables dans les réseaux ad hoc avec une certaine qualité de service rendue aux applications est un problème de recherche d'actualité. Dans le chapitre suivant nous présenterons un état de l'art sur les classifications de protocoles de routage les plus connus, ainsi que les caractéristiques de chaque classe.

Chapitre 2

Classification des protocoles de routage

Le routage dans les réseaux ad hoc est devenu une matière populaire de recherche. Au début des années 80, un grand nombre de protocoles de routage sont conçus et proposés pour les réseaux ad hoc. Ces protocoles couvrent un éventail de choix et des approches de conception, des modifications simples des protocoles d'Internet à des protocoles hiérarchiques à multi niveaux plus complexes.

Les protocoles de routage qui ont été conçus, sont basés sur les ensembles semblables de suppositions. Par exemple, la plupart des protocoles de routage supposent que tous les noeuds ont des ressources et des possibilités homogènes. Ceci inclut les gammes de transmission des nœuds et les liens bidirectionnels qui sont souvent assumés. Dans ce dernier cas, les protocoles éliminent alors des liens unidirectionnels de la considération. Hors d'autres protocoles ont des mécanismes pour déterminer ces liens unidirectionnels.

Ce chapitre décrit les classifications des approches de routage les plus connues et leurs caractéristiques. Les protocoles de routage mentionnés comme exemples de classes, sont choisis pour un certain nombre de raisons. Ils peuvent être des choix populaires pour la recherche parmi la communauté ad hoc. Ils peuvent, à l'heure de cette écriture de ce mémoire, mis à l'étude par le groupe de travail ad hoc mobile des réseaux (MANET) de l'IETF pour la standardisation ou ils peuvent simplement être de bons exemples de protocoles d'illustration de leurs classes.

Dans les classifications citées par la suite, les classes de protocoles n'ont pas de structure sous forme d'arbre, bien que certaines caractéristiques, sont en général communes entre les classes. Quelques classes sont d'une nature d'opposition par exemple : réactive et proactive. D'autres n'auront pas des contreparties avec des caractéristiques intéressantes à préciser (comme le routage hiérarchique, puisque les protocoles non hiérarchiques ne

partagent pas des caractéristiques distinguées, à part qu'ils ne sont pas hiérarchiques) [13, 14,45,71].

2.1 Protocoles de routage proactifs versus protocoles réactifs

Les protocoles de routage dans les réseaux ad hoc peuvent être répartis en deux grandes classes à savoir : proactifs et réactifs, selon la façon dont les routes sont créées et maintenues. Une troisième classe peut être rajoutée, c'est celle des protocoles hybrides qui est la combinaison des deux classes. Les protocoles de routage proactifs maintiennent les routes de manière permanente vers toutes les destinations. Ces protocoles ont l'avantage de la disponibilité immédiate des routes. Cependant, un trafic de contrôle important est nécessaire pour mettre à jour ces routes. Les protocoles de routage réactifs établissent les routes uniquement à la demande "On-Demand". Ces protocoles ne consomment pas beaucoup de bande passante du réseau. En revanche, ils présentent un délai supplémentaire dû à la recherche d'itinéraire.

2.1.1 Protocoles de routage proactifs

Les approches proactives de routage, conçues pour les réseaux ad hoc, sont dérivés des protocoles traditionnels du vecteur d'état [16] et à vecteur de distance [17] développés pour l'usage dans les réseaux filaires. La caractéristique primaire des approches proactives est que chaque nœud dans le réseau maintient un itinéraire à tous les autres nœuds dans le réseau à tout moment. La création et l'entretien d'itinéraire sont accomplis par une combinaison des mises à jour périodiques et par le déclenchement des événements :

Les mises à jour périodiques se font par des échanges d'informations de routage entre les nœuds dans des intervalles de temps prédéfinis, indépendamment des caractéristiques de mobilité et du trafic du réseau.

Les mises à jour par le déclenchement d'événements, se faites par des échanges d'informations de routage toutes les fois qu'un certain événement est produit, tel qu'une addition de liens ou un déplacement d'un nœud. Dans ce cas, le taux de mobilité affecte directement la fréquence des mises à jour parce que les changements de lien se produisent à mesure que la mobilité augmente.

Les approches proactives ont l'avantage que les itinéraires sont disponibles à tous moment. Cependant, l'inconvénient majeur de ces protocoles est que les frais généraux de contrôle peuvent être important surtout dans de grands réseaux ou dans les réseaux de forte

mobilité. De plus, la quantité d'état de routage maintenue dans chaque nœud est de O(n), où n est le nombre de nœuds dans le réseau. Les protocoles proactifs tendent à être performants dans les réseaux où il y a un nombre significatif de sessions de données actives. Dans ces derniers, les frais généraux sont justifiés parce que la majorité de ces chemins sont utilisés [1].

Exemple de protocoles qui mettent à ajour leurs tables par le déclenchement d'événements :

CBRP [18], DSDV [19], TORA [20]…

Exemple de protocoles qui mettent à ajour leur table périodiquement :

OLSR [21], TBRPF [22] .....

2.1.2 Protocoles de routage réactifs

Les techniques de routage réactives, également appelées le *routage sur demande*, adoptent une approche très différente par rapport aux protocoles proactifs. Une grande partie des frais de contrôle des protocoles proactifs proviennent du besoin de chaque nœud de maintenir un itinéraire vers chaque autre nœud du réseau et à tout moment. Dans un réseau filaire, où les modèles de connectivité changent rarement et les ressources sont abondantes, maintenant la connectivité totale du graphe a des dépenses valables. Dans un réseau ad hoc, cependant, la connectivité de lien peut changer fréquemment et les frais généraux de contrôle sont coûteux. Pour ses raisons, les approches réactives de routage ont prit place aux approches de routage proactives. Ces approches réactives ne maintiennent pas, sans interruption, des itinéraires entre les nœuds du réseau. Au lieu de cela, les itinéraires sont seulement découverts quand ils sont réellement nécessaires. Quand un nœud source veut envoyer des paquets de données à une certaine destination, il vérifie sa table d'itinéraire à la recherche d'itinéraire vers cette destination. Si aucun itinéraire n'est trouvé, il lance une procédure de découverte d'itinéraire pour trouver un chemin à la destination. Par conséquent, la découverte d'itinéraire devient sur demande. Si deux nœuds ne doivent jamais communiquer entre eux, alors ils n'ont pas besoin d'utiliser leurs ressources pour le maintien d'un chemin entre eux. La découverte d'itinéraire se faite typiquement par l'inondation du réseau par des messages de demande d'itinéraire. Pour réduire les frais généraux, le secteur de recherche peut être réduit par un certain nombre d'optimisations.

L'avantage de cette approche est que les frais généraux des messages de contrôle sont susceptibles d'être réduits par rapport aux approches proactives, en particulier dans les réseaux avec moins de connexions actives. En contre partie, si le nombre de sessions de données dans le réseau est important, alors les frais généraux produits par les découvertes d'itinéraire s'approchent, et peuvent même surpasser les approches proactives. L'inconvénient des approches réactives est l'introduction d'une latence de temps pour

l'acquisition d'un itinéraire. C'est-à-dire, quand un itinéraire est demandé par un noeud source, il y a une durée finie pour que l'itinéraire soit découvert. En revanche, avec une approche proactive, les itinéraires sont en général disponibles au moment où ils sont nécessaires. Par conséquent, il n'y a aucun retard pour commencer la session de données.

Exemple de protocoles : ABR [23], AODV [32], DREAM [33], DSR [34], FORP [35], LAR [24], SSA [25].

2.1.3 Protocoles de routage Hybrides

Les protocoles de routage hybrides sont une nouvelle génération de protocoles, qui sont proactifs et réactifs en nature. Ces protocoles sont conçus pour augmenter la scalabilité en permettant aux nœuds, avec la proximité étroite, de travailler ensemble pour former une sorte d'épine dorsale pour réduire les overheads de découverte d'itinéraire. Ceci, est la plupart du temps, réalisé proactivement en maintenant des itinéraires entres les nœuds les plus proches et en déterminant les itinéraires aux nœuds lointains en utilisant une stratégie réactive. La plupart des protocoles hybrides proposés se basent sur des zones, ce qui signifie que le réseau est divisé ou vu en tant qu'un certain nombre de zones par chaque nœud. D'autres groupent les noeuds dans des arbres ou des clusters.

Exemple de protocoles : ADV [26], ZRP [27]...

2.2 Protocoles basés sur la topologie hiérarchique ou la construction des clusters

Un routage plat, a un certain nombre d'inconvénients. L'inconvénient primaire est qu'il n'est pas scalable. Pour augmenter la scalabilité du réseau ad hoc, des protocoles hiérarchique ou à cluster, peuvent être utilisés.

Le clustering consiste en un découpage virtuel du réseau en groupes de nœuds géographiquement proches. Ces groupes sont appelés *clusters*. Ils sont généralement identifiés par un nœud particulier, un chef de groupe, aussi nommé cluster-head. La plupart des approches de clustering, sont construits à partir d'une métrique particulière qui permet d'assigner un chef à un nœud. Le cluster étant alors constitué du cluster-head et de tous les

nœuds qui lui sont rattachés. L'idée initiale du routage hiérarchique est de permettre à chaque entité de stocker la totalité des informations de son cluster et seulement une partie des informations concernant les autres clusters. Cela minimise la taille des tables de routage et la quantité de trafic généré.

Le clustering présente d'autres avantages. Il peut faciliter le partage des ressources et/ou la synchronisation au sein d'un cluster et il permet une réutilisation spatiale des fréquences radio pour minimiser les interférences.

De nombreuse solution de clustering ont été proposées, dont La majorité proposent l'utilisation d'une métrique qui permet aux nœuds de se choisir un chef. Cette métrique peut être par exemple l'identifiant ou le degré des nœuds. Une grande partie des solutions de clustering construisent des clusters à 1 seul saut (dit 1-cluster), c-à-d des clusters où chaque nœud est a un saut de son chef de cluster. Les protocoles donnent naissance à des k-clusters (clusters où chaque nœud a au plus k saut de son cluster-head) sont plus récents et plus rares [36].

Exemple de protocoles : CBRP [18], HSR [28], ZRP [27]……

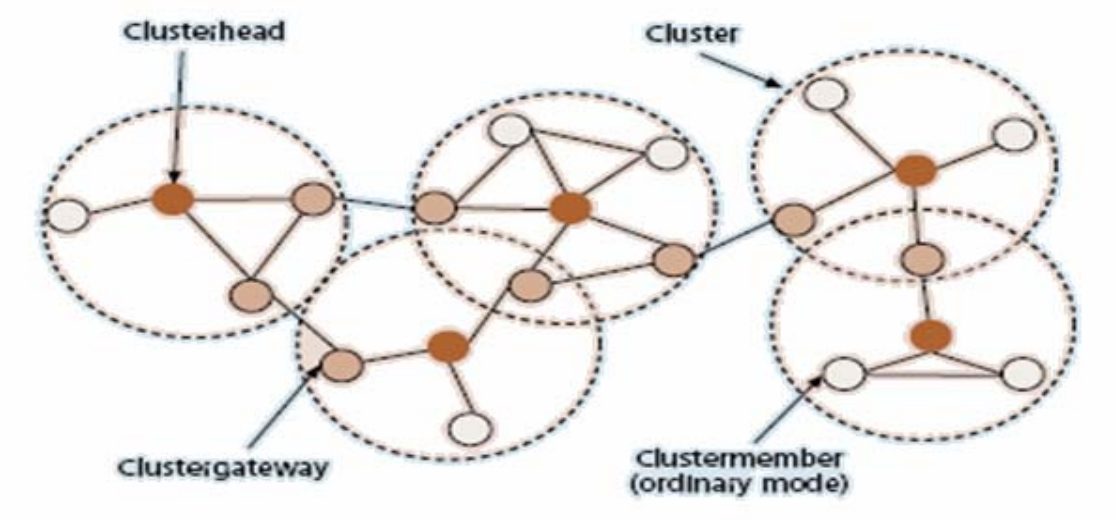


Figure 2.1. Topologie hiérarchique basée sur la construction des clusters

2.3 Protocoles basés sur le positionnement

Les protocoles basés sur le positionnement éliminent certaines limitations des protocoles basés sur la topologie en utilisant des informations additionnelles. Elles exigent des informations sur les positions physiques des nœuds. Généralement, chaque nœud détermine sa propre position par l'utilisation du GPS ou un autre type de service de localisation [37, 38], un aperçu de ces méthodes peut être trouvé dedans [39]. Un service de localisation est employé par l'expéditeur d'un paquet pour déterminer la position de la destination et pour l'inclure dans l'adresse de destination du paquet [41].

La décision de routage à chaque nœud est alors basée sur la position de la destination contenue dans le paquet et la position des voisins du nœud expéditeur. Le routage basé sur le positionnement n'exige pas l'établissement ou l'entretien des itinéraires. Ainsi Les nœuds n'ont pas à stocker ni des tables de routage ni à transmettre des messages de mise ajour des tables de routage. Comme un autre avantage, le routage basé sur le positionnement supporte la livraison des paquets à tous les nœuds dans une région géographique donnée d'une manière normale. Ce type de service s'appelle le geocasting [40].

Exemple de protocoles : DREAM [33], LAR [24]…..

2.4 Protocoles uniformes versus protocoles non uniformes

Un protocole uniforme n'assigne aucun rôle spécial à un nœud. Dans un protocole non-uniforme quelques nœuds peuvent être assignés un rôle spécial, qui doit être exécuté dans un mode distribuée. Les protocoles basés sur le clustering sont non-uniformes.

Exemple de protocole uniforme : AODV [32], DSR [34], ABR [23]……
Exemple de protocole non uniforme : OLSR [21], CBRP [18]…..

2.5 Protocoles basés sur la connaissance de la topologie totale versus réduite

Plusieurs protocoles de routage transmettent l'information de la topologie du réseau, mais pas tous distribuent l'information complète de la topologie.

Il est difficile de classifier les protocoles selon cette caractéristique. En outre même si des informations de topologie complètes sont maintenues dans chaque nœud. Les messages diffusés ne portent que l'information suffisante pour refléter les changements de la topologie totale, et elle ne porte jamais l'information entière de la topologie.

Exemple de protocole qui maintient l'information sur la topologie complète du réseau : OLSR [21], DDR [29]...

Exemple de protocole qui maintient l'information sur la topologie réduite du réseau : FSR [30], ZRP [27]…

2.6 Utilisation de routage par la source

Une classe de protocoles de routage utilise le routage de source, c-à-d, l'envoie de données dépend de la source de message. La source met toute l'information de routage dans l'en-tête du paquet. Les nœuds intermédiaires du chemin utilisent cette information pour faire suivre les paquets. Dans certains cas, les nœuds intermédiaires du chemin peuvent changer l'information de cheminement du paquet à expédier.

Exemple de protocole : CBRP [18], DSR [34]……

2.7 Stratégie de diffusion des messages

Les protocoles de routage dans les réseaux ad hoc peuvent être classés selon la stratégie de diffusion des messages en deux catégories :

- Diffusion complète (full netwide broadcast) : Dans ce type de diffusion les messages de routage vont inonder tous le réseau et chaque nœud intermédiaire doit retransmettre ces messages.

  Exemple de protocole : ABR [23], AODV [32], DSR [34].....

- Diffusion limitée : dans ce type de diffusion les messages de routage vont attendre un nombre de sauts limité à un nombre maximum (TTL time to live).

  Exemple de protocole : AODV [32], LAR [24].....

2.8 Protocoles basés sur la stratégie de sélection de route

La stratégie de sélection de route dans les protocoles de routage est un concept très important, et peut différentier les protocoles et les classifier. Nous donnons dans ce qui suit les stratégies les plus connues et les plus utilisées.

- Qualité du signal : Dans cette stratégie, les protocoles de routage emploie la puissance du signal afin d'estimer la stabilité d'un lien. Si un noeud reçoit un signal de puissance élevée venant d'un noeud adjacent alors que ces deux noeuds sont l'un dans la portée de l'autre et le lien entre eux peut être considéré stable. Ces protocoles essayent de trouver la route contenant des liens stables afin d'avoir une route durable.

  Exemple de protocole : ABR [23], SSA [25]....

- Stabilité de lien : Dans cette stratégie, les protocoles estiment la durée maximale de connexion de deux nœuds, et utilise cette durée comme métrique de choix de route la plus durable.

  Exemple de protocole : DST [31], FORP [35]...

- Plus cours chemin/Etat de lien : dans cette stratégie, les protocoles de routage vont choisir le plus court chemin pour vérifier certaine métriques.

  Exemple de protocole : OLSR [21], TBRPF [22]...

- Vecteur de distances : dans cette stratégie, les protocoles de routage choisissent la route la plus courte en terme de nombre de sauts :

  Exemple de protocole : AODV [32], DSR [34], ZRP [27]...

- direction du routage : le principe est de choisir les nœuds de l'itinéraire qui se trouve sur la direction géographique de la destination.

Exemple de protocole : DREAM [33], LAR [24]...

2.9 Mécanisme de rétablissement de route

Du fait de la forte dynamicité des noeuds dans le contexte des réseaux ad hoc, les ruptures de routes sont beaucoup plus fréquentes et doivent être recalculées très souvent. Ceci provoque évidemment des interruptions de connexions et une perte de qualité au niveau des applications. Il est clair, que les protocoles proactifs de routage n'ont pas besoin d'un mécanisme de rétablissement spécifique, puisqu'ils réagissent aux changements de topologie au cours d'une période courte. Les protocoles réactifs cependant, ont besoin de rétablir les itinéraires qui se rompent.

On peut distinguer trois grandes classes de stratégies de rétablissement :

- Stratégie de rétablissement de route globale

Si un échec de lien était détecté pendant la transmission d'un paquet de la source A à la destination B, le noeud qui a détecté l'échec renverrait un message d'erreur à la source. Puis, la source lance une nouvelle phase de découverte d'itinéraire pour trouver un nouveau chemin entre A et B. Cette phase a besoin de beaucoup de temps pour être accomplie et surcharge le réseau par les messages de routage. Ceci a comme conséquence la perte de largeur de la bande passante et minimise les performances globales du réseau (Figure 2.2) [46].

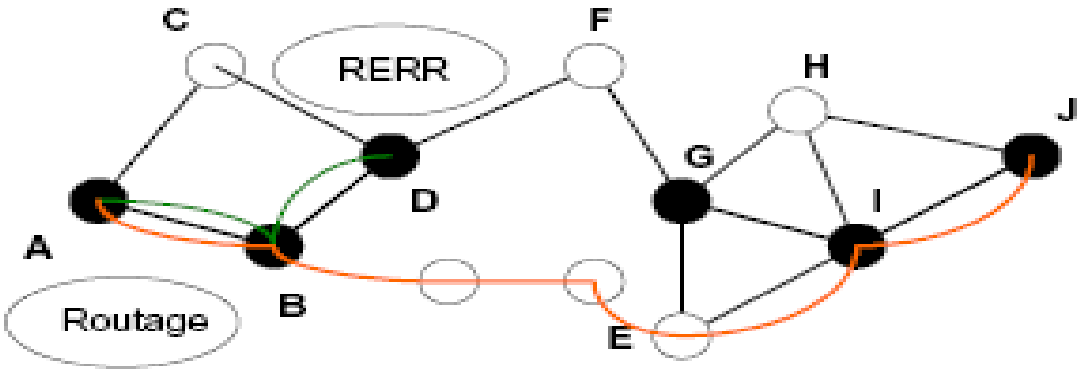


Figure 2.2 Stratégie de rétablissement de route globale.

Exemple de protocole : SSA [25]...

- Stratégie de rétablissement de route locale

Quand un noeud détecte un échec de lien, il n'envoie pas systématiquement un message d'erreur à la source. En premier lieu, il essaye de réparer l'itinéraire lui-même. Si cette première tentative échoue, il envoie un message d'erreur à la source. Cette réparation locale d'itinéraire à comme avantages d'être rapide et de consommer peu de largeur de bande ainsi elle améliore les performances globales du réseau. Dans la figure 2.3 nous donnons un exemple de la réparation locale d'itinéraire.

L'étape (a) représente l'itinéraire avant l'échec des liens constituant l'itinéraire. Dans l'étape (b), le nœud D détecte un échec de lien qui le relie avec le nœud G. Dans l'étape (c), une procédure de réparation locale d'itinéraire est lancée. Le nœud D annonce un paquet de demande d'itinéraire (RREQ) qui est propagé à travers le réseau. Quand il reçoit ce paquet, J renvoie un paquet de réponse d'itinéraire (RREP) à D. Ce paquet est expédié par I, G et F. le nœud D le reçoit finalement et l'itinéraire est réparé sans informer la source [46].

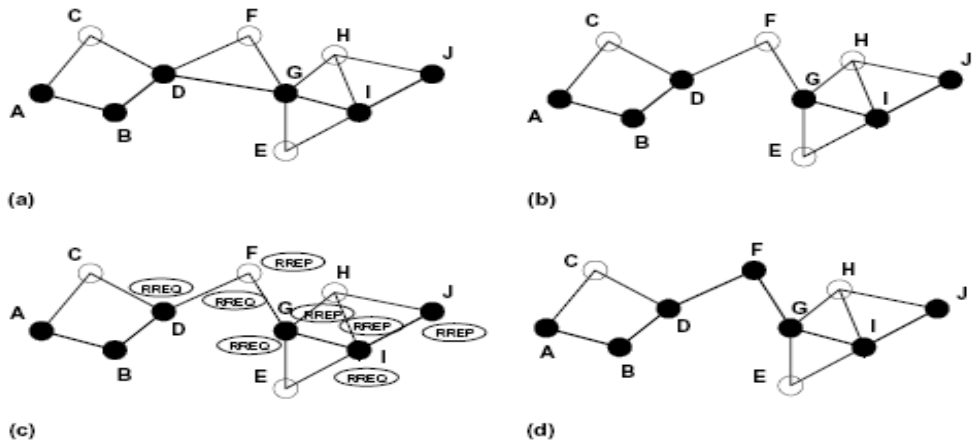


Figure 2.3 Stratégie de rétablissement de route locale

- Utilisation de multi chemins (multipath)

Les protocoles qui utilisent cette stratégie, enregistrent plusieurs chemins alternatifs, au lieu d'enregistrer un seul chemin dans la phase de recherche d'itinéraire. Ces chemins alternatifs sont utilisés en cas de ruptures de route par la source ou aussi par des nœuds intermédiaires. Quand un noeud détecte un échec de lien, il renvoi un message d'erreur à la source ou aux nœuds qu'ils lui précèdes pour choisir un autre chemin alternatif, et la route

sera rétabli dans un délai très court. Nous présenterons dans le dernier chapitre notre proposition qui utilise cette stratégie pour mieux amélioré les performances du réseau et accélérer le rétablissement d'itinéraire (figure 2.4) [46].

Exemple de protocole : AOMDV [42], SMR [43].....

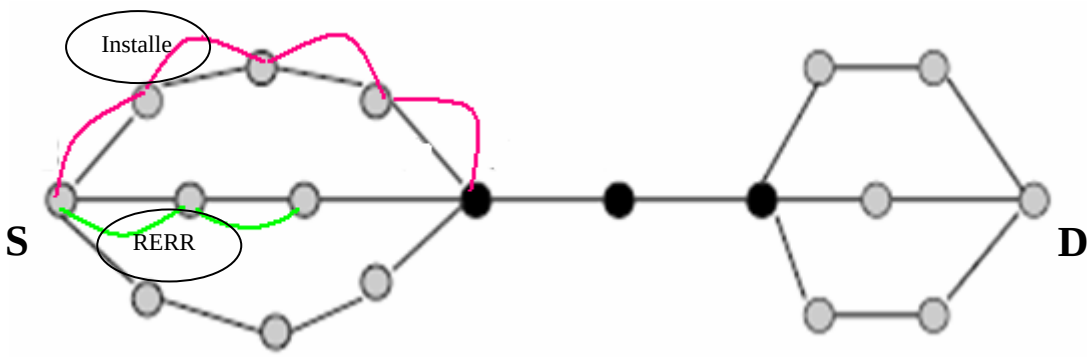


Figure 2.4. Stratégie de rétablissement de route utilisation le multi-chemins (multipath)

2.10 Conclusion

Nous avons présenté dans ce chapitre un état de l'art sur les classifications des protocoles et les caractéristiques spécifiques à chaque classe. Les classifications citées dans ce chapitre n'ont pas de structure sous forme d'arbre, bien que certaines classes aient des caractéristiques communes. D'autres classes sont d'une nature d'opposition.

Dans le chapitre suivant nous présenterons une sélection de protocoles de routage pour les réseaux ad hoc qui présente des exemples pour les classifications citées dans ce chapitre.

Chapitre **3**

Protocoles de routage dans les réseaux Ad Hoc

Les réseaux ad hoc sont parfois définis comme des réseaux spontanés sans fil. Ils réunissent un grand nombre d'objets communicants sans fil, sans infrastructure et tous ces objets peuvent se déplacer. De tels réseaux fonctionnent différemment des réseaux classiques, qui utilisent une dorsale filaire et des collecteurs de trafic pour connecter un certain nombre de réseaux locaux filaires ou sans fil. Les réseaux ad hoc doivent s'auto-organiser pour acheminer le trafic d'un point à l'autre du réseau ad hoc. En plus, un réseau ad hoc ne possède pas d'infrastructures, qu'elles soient fixes ou mobiles. Les fonctionnalités réseaux sont donc intégralement prises en charge par des terminaux, ce qui explique le terme de nœud routeur pour tout terminal. Chaque noeud doit être capable de maintenir une vue partielle ou globale du réseau, qui lui permettra de participer dans la fonction de routage. Sachant que le destinataire potentiel peut se trouver à plusieurs sauts. Les données envoyés par un nœud source, doivent être obligatoirement passer par d'autres noeuds tels que lui pour atteindre le destinataire. Tout noeud doit donc participer de façon collaborative à l'établissement et à la maintenance du réseau. A cet environnement multi-sauts à noeuds hétérogènes s'ajoute la mobilité potentielle de chaque noeud : tout terminal peut se déplacer dans une direction quelconque, obligeant une adaptation dynamique des capacités du réseau.

Les réseaux ad hoc présentent des spécificités fortes. Tout d'abord, les utilisateurs de tels réseaux peuvent être constitués d'ordinateurs de bureau, d'ordinateurs portables, de PDA, et également à terme de téléphones portables ou tout équipement communicant sans fil. De tels types de terminaux engendrent immédiatement des contraintes en terme d'énergie, d'hétérogénéité, de puissance de calcul et de performances matérielles.

Le routage dans les réseaux ad hoc constitue un thème important de recherche. Il faut avant tout trouver un moyen pour router les données dans le réseau de façon efficace. Il faut donc économiser la bande passante, ressource rare en radio, ainsi de minimiser au maximum le nombre de collisions. Il est également indispensable de concevoir des protocoles efficaces dans les cas, où le nombre de participants et leurs mobilités respectives augmentent. Un protocole doit répondre généralement à un certain nombre de contraintes. Il faut donc quantifier le comportement de chaque protocole face à ces différents critères.

Dans le chapitre précédent on a présenté les classifications les plus connues des protocoles de routage. On va présenter dans ce chapitre un état de l'art sur quelques protocoles existants, ainsi que des exemples de protocoles pour chaque classification cités dans le chapitre précèdent.

3.1 Le protocole AODV (Ad-Hoc On-Demand Distance Vector Routing)

3.1.1 Recherche et installation d'itinéraire

AODV est un algorithme distribué qui utilise des algorithmes de vecteur de distance, tels que l'algorithme de Bellman Ford. Quand un itinéraire à une destination est inconnu, AODV crée le paquet de demande de route (RREQ) et l'annonce à ses voisins. Les messages de demande d'itinéraire contiennent les champs suivants :

• L'identification de source (Source address)
• Le nombre d'ordre de source (Source sequence number)
• L'identification d'émission (Broadcast ID)
• L'identification de destination (Destination address)
• Le nombre d'ordre de destination (Destination sequence number)
• Le nombre de sauts (Hop count)

Le nombre d'ordre de source s'incrémente chaque fois qu'une nouvelle demande d'itinéraire s'est produit.

Chaque noeud intermédiaire recevant une demande d'itinéraire enregistre le noeud précédent qui lui a diffusé le message; ceci aide à la création de chemin de retour pour les paquets de réponse. AODV emploie le nombre d'ordre de destination pour maintenir la fraîcheur des itinéraires. Le noeud destinataire ou n'importe quel noeud intermédiaire peut répondre à une demande d'itinéraire. Si un noeud intermédiaire a précédemment appris le

chemin au noeud destinataire, il peut répondre à la demande seulement s'il satisfait la condition suivante :

Le nombre d'ordre de destination localement stocké soit plus haut ou comparable au nombre d'ordre de destination dans le paquet de demande d'itinéraire.

AODV se fonde fortement sur les nombres d'ordre pour éviter le problème du comptage à l'infini, lié aux protocoles de vecteur de distance. La paire d'identification d'émission (broadcast ID) et l'identification de source (source ID) aide à la destruction de toutes les demandes redondantes qui atteignent un noeud.

La destination ou les noeuds intermédiaires répondent par un message de réponse d'itinéraire RREP (Route REPonse) à la source. Les noeuds recevant un message de réponse d'itinéraire stockent l'identification de source (source ID) du noeud qui lui a diffusé le message comme prochain noeud vers la destination afin d'expédier le futur trafic vers cette destination.

Un noeud produisant d'une demande d'itinéraire ou d'une réponse d'itinéraire place le nombre de saut (TTL Time To Leave) à zéro. Ce dernier est incrémenté par chaque noeud intermédiaire avant la rediffusion du paquet de demande d'itinéraire. Cette incrémentation aide le noeud intermédiaire, à déterminer le nombre de saut pour atteindre la source ou la destination à l'aide du chemin courant. Le noeud source recevant un certain nombre de réponses d'itinéraire de différents chemins, emploie le nombre de saut dans les messages de réponse d'itinéraire pour choisir celui qui a la valeur minimale de sauts, comme itinéraire le plus court à la destination [44].

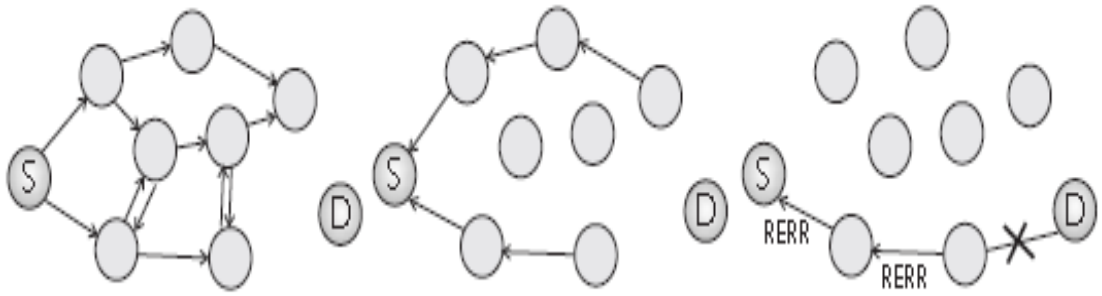
3.1.2 Maintenance d'itinéraire

Une fois où un itinéraire est établi, il doit être maintenu aussi longtemps qu'il est nécessaire. Un itinéraire qui a été récemment utilisé pour la transmission des paquets de données s'appelle un itinéraire actif.

En raison de la mobilité des noeuds, les liens le long d'un chemin sont susceptibles de se casser. Ces cassures sur les liens qui ne sont pas utilisés pour la transmission des paquets de données n'exigent aucune réparation, cependant, des coupures dans des itinéraires actifs doivent être rapidement réparées de sorte que des paquets de données ne soient pas détruites. Quand une coupure de lien le long d'un chemin actif se produit, le noeud d'amont de la coupure (c-à-d, le plus près du noeud de source), marque les itinéraires à chacune de ces destinations dans sa table d'itinéraire qui utilise ce lien comme invalide. Il crée alors un message d'erreur d'itinéraire (RERR). Dans ce message il énumère toutes les destinations qui sont inaccessible dues à la perte du lien. Après avoir créé le message de RERR, le nœud qui a détecter la rupture envoie ce message à ses voisins (seuls qui utilisent ce lien sont concerné par le message RERR). Les nœuds voisins, à leur tour, marque les itinéraires cassés comme

invalide, et envoient leurs propres messages de RERR à leurs voisins. Le message de RERR traverse ainsi le chemin renversé au noeud de source, comme c'est illustré sur la figure 3.1. Une fois que le noeud de source reçoit le paquet RERR, il peut réparer l'itinéraire si l'itinéraire est toujours nécessaire.

Une optimisation est proposée dans le protocole AODV, c'est la réparation locale des coupures de lien dans les itinéraires actifs. Quand une coupure de lien se produit, au lieu d'envoyer un message RERR à la source, le noeud amont de la coupure peut essayer de réparer lui-même le lien localement. S'il réussit, peu de paquets de données sont détruits parce que l'itinéraire sera réparé dans un temps court. Si la tentative de réparation locale n'a pas réussie, un message de RERR est envoyé au noeud source pour l'informer de l'invalidité de l'itinéraire. Pour réparer localement l'itinéraire, le noeud qui doit le réparer augmente le nombre de séquence de destination et envoie un message RREQ à ses voisins. Le TTL doit être calculé, de sorte que le processus de réparation n'écarte pas dans tout le réseau. Le noeud réparateur attend les messages RREP de son message RREQ pour une durée de temps indiquée. Si le message RREP est reçu, alors le chemin est réparé localement [32,47].



a- Demande d'itinéraire b- Propagation de RREP c- Message RERR

Figure 3.1 Recherche, installation et maintenance d'itinéraire dans AODV

3.2 Le protocole DSR (Dynamic Source Routing)

3.2.1 Recherche et installation d'itinéraire

Ce protocole est caractérisé par l'utilisation du routage par la source. C'est à dire, que la source connaît l'itinéraire complet saut par saut jusqu'à la destination. Les paquets de données portent l'itinéraire de source dans l'en-tête de paquet.

Quand un nœud dans le réseau ad hoc essaye d'envoyer un paquet de données à une destination pour laquelle il ne connaît pas déjà l'itinéraire, il lance une procédure de découverte de route pour déterminer dynamiquement un tel itinéraire. La découverte d'itinéraire fonctionne par l'inondation du réseau par des paquets de demandes d'itinéraire (Route REQuest RREQ). Chaque noeud recevant une demande d'itinéraire RREQ, il la rediffuse, à moins que ce soit la destination en question ou s'il a dans son cache d'itinéraires, un itinéraire vers la destination. Un tel noeud répond à la demande avec un paquet de réponse d'itinéraire (Route REPly RREP), qui est routé de nouveau à la source demandeuse. Les paquets de demande d'itinéraires et de réponse d'itinéraire, sont routés de la même façon, c-à-d les paquets contiennent le chemin à suivre. La demande accumule le chemin traversé jusqu'à la destination. La réponse se conduit de nouveau à la source en traversant ce chemin vers la source. L'itinéraire porté par le paquet de réponse est caché par la source pour un futur usage [34].

3.2.2 Maintenance d'itinéraire

Si n'importe quel lien sur un itinéraire actif est cassé, un paquet de signalisation d'erreur (RERR Route ERRor) s'est produit. Ce dernier est envoyé vers la source, ainsi tous les nœuds intermédiaires recevant ce paquet, consultent ses caches, à la recherche des itinéraires qui contiennent le lien cassé. Si tels itinéraires sont trouvés, ils seront supprimés. Une nouvelle procédure de découverte d'itinéraire doit être lancée par la source, si cet itinéraire est toujours nécessaire et aucun itinéraire alternatif n'est trouvé dans la cache [47].

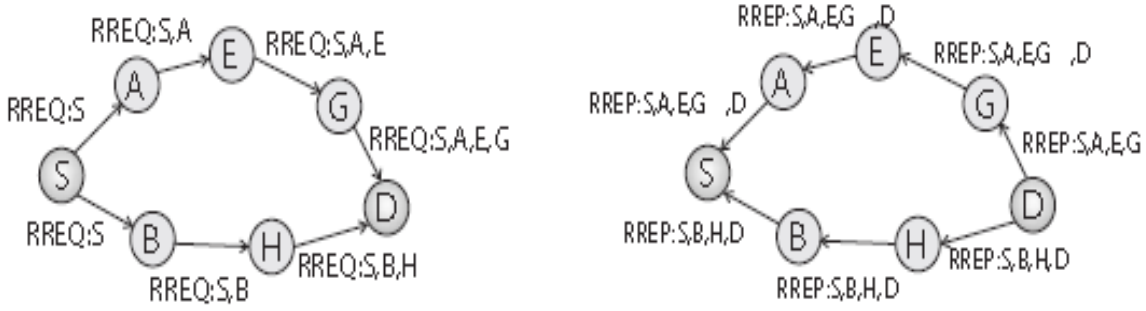
3.2.3 Avantages et Désavantages

DSR soufre de l'utilisation agressive du cheminement de source et de l'utilisation du cache d'itinéraire. Avec le cheminement de source, l'information complète de chemin est disponible et on peut facilement détecter et éliminer les boucles de cheminement sans exiger

n'importe quel mécanisme spécial. Puisque les demandes et les réponses d'itinéraire sont routées par la source. En plus les nœuds intermédiaires recevant ces paquets, peuvent apprendre de nouveaux itinéraires et les enregistrer dans leurs caches pour un futur usage.

DSR utilise plusieurs optimisations comprenant l'écoute passive (promiscuous listening), qui permet aux noeuds qui ne participent pas à la diffusion, de surprendre les transmissions de données en cours à proximités et d'apprendre différents itinéraires exempte de coût. Pour profiter pleinement d'itinéraires cachés. DSR répondent à toutes les demandes atteignant une destination d'une simple demande d'itinéraire. Ainsi la source apprend plusieurs itinéraires alternatifs vers la destination, qui seront utiles dans le cas où l'itinéraire (le plus court) primaire échoue. Avoir accès à plusieurs itinéraires alternatifs minimise les inondations de découverte d'itinéraire, qui est souvent un goulot d'étranglement d'exécution. Ceci peut, cependant, avoir comme conséquence l'inondation du réseau par des paquets de réponse d'itinéraire à moins que des précautions soient prises.

Cependant, l'utilisation agressive des caches d'itinéraire vient avec une pénalité. Le protocole de base de DSR manque des mécanismes efficaces pour purger les itinéraires éventés. L'utilisation des itinéraires éventés gaspille non seulement la largeur de bande précieuse de réseau, par des paquets qui seront par la suite détruites, mais cause également la pollution d'information cachée dans leurs caches quand ils surprennent des itinéraires éventés. Plusieurs études de performances [49, 50] ont prouvés que les caches éventées, peuvent de manière significative baisser les performances, particulièrement dans le cas de la mobilité élevée et/ou aux charges élevées. Ces résultats ont motivé le travail sur des stratégies de cache améliorées pour le DSR [51, 52, 53]. En plus de ce problème, l'utilisation du routage de source augmente les frais généraux du réseau [54].



a- demande d'itinéraire (RREQ) b- installation d'itinéraire (RREQ)

Figure 3.2 Recherche et installation d'itinéraire dans le protocole DSR

3.3 Le protocole OLSR (Optimized Link State Routing Protocol)

3.3.1 Concept de relais Multipoints (MPR Sets)

Le concept de relais multipoints a été introduit par la norme HyperLan 1 [55] du projet Hypercom/INRIA. Le relais multipoint est une technique utilisée pour réduire le nombre de retransmissions redondantes dans le réseau lors de la diffusion des messages. Chaque noeud doit choisir un sous-ensemble de ses voisins à un saut afin de servir de relais pour atteindre le voisinage à deux sauts. L'inondation s'effectue alors de la manière suivante : quand un noeud reçoit le message pour la première fois, il ne le retransmet que s'il est un relais multipoint de celui qui a envoyer le message. Récursivement et en répétant ce processus, le message atteint la totalité du réseau [56]. La Figure suivante illustre un exemple de diffusion en utilisant les relais multipoints. Le calcul d'un ensemble de relais multipoint de taille minimale est un problème NPcomplet [57].

Figure 3.3. Diffusion de messages en utilisant les relais multipoints.

3.3.2 Election des MPR dans OLSR

Le protocole OLSR [21] est un protocole proactif qui adapte un protocole de routage classique d'état de lien pour le routage dans les réseaux ad hoc. Comme tous protocoles proactifs de routage, il emploie les messages périodiques pour mettre à jour l'information de

topologie à chaque noeud. Dans un protocole classique d'état de lien, le paquet d'état de lien inclut la liste entière des voisins avec le coût de la métrique associé à ce lien, de ce fait, ils produisent des grands paquets de contrôles dans le réseau. En outre, ces paquets sont diffusés dans le réseau entier, qui ne répond pas bien aux basses conditions de largeur de bande des réseaux ad hoc sans fil. OLSR optimise le protocole classique d'état de lien, en réduisant les frais généraux de paquet de contrôle et en créant des mécanismes efficaces d'inondation dans le réseau.

Le protocole OLSR utilise le concept de relais Multipoints (MPR) qui minimise la reproduction des paquets de contrôle pour inonder tout le réseau. Chaque nœud, maintient deux ensembles de noeuds, l'ensemble *MPRset* et l'ensemble *MPRselectorset*. Le *MPRset* comprend l'ensemble de noeuds MPR que le noeud courant a choisi, ainsi que le *MPRselector* se compose d'un ensemble de noeuds qui ont choisi le noeud courant comme noeud MPR. Les noeuds MPR agissent en tant que stations de diffusion quand ils reçoivent des données ou destiné aux noeuds parmi son *MPRselectorset*. Le choix des noeuds MPR comme stations de diffusion réduit l'information d'état de lien parce que seulement la connectivité d'état de lien du noeud MPR qui doit être incluse dans les paquets de contrôle d'état de lien. Puisqu'un noeud MPR représente efficacement ses noeuds de *MPRselector*.

Au commencement un noeud commence avec un *MPRset* et un *MPRselectorset* vides. Tous les noeuds voisins d'un seul saut sont considérés comme noeuds MPR. Cet ensemble diminue dans la taille avec le temps pendant que les messages Hello sont reçus. Dans une certaine période de temps, le noeud apprendra tous ses voisins à deux sauts et ses MPR. OLSR emploie trois éléments importants : un élément de sensation voisin, un élément efficace de message de diffusion, et un élément de diffusion de topologie. OLSR utilise une manière de sensation voisine simple pour détecter le statut du lien voisin. Le statut de lien peut avoir trois états possibles : unidirectionnel, bidirectionnel et MPR. OLSR envoie périodiquement des messages Hello contenant une liste de voisins avec l'état de lien qui le relie avec chaque voisin. Un noeud recevant un message Hello pour la première fois d'un voisin, marque le lien comme unidirectionnel dans la table voisine locale et inclut l'identification de ce voisin dans son prochain message Hello. Le voisin recevant ce Hello va trouver son identification de noeud dans le message, ainsi il marque le lien comme bidirectionnel. Puis, pour les messages de Hello suivants de ce voisin, le lien est marqué comme bidirectionnel.

Chaque noeud utilise un algorithme distribué d'approximation pour calculer son *MPRset* [58], ainsi il marque les liens correspondants de noeud comme MPR dans sa table voisine locale. Un MPR marqué par un voisin signifie que le lien voisin est bidirectionnel et également le voisin est un MPR pour le noeud courant. Les messages Hello sont seulement diffusés seulement aux voisins à un saut, et il ne sont pas transmis par relais plus loin. Chaque noeud apprend tous ses voisins de deux saut par le message périodique de Hello. En outre

les messages Hello diffusent les noeuds de transmission *MPRset*. A partir des messages Hello, les noeuds savent s'ils ont été choisis comme MPR. Si c'est le cas, ils placent le noeud correspondant dans son *MPRselectorset* [44].

3.3.4 Construction de la topologie dans OLSR

Pour diffuser l'information de topologie d'état de lien dans le réseau, chaque noeud MPR avec un *MPRselectorset* non vide annonce périodiquement un message de contrôle de topologie (Topology Control TC). Les messages TC contiennent l'identification de noeud MPR et son *MPRselectorset*. Les noeuds MPR recevant ces messages TC les retransmettrent une autre fois. En utilisant l'information de topologie obtenue à partir des messages de TC, les noeuds MPR peuvent calculer le chemin le plus court à chaque noeud dans le réseau et former la table de routage. Les itinéraires dans OLSR contiennent toujours des noeuds de MPR comme transitaires. Par conséquent, OLSR ne construit pas toujours le chemin le plus court, mais garantit un chemin à la destination.

3.3.5 Avantages et Désavantages

OLSR s'exécute bien dans un réseau fortement dense avec le mouvement sporadique des noeuds. Cette caractéristique peut être attribuée à OLSR étant un protocole proactif et ayant des itinéraires toujours disponibles. L'avantage d'OLSR est qu'il réduit au maximum les informations de contrôle. Bien qu'OLSR fournisse un chemin de la source à la destination, ce n'est pas nécessairement le chemin le plus court, parce que chaque itinéraire contient les noeud MPR.

3.4 Le protocole ABR (Associativity-Based-Routing for Mobile Networks)

3.4.1 Concept d'associativité Tick des noeuds

Ce protocole défini une nouvelle métrique de routage appelée degré de stabilité d'association. L'associativité d'un mobile (Mobile Hote MH) avec ses voisins change lorsqu'il émigre, ainsi le degré de cette associativité peut être identifié par un enregistrement appelé l'associativité Tick. Dans ABR [23], le chois des routes est basé sur ce qu'on appelle les états d'associativité des noeuds. Les noeuds du réseau ad hoc génèrent périodiquement des signaux

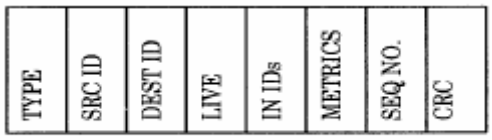
de contrôle afin de montrer leur existence par rapport aux autres noeuds et met à jour constamment ses associativités Tick's avec ses MH's aperçus dans son voisinage. Il existe une période de stabilité, où le MH passera une certaine période dormante dans une cellule avant de commencer à se déplacer encore, ou bien après un seuil $A_{threshold}$ d'associativité Tick's le mobile émigre.

On dit que le MH est dans l'état de stabilité si les Tick's d'associativité sont élevés. Ce dernier constitue le critère principal de choix des liens stables pour réaliser le cheminement dans ce protocole. Les Tick's d'associativité sont remis à zéro quand les voisins ou le MH lui-même se déplace hors de la proximité, pas quand la session de communication est accomplie

3.4.2 Installation d'itinéraire dans ABR

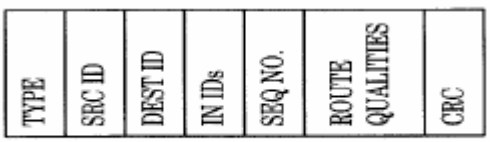
Un noeud désirant un itinéraire, diffuse un message de demande d'itinéraire BQ (broadcast query). Chaque nœud intermédiaire (Intermediate Node IN) qui reçoit la requête vérifiera s'il a précédemment traité le paquet. Si c'est le cas, le paquet de requête sera jeté, autrement le noeud vérifiera s'il est la destination en question. Si ce n'est pas la cas, le IN ajoute son identification dans l'entête du paquet de requête et le rediffuse à ses voisins. Les Tick's d'associativité avec leurs voisins seront également ajoutés, avec touts les autres métriques de cheminement.

Le prochain IN effacera les entrées des Tick's d'associativité des voisins du noeud ascendants et maintiendra seulement ceux concernés par lui-même et son noeud ascendant. De cette manière, le paquet de requête atteignant la destination contiendra seulement les adresses des MH's intermédiaires et leurs Tick's d'associativité, avec les informations d'équilibrage de charge, délai de propagation et l'information de nombre de MH's. Le paquet résultant de BQ est variable dans la longueur dont le format et le suivant :



BQ Control Packet

La destination, et après réception des paquets de BQ. Choisi le meilleur itinéraire et envoie un paquet de RÉPONSE (REPLY) de nouveau à la source. Ceci fait marquer les IN's de l'itinéraire valides. Le paquet REPLY est variable dans la longueur dont le format est le suivant :



REPLY Control Packet

3.4.3 Choix d'itinéraire dans ABR

Les métriques suivantes sont considérées pour le choix d'un itinéraire en ce protocole :

- Durée de vie d'un itinéraire : La longue durée de vie d'un itinéraire est plus importante qu'un itinéraire plus court mais de courte durée qui aura comme conséquence des interruptions fréquentes. Un chemin est de longue durée de vie, si la moyenne d'agrégat des degrés de Tick's supérieur au seuil $A_{threshold}$ et plus grande que la moyenne d'agrégat de Tick's inférieur à $A_{threshold}$.
- Transmission par équilibrage de la charge : La répartition des charges est importante car un unique MH particulier ne devrait être seulement chargé pour soutenir beaucoup de fonctions de relais. Ceci allège également la possibilité de congestion de réseau.
- Le plus cours chemin : Après l'exécution des deux métriques précédentes, cette métrique est appliquée sur les itinéraires résultants.

3.4.4 Maintenance d'itinéraire dans ABR

La phase d'entretien d'itinéraire effectue les opérations suivantes : (a) découverte d'itinéraire partielle, (b) itinéraire invalide (c) mise à jour d'itinéraire valide (d) et nouvelle découverte d'itinéraire (le plus mauvais cas). Ces opérations peuvent être appelées par différents types de mouvements de MH que ce soit des mouvements simultanés des MH's soit des mouvements individuel d'un noeud. Les figures suivantes illustre ce phénomène

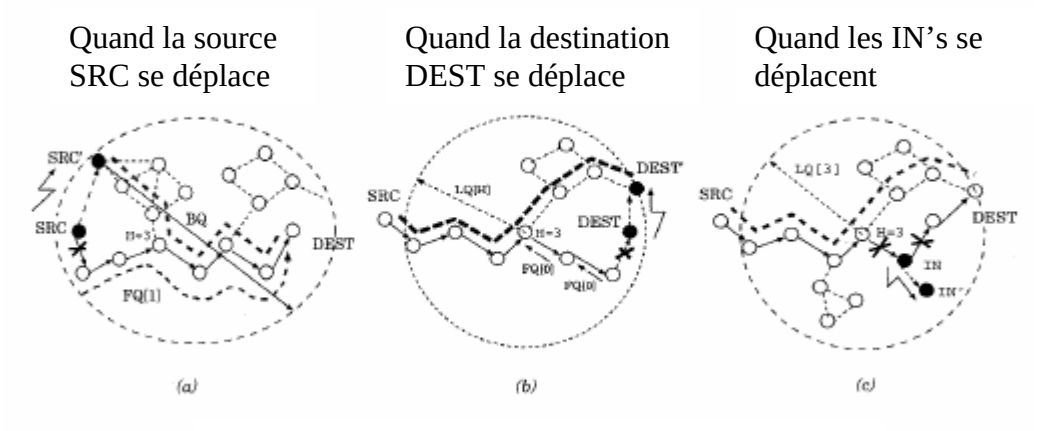


Figure 3.4 Maintenance de route dans ABR

3.5 Le protocole SSA (Signal Stability-Based Adaptive Routing)

SSA [25] est un protocole réactif, il essaye de trouver le chemin le plus stable et de longue durée de vie en utilisant la qualité du signal reçu.

3.5.1 Concept de lien Stable (métrique de stabilité)

Le choix d'itinéraire dans SSA est basé sur deux concepts :
1- La puissance du signal : Le critère de puissance de signal permet au protocole de différencier entre les canaux forts et faibles en calculant la puissance moyenne de signal à laquelle les paquets sont échangés périodiquement entre les noeuds.
2- La stabilité d'endroit : Le critère de stabilité d'endroit permet au protocole de choisir un canal fort qui a existé pendant une plus longue période.

3.5.2 Les Modules du protocole

Fonctionnellement, le protocole SSA se compose de deux protocoles, le protocole de diffusion FP (Forwarding Protocol) et le protocole de cheminement dynamique DRP (Dynamic Routing Protocol), qui utilise l'interface du module de gestion de périphérique. Cette interface est responsable de faire disponible aux protocoles de cheminement l'information de puissance du signal à partir du dispositif.

Deux tables sont maintenues pour permettre le cheminement dans SSA: la Table de stabilité de signal SST (Signal Stability Table) et la table de routage RT (Routing Table).

Chaque mobile envoie un message d'identification à ses voisins à chaque durée de temps. Chaque mobile recevant ce message, enregistre la force à lequel il a été reçu dans la Table SST. Chaque mobile classifie également ses voisins comme fortement connectés SC (strongly connected) ou faiblement connectés WC (weakly connected). Le SST a également une colonne, CLICKS, pour enregistrer le nombre de fois où les messages ont été sans interruptions reçus avec une forte puissance de chaque mobile voisin.

Le DRP met à jour le SST et la RT et fait passer les types appropriés de paquets au FP. Le FP alors passe le paquet à la couche application, ou diffuse le paquet via le module de gestion de périphérique au prochain noeud. Toutes les transmissions sortent par l'intermédiaire du FP, et toutes les entrées se reçoivent par le DRP. Cette division simplifie le protocole en exportant une interface simple pour les paquets sortants et en séparant également hors du filtre, les paquets entrants.

Le paquet SSA comporte les champs suivants :
DA : adresse de la destination ; SA : adresse de la source ; SEQ : numéro d'ordre unique donné par la source, utile pour des recherches d'itinéraire; TTL : employé pour éliminer les paquets incorrectes faisant une boucle dans le réseau ; TYPE : distingue les messages parmi : UNICASTDATA, FLOODDATA, ROUTESEARCH, ROUTEREPLY, ERREUR, ARASE ; PREF : permet à un mobile qui a lancé une recherche d'itinéraire d'indiquer la qualité de l'itinéraire désiré parmi STRONGLINKONLY ou NOPREFERENCE ; LEN : la longueur du paquet entier, CRC : est le checksum.

Les tables SST et RT sont maintenus par le module DRP. À la réception d'un paquet du module de gestion de périphérique, DRP déchiffre le type de paquet, met à jour les tables, modifie certains champs d'en-tête, et enfin les faite passer au FP.

La fonction de mise a jour de la table SST est comme suit :

SS cumulative = α x SS cumulative + (1 - a) X SS

SS cumulative est la quantité enregistrée dans le SST, et le SS est la valeur de la force moyenne du signal pour le paquet fourni par le module de gestion de périphérique. α est une constante expérimentalement déterminée.

Périodiquement, un processus asynchrone fonctionne sur la SST comparant $SS_{cumulative}$ à une quantité expérimentalement déterminée $SS_{threshold}$ pour classifier les noeuds en SC ou en WC. Un noeud mobile qui montre une forte puissance de signal pour des CLICK consécutifs de $CliCk_{Sthreshold}$ sera classifié en SC et s'ajoute à la RT.

3.5.3 Recherche et Installation d'itinéraire

La source produit et annonce un paquet de recherche d'itinéraire avec une partie vide de données, un nombre de séquence unique SEQ, et le champ TYPE par ROUTESEARCH. La source peut choisir la valeur appropriée du champ PREF selon les besoins de l'application ou du protocole des couches supérieures. Elle choisi STRONGLINKONLY pour le premier essai et NOPREFERENCE pour toute nouvelle tentative suivante de demande après l'expiration de TIMEOUT. Si, après plusieurs tentatives, aucun itinéraire satisfaisant n'a été trouvé, le paquet de données est diffusé et atteint la destination par l'intermédiaire de l'inondation. Une approche alternative à l'inondation serait de rapporter une exception à l'application qui a produit le paquet.

Chaque noeud intermédiaire détermine si le paquet devrait être diffusé ou détruit (s'il été précédemment vu par ce noeud ou si le TTL a expiré). Le noeud recevant les paquets de recherche d'itinéraire, enregistre dans une table, la paire d'adresse de source (SA) et le nombre de séquence (SEQ). Il modifie le paquet de recherche d'itinéraire en ajoutant son adresse à la liste des nœuds et rediffuse le paquet.

Quand la destination prévue reçoit le paquet de demande d'itinéraire, une réponse d'itinéraire et envoyé à la source le long du chemin renversé de l'itinéraire contenu dans le paquet de demande. Chaque noeud d'intermédiaire, reçoit le paquet de réponse d'itinéraire, installe dans sa RT tous les chemins possibles qui sont impliqués par la réponse d'itinéraire et envoie le paquet au prochain nœud.

3.5.4 Entretien d'itinéraire

L'entretien d'itinéraire est déclenché par un nœud qui a des données à envoyer, et qui a trouvé le lien cassé. Il envoie un message d'erreur ERROR à la source pour l'informer par la rupture du chemin. Le nœud source lance une recherche d'un nouvel itinéraire et envoie un message d'effacement ERASE pour annuler l'itinéraire rompu.

3.6 Le protocole TORA (Temporally Ordered Routing Algorithm)

Ce protocole présenté par Park et Corson dans Park1997 [20], appartient à la famille des algorithmes de cheminement d'inversion de lien (Link Revesral). Il a été conçu principalement pour minimiser l'effet des changements de la topologie qui sont fréquents dans les réseaux ad hoc. Le protocole TORA s'adapte à ces changements en stockant plusieurs chemins vers une même destination. Ce qui fait que beaucoup de changements dans la 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 réduit de noeuds. Cet ensemble représente les nœuds proches du lieu de l'occurrence du changement de la topologie.

Comme c'est le cas pour tous les protocoles réactifs, les chemins sont crées est installés lors du besoin. TORA est basé sur le principe des algorithmes qui essaient de maintenir la propriété appelée "orientation destination" des graphes acycliques orientés (ou DAG : Directed Acyclic Graph) [59]. 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 dernier cas, l'algorithmes utilise le concept d'inversement de liens. Ce concept assure la transformation du graphe précédent, en un graphe orienté destination durant un temps fini.

TORA maintient également un DAG au moyen d'un quintuple commandé avec l'information suivante :

t : période d'un échec de lien
oid : identification de créateur
r : un bit indicateur de réflexion 0= le niveau original 1= le niveau reflété
d : le paramètre d'ordre de propagation
i : l'identification de nœuds.

Cette nouvelle notion est utilisée dans l'orientation des liens du réseau. Un lien est toujours orienté vers le noeud qui à la plus grande valeur de paramètre d'ordre de propagation, vers le noeud qui a la plus petite valeur du paramètre d'ordre de propagation. Les concepts de taille (valeur du paramètre d'ordre de propagation) et d'inversement de liens sont orientés destination, cela veut dire que chaque noeud du réseau, exécute une copie logique indépendante de l'algorithme TORA pour chaque noeud destination.

3.6.1 Recherche et Installation d'itinéraire

Le processus de création d'itinéraire de la source à la destination, va créer un DAG orienté destination. Initialement la taille de tous les noeuds est initialisé à NULL (indéfinie) excepte la taille de la source qui est initialisée a 0.

La source diffuse un paquet de demande d'itinéraire QRY (query) qui contient l'Id de la destination. Un noeud recevant un paquet de QRY fait un de ce qui suit :

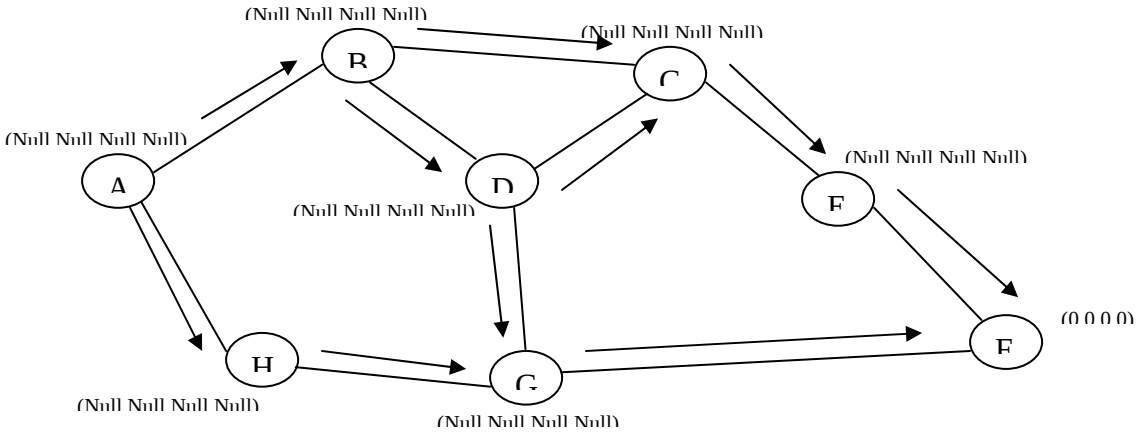
- Si le paquet a une taille indéfinie (NULL), il rediffuse le paquet à ses voisins.
- Un noeud qui a une valeur de taille différente de NULL, répond par l'envoi d'un paquet UPD (update) qui contient sa propre taille.

Un noeud recevant un paquet UPD, affecte la valeur de taille contenant dans le paquet reçu plus un, à sa propre taille, à condition que cette valeur soit la plus petite par rapport à celles des autres voisins.

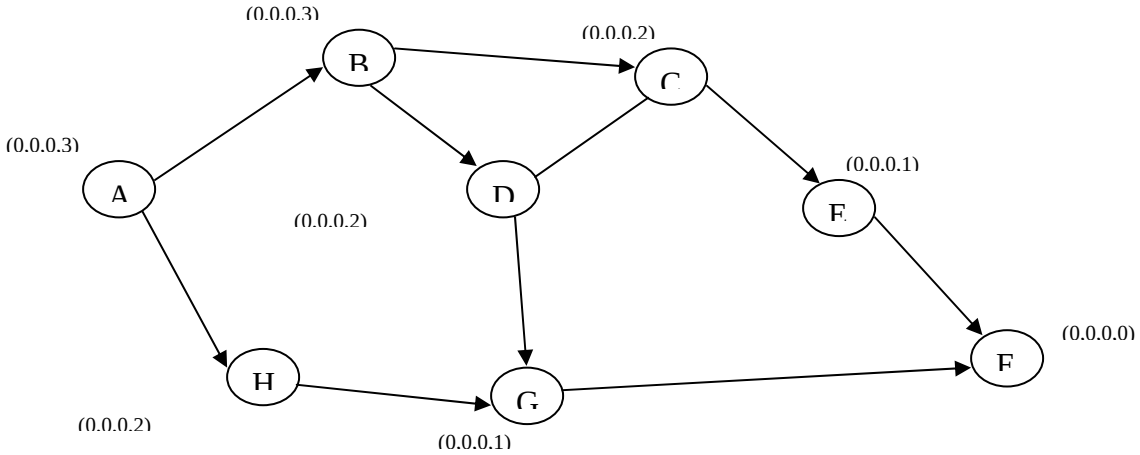
Dans l'exemple de la figure suivante le nœud A veut communiquer avec le nœud F, il propage le message QRY comme il est montré sur le schéma suivant (figure 3.5 (a)). Le nœud F est après réception du message QRY répond par le message UPD (figure 3.5 (b).

3.6.2 Maintenance d'itinéraire dans TORA

Les itinéraires crées peuvent être rompus à cause de la mobilité des nœuds, ainsi une procédure de maintenance doit être effectué afin de rétablir les chemins rompus. Dans ce dernier cas, un nouveau DAG de la source vers la même destination va être crée. Quand un noeud 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 noeud i ajuste sa taille pour qu'elle prenne la valeur maximale des tailles de ses noeuds voisins. Le noeud i, transmet par la suite un paquet UPD contenant la nouvelle taille. Par conséquent tous les liens, vont être orienté du noeud i vers ses voisins, car la taille de i est devenue la plus grande. 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. Des études de performance ont montré quelques avantages et inconvenants de ce protocole [60, 61].



(a) Propagation du message QRY



(b) création du DAG par la propagation des messages UPD

Figure 3.5 Recherche et installation d'itinéraire dans TORA

3.7 Le protocole POWER (POwer and Link failure aWare rEliable Routing)

Le protocole POWER [62] défini le concept de nœud fiable qu'il est soit placé dans un environnement moins dynamique ou qu'il se déplace avec ses voisins. Un itinéraire fiable se compose seulement avec des nœuds fiables.

3.7.1 Concept de lien stable

Pour réaliser un itinéraire fiable on dénote les quatre notations suivantes :

a) LF (Link Failure) :

Ce concept défini la fréquence d'échecs des liens d'un nœud i qui les relient avec ses noeuds voisins dans chaque durée de temps désignée par TIME_WINDOW. Il représente le degré de dynamicité de l'environnement où le noeud est actuellement placé.

$$LF_i = \frac{\#of\ link\ failure\ observed}{TIME\_WINDOWS \times \#of\ neighbors} \times AVG(LF_{neighbors}) \quad (1)$$

$$NLF = \alpha * NLF_i + (1-\alpha) * NLF_{i-1} \quad (2)$$

α et un facteur de lissage entre 0 et 1. Initialement, le LF est à zéro. Le terme AVG ($LF_{neighbors}$) signifie le LF moyen des noeuds voisins. Puisque l'échec du lien d'un noeud est affecté par des noeuds voisins.

b) NS (Node Survival) :

L'échec d'itinéraire n'est pas seulement influencé par la mobilité des nœuds mais aussi par la capacité de batterie, alors, des paramètres concernés par la puissance d'énergie devraient être inclus dans la réalisation d'itinéraire fiable.

La batterie d'un mobile est sensiblement consommée en transmettant des paquets plutôt que d'en recevoir, ou exécutant des traitements tel que la recherche d'itinéraire. Le concept NS est défini pour calculer la probabilité de l'échec d'un nœud. Ce concept est calculé par la formule suivante :

$$NS = \frac{AB}{TIME\_WINDOW \times \#of\ fp} \quad (3)$$

Le fp représente le nombre de paquets expédiés dans l'intervalle de temps TIME_WINDOW, et AB la quantité de batterie disponible (%).

c) NR (Node Reliability) :

La fiabilité d'un noeud augmente si le nombre d'échecs de liens diminue ou une plus grande capacité de batterie soit disponible. D'où on peut définir la fiabilité d'un nœud NR comme suit :

$$NR_i = \frac{NS}{LF} \qquad (4)$$

d) N.P.R.s,d (Normalized Path Reliability):

Ce concept détermine la fiabilité d'un chemin N.P.R.s,d (de la source s à la destination d). Il est calculé par le produit des valeurs des NR des noeuds de ce chemin, comme suit :

$$NPR_{s,d}^{k}(TIME\_WINDOW) = \prod_{i \in k^{th} path} NR_i \qquad (5)$$

3.7.2 Installation d'itinéraire

POWER construit les itinéraires fiables sur demande. Quand une source souhaite établir une connexion à une telle destination dont le chemin est inconnu, elle inonde réseau par le paquet de demande d'itinéraire RREQ. Quand les noeuds autres que la destination, reçoivent les paquets RREQ, ils créent une entrée d'itinéraire < source, destination > et enregistrent les adresses des noeuds voisins afin d'établir le chemin renversé. Chaque noeud ajoute sa valeur de NR aussi bien que son adresse dans la liste de noeud (NL) dans le paquet RREQ et le rediffuse.

Après que la destination reçoive le premier paquet RRREQ, attend une durée de temps en vue d'apprendre d'autres itinéraires possibles. La destination choisit alors l'itinéraire le plus robuste selon un algorithme de choix d'itinéraire parmi les suivants :

a) min_PR :

Choisi simplement l'itinéraire le plus fiable avec les moindres N.P.Rs.d parmi tous les itinéraires instruits.

b) min_least_NR :

Le noeud destinataire choisit l'itinéraire ayant le moindre NR parmi l'ensemble des plus petit NR, qui se compose des paires < moindre valeur de NR sur chaque itinéraire, ID d'itinéraire >.

c) PR_Com et NR_Com :

Complètent le min_PR et min_least_NR. En général, chaque destination essaye seulement d'établir un seul itinéraire. Cependant, dans les réseaux fortement dynamiques, il est très difficile de garantir la livraison réussie de données. Pour cela, POWER établie des chemins multiples entre la source et la destination c-à-d en utilisant un seuil d'acceptation θ, tous les itinéraires qui vérifient ce seuil sont acceptés comme des itinéraires alternatifs en cas d'échec. La valeur du seuil θ est déterminée soit d'une façon manuelle ou d'une façon à organisation automatique (self-organizing). La façon à organisation automatique, est basée sur l'agrégation d'histoire de min_PR ou min_least_NR observé sur chaque membre de groupe.

La destination et après le choix d'itinéraire robuste, répond par un message RREP sur le chemin choisi (dans le sens inverse du chemin).

3.7.3 Maintenance d'itinéraire dans POWER

Durant la connexion des données actives, et quand la valeur du NR d'un noeud intermédiaire est nouvellement mise à jour, le noeud intermédiaire enregistre sa valeur de NR et sa propre adresse dans les paquets de données. Avec cette information, la destination peut surveiller le changement du P.R. Si la destination observe la croissance excessive des P.R. sur le chemin courant, elle inonde le réseau par le paquet RREQ pour atteindre la source et l'informe par l'expiration du chemin dans un temps court. Quand la source reçoit le paquet RREQ, elle installe un chemin alternatif plus stable.

Dans le cas où les noeuds intermédiaires détectent l'échec de lien avant que l'itinéraire expire, ils renvoient immédiatement les paquets de données reçues à la source. Ces nœuds intermédiaires envoient aussi le paquet RREQ à la source afin de l'informer de l'invalidité de l'itinéraire. Si la source a déjà établi le chemin alternatif avant l'échec de lien, le paquet d'erreur d'itinéraire est ignoré. La source retransmet les paquets retournés de données du point échoué aussi bien que de nouveaux paquets de données sur le nouvel itinéraire.

3.8 Le protocole FORP (Flow Oriented Routing Protocol)

3.8.1 Concept de Durée d'expiration d'un Lien

Le protocole FORP [35] utilise une nouvelle métrique qui est le temps d'expiration de lien LET (Link Expiration Time). Avec ce concept on va prédire la durée de temps que deux nœuds voisins, restent toujours l'un à la portée de l'autre. Pour le calcul de cette métrique, plusieurs paramètres doivent être fournis par un système de localisation tel que GPS.

Le calcul de l'estimation de cette durée LET est comme suites :

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

Où

$a = v_i \cos \theta_i - v_j \cos \theta_j$

$b = xi - xj$

$c = v_i \sin \theta_i - v_j \sin \theta_j$

$d = y_i - y_j$

v : vitesse du noeud, θ l'angle de direction, (x,y) position géographique du nœud. Ces paramètres sont fournis par un système de localisation.

3.8.2 Concept de Durée d'expiration d'une route

Le temps d'expiration d'une route RET (Route Expiration Time RET), où bien la durée pendant laquelle route reste valide est représentée par le minimum des LET's des liens constituant cette route.

3.8.3 Recherche et Installation d'itinéraire

Dans ce protocole si un noeud source veut communiquer avec un autre nœud (la destination), et comme tous les protocoles réactifs, il lance une recherche d'itinéraire dans le réseau par le paque de demande Flow-REQ. Ce message contient un numéro de séquence, l'ID source, l'ID destination, et la liste des Id's des nœuds déjà parcourus (qui ont déjà reçu le message Flow-REQ). Si un nœud intermédiaire reçoit le paquet Flow-REQ et s'il trouve qu'il a déjà reçu un paquet avec le même couple (source, destination) et le même numéro de

séquence, il rejette le paquet. Autrement il insert dans le paquet son Id et le LET entre lui et le nœud qui à lui diffuser le paquet, et il retransmet le paquet à ces voisins. Ce processus se Réitère jusqu'à que le paquet Flow-REQ atteint la destination.

La destination et après réception des paquet Flow-REQ, calcule la durée d'expiration des routes RET en utilisant les LET des lien contenus dans les paquets Flow-REQ. De plusieurs paquets Flow-REQ reçus, la destination choisit celle qui a la plus grande valeur de RET. La destination envoie ensuite le paquet Flow-SETUP sur le chemin choisi. Les nœuds intermédiaires vont installer la route après réception de message Flow-SETUP.

La figure 3.6 montre un exemple de la procédure de recherche et d'installation de route dans ce protocole. Le nœud source A veut communiquer avec le nœud F, il lance une procédure de découvert de chemin par la diffusion du message Flow-REQ, chaque nœud intermédiaire B, C, D, E reçoit ce message ajoute au paquet Flow-SETUP son adresse et le LET entre lui et le nœud qui lui a diffusé le paquet, et rediffuse le paquet. Le nœud F et après réception des paquets Flow-REQ, calcule les valeurs RET de chaque itinéraire. Dans l'exemple suivant, le RET = 4 pour le chemin ABDEF et pour le chemin ABCEF le RET = 3. On peut bien constater que le chemin ABDEF a la plus grande valeur de RET, d'où ce chemin qui sera élu par la destination (figure 3.6 (a)). Dans la fig3.6 (b) la destination envoie un message Flow-SETUP sur le chemin ABDEF.

3.8.4 Maintenance d'itinéraire dans FORP

Durant la communication les nœuds intermédiaires ajoutent leurs nouvelles valeurs LET aux paquets de données. La destination et après réception des paquets de données recalcule le RET de la route. Lorsque la destination s'aperçoit que la durée d'expiration de la route atteinte un seuil critique "Tc", un message Flow-HANDOFF est diffusé par la destination de la même manière que le message Flow-REQ. Une fois que la source reçoit le message Flow-HANDOFF, elle détermine la meilleure route en se basant sur les informations contenues dans le message Flow-HANDOFF (RET, nombre de sauts, etc). La source envoie par la suite un message Flow-SETUP pour installer la nouvelle route de la même manière qu'auparavant, à la différence que cette fois ci, le message Flow-SETUP est envoyé par la source vers la destination. Dans FORP, le seuil critique Tc est définit par Tc = RET - T_d où T_d est le délai de transfert du dernier paquet. Ce calcul permet au protocole de s'adapter aux changements de charge dans le réseau.

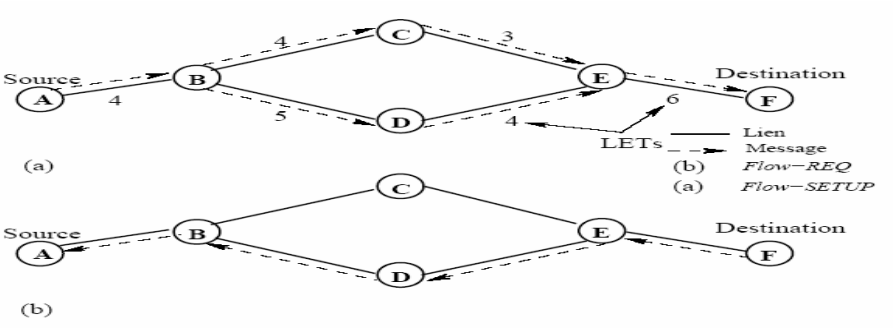


Figure 3.6. Recherche et installation d'itinéraire dans FORP

3.9 Le protocole TBRF (Topology Broadcast Based On Reverse-Path Forwarding)

Le TBRPF [22] est un protocole proactifs qui a pour but de réduire au maximum les frais généraux du réseau et d'utiliser efficacement l'inondation des paquets de contrôle dans le réseau. Dans TBRPF, chaque noeud maintient un arbre couvrant de tous les noeuds à la source. Les arbres couvrants sont constitués par le chemin le plus court de tous autres noeuds à la source. Les messages annoncés par la source sont diffusés seulement dans le chemin renversé le long de l'arbre couvrant précédemment produit. Les noeuds parents annoncent le trafic de contrôle commençant de la source u à tous ses enfants. En d'autres termes, les noeuds non-feuilles de l'arbre de source, rediffusent le trafic de contrôle parvenu de ses parents (ex : la figure suivante).

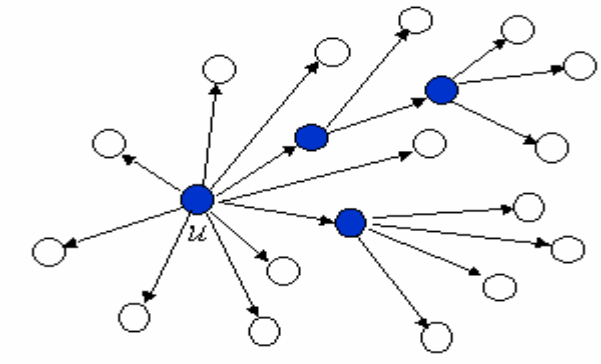


Figure 3.7 Exemple d'illustration d'arbre couvrant dans TBRF

3.9.1 Découverte de voisinage et Construction d'arbre de cheminement

TBRPF contient deux modules, un module de découverte de voisins et un module de cheminement. Le module de découverte de voisins est unique à TBRPF dans le sens qu'il utilise plusieurs types de message HELLO, qui rapportent seulement les changements de statut de lien. TBRPF annonce périodiquement des messages HELLO. Ces messages contiennent trois catégories de listes : demande voisin, réponse voisin et voisin perdu. Chaque message HELLO contient également un nombre de séquence incrémental HSEQ. La première fois que le noeud A détecte un nouveau voisin B, il crée une entrée de lien unidirectionnelle pour le noeud B dans sa table voisine. Le suivant message Hello envoyer par le nœud A, catégorie le noeud B dans la liste de demande voisine. Le noeud B, après sa réception du message Hello, crée une entrée semblable dans sa table voisine pour le noeud A et place le noeud A dans sa liste de réponse voisine dans le message Hello suivant. Ainsi le noeud A et B déterminent que la connectivité est bidirectionnelle et mettent à jour ses tables voisines locales en conséquence.

TBRPF inclut chaque changement de statut de son voisin dans au moins trois messages HELLO consécutifs pour assurer aux voisins l'enregistrement du changement du statut de lien. Les messages Hello contiennent toujours une liste de demande de voisines, même lorsqu'il n'y a aucun changement de lien, pour confirmer aux voisins le statut de lien du noeud de source. Si un noeud ne reçoit plus de message Hello d'un voisin, il marque le lien avec ce voisin comme perdu dans sa table locale de voisinage et rapporte le changement dans ses prochains messages Hello.

De cette façon chaque nœud maintien une route vers tous les nœuds du réseau.

3.9.2 Maintenance d'itinéraire dans TBRF

Dans la figure 3.8 si le lien (I, J) échoue le nœud I sélectionne le nœud R comme nouveau parent pour la racine U et informe le nœud K de l'échec du lien. Le nœud K à son tour sélectionne le nœud R comme parent pour la racine d'arbre de source U. Le nœud K répond au nœud I par un message Update généré par le noeud U avec un nombre de séquence plus grand que le sien et la même chose pour le nœud K.

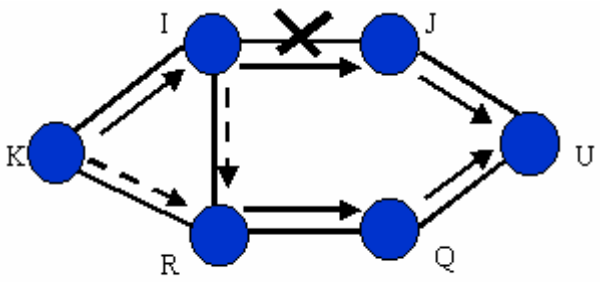


Figure 3.8 Maintenance d'itinéraire dans TBRF

3.10 Le protocole CBRP (Cluster Based Routing Protocol)

L'idée de ce protocole est de diviser le réseau en groupes ou clusters et chaque groupe élu a un représentant ou chef de groupe pour la communication avec les autres noeuds des autres groupes [18].

3.10.1 Principe de formation de groupe

Un noeud p qui n'a pas de statut (indéfini) (i.e. qui n'est ni membre ni représentant de groupe), diffuse périodiquement un message Hello pour s'identifier. Si un chef de groupe (cluster head) reçoit ce message il lui répond par l'envoie d'un message de réponse. Le nœud qui a le statut indéfini et après réception de la réponse d'un chef de groupe change son état en membre. Dans le cas où le nœud ne reçoit aucune réponse dans une durée de temps (durée prédéfinie) et s'il a au moins un lien bidirectionnel avec ses voisins, il se déclare lui-même comme chef de groupe. Dans le cas contraire il reste dans l'état indéfini et rediffuse toujours des messages Hello en attente de réponse ou a l'existence de lien bidirectionnelle avec l'un de ses voisins. En générale les réseaux ad hoc sont caractérisés par la forte mobilité se qui implique que l'attente des nœuds dans l'état indéfini est très court.

Chaque nœud maintien une table des voisins qui contiennent les colonnes suivantes :

Neighbour ID : L'Id du voisin ; *Neighbour Status* : le statut de ce voisin c-à-d Cluster-Head ou Member ; *Link status* : le statut de lien entre les deux nœud c-à-d unidirectionnel ou bidirectionnel.

Les chefs de groupe maintiennent en plus une table des groupes adjacents. Une entrée dans cette table est associée à un groupe voisin. Cette table contient l'identificateur du groupe et l'identificateur du noeud de liaison (Gateway) à travers lequel le groupe peut être atteint (Figure 3.9).

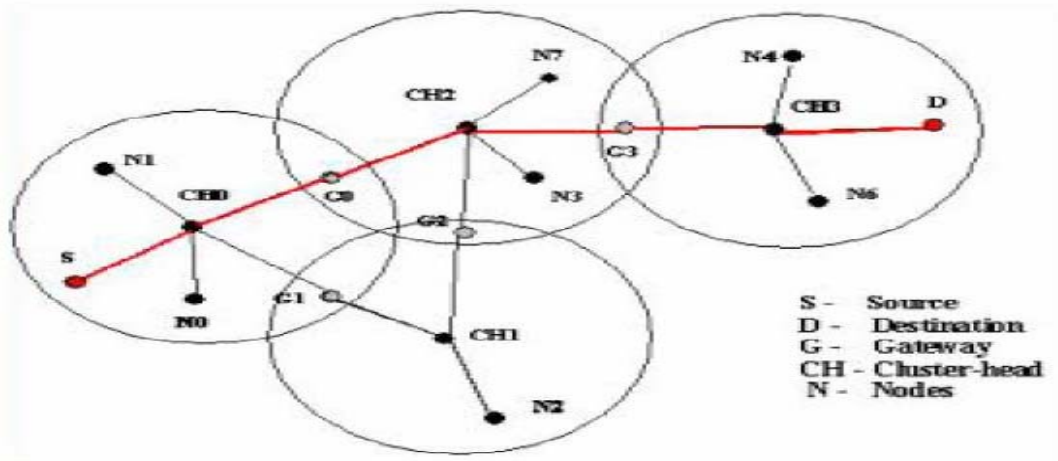


Figure 3.9 Répartition du réseau en clusters et installation de route dans CBRP

3.10.2 Recherche et Installation d'itinéraire

Lorsqu'un noeud souhaite envoyer des paquets de données à une telle destination et qu'il ne connaît aucune route vers cette dernière, son chef de groupe sert de *Proxy* pour la découverte de route. Le chef envoie une demande d'itinéraire RREQ (message unicast) à chacun des groupes adjacents, en choisissant pour chacun un nœud de liaison (Geteway) (un nœud qui appartient simultanément au deux groupes adjacents), qui sert de passerelle vers les chefs de ces groupes. Chaque chef de groupe recevant un RREQ, relaie le paquet vers ses groupes adjacents à moins qu'il soit la destination en question, ou qu'un membre parmi son groupe soit la destination. La destination renvoie un RREP suivant la route inverse contenue dans le paquet RREQ. Chaque chef de groupe tente de calculer une optimisation de routes grâce à sa connaissance locale avant de relayer ce paquet vers la source. Cependant, la route est donnée comme un liste de nœuds intermédiaires. Dans le cas où le noeud source ne reçoit pas de réponse en expirant une certaine période, il envoie de nouveau une requête de demande de chemin.

3.10.3 Maintenance d'itinéraire dans CBRP

Le CBRP propose une solution de reconstruction locale de route. Lors de l'acheminement des données. si un noeud p trouve qu'un noeud suivant n, ne peut pas être atteint, il essaie de vérifier si le noeud n ou le noeud qui vient après n, peuvent être atteints à travers un autre noeud voisin. Si l'un des deux cas est vérifié, les données sont envoyées en utilisant le chemin réparé.

3.11 Le protocole LAR (Location Aided Routing)

Le protocole du cheminement LAR [24] est un protocole réactif, qui utilise les coordonnées géographiques fournis par un système de localisation tel que le système GPS, pour envoyer les messages de demande d'itinéraire directement vers l'endroit de la destination précédemment connu. Le protocole définit deux secteurs : la zone prévue et la zone de demande (figure 3.10). La zone prévue est le secteur dans lequel la destination est le plus susceptible d'être découverte. Pour calculer ce secteur, la source doit connaître un endroit précédent de la destination au temps t_0, aussi bien qu'une évaluation de la vitesse V, auquel la destination voyageait à T_0. Dans le temps courant T_1, la zone prévue peut être calculée comme le cercle du rayon $V*(T_1 - T_0)$. La zone de demande est le secteur dans lequel la demande d'itinéraire de la destination devrait propager. Afin d'avoir la plus grande probabilité de trouver la destination, la zone de demande est définie pour être le plus petit rectangle qui englobe la zone prévue de la destination et le noeud source. La figure 3.10 illustre un exemple de la zone prévue et de la zone de demande.

3.11.1 Recherche et Installation d'itinéraire

Le procédé de base de découverte d'itinéraire dans LAR est semblable à celui d'autres protocoles réactifs de cheminement. Quand une source a besoin d'un itinéraire à une telle destination, elle crée un message de demande d'itinéraire RREQ pour cette destination.

Si la source avait récemment un itinéraire à la destination, alors la source calcule la zone prévue et la zone de demande, et place les coordonnées de la frontière de zone de demande dans le message de RREQ. Si la source n'a aucune information précédente sur la destination, alors il ne peut pas calculer la zone prévue et la zone de demande. Dans ce cas-ci, l'algorithme ce transfère sur l'inondation de base.

Deux approches peuvent être utilisées :

- Dans la première approche, le noeud source définie une région circulaire dans laquelle la destination peut être localisée. La position et la taille de la région, sont estimées en se basant sur : La position de la destination, telle qu'elle est connue par la source, l'instant qui correspond à cette position et la vitesse moyenne du mouvement de la destination.
- Dans la deuxième approche, le noeud source calcule la distance qui le sépare de la destination, et l'inclut dans le paquet de requête de route. Ce dernier est envoyé par la suite aux noeuds voisins. Quand un noeud reçoit le paquet de requête, il calcule la distance qui le sépare de la destination, et la compare avec la distance contenue dans le paquet reçu. Dans le cas où la distance calculée est inférieure ou égale à la distance reçue, le noeud rediffuse le paquet reçu. Lors de la diffusion, le noeud met à jour le champ de distance avec sa propre distance qui le sépare du noeud destination.

Dans les deux méthodes, si aucune réponse de route n'est reçue en dépassant une certaine durée de temps, le noeud source rediffuse une nouvelle requête de route en utilisant une diffusion pure.

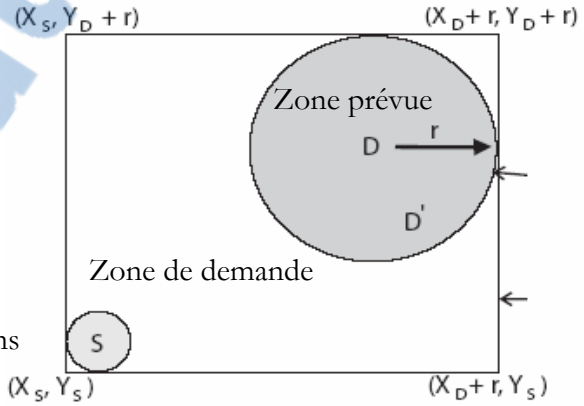


Figure 3.10. Zones de requête dans LAR

3.12 Le protocole DREAM (Distance Routing Effect Algorithm for Mobility)

C'est un algorithme proactif basé sur les informations de localisation géographique des unités mobiles [33]. Chaque nœud diffuse proactivement des messages de contrôle contiennent ses propres informations de localisation fournies par un système de localisation tel que le GPS. Chaque nœud maintient une table de localisation pour enregistrer les informations de localisation (Location Table LT) des autres nœuds. La distance influe dans cet échange, du fait que les messages de contrôle sont envoyés fréquemment aux noeuds les plus proches. Ainsi le protocole s'adapte à la mobilité des nœuds du réseau par le contrôle de mise à jour de fréquence qui se base sur les vitesses des mouvements.

3.12.1 Choix d'itinéraire

Chaque nœud maintient une table de localisation LT pour l'utiliser dans le cas ou un nœud a besoin de communiquer avec une telle destination. Dans ce cas le nœud source extrait tous les nœuds qui se trouvent dans la direction source\destination. Si un tel ensemble n'existe pas, les données sont diffusées dans le réseau entier. Dans le cas où de tels noeuds existent, une liste contenant leurs identificateurs, est insérée dans l'entête du paquet de donnée avant la transmission de ces derniers. Seulement les noeuds qui sont spécifiés dans l'entête, ont à traiter le paquet. Lors de la réception d'un paquet de donnée, le nœud de transit détermine sa propre liste des noeuds prochains et envoie le paquet avec le nouvel entête. Si aucun voisin n'est localisé dans la direction de la destination, le paquet reçu est ignoré. Quand le noeud de destination reçoit les données, il envoie des acquittements à la source de la même manière. Cependant, dans le cas de réception par inondation, les acquittements ne sont pas envoyés.

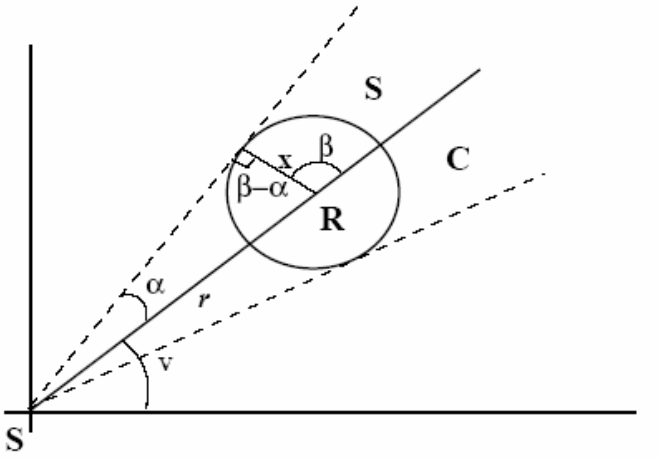


Figure 3.11 Principe d'envoie de donnée dans DREAM

Dans la figure précédente le nœud S veut envoyer des données au nœud R. à l'instant T_0 l'angle entre le segment reliant les deux noeuds R et S et l'axe de référence est v. Les positions géographiques au temps T_0 sont aussi motionnées sur le schéma. Ces informations sont déjà enregistrées dans la table LT du nœud S au temps T_1.

S doit calculer l'angle α afin qu'elle envoie ses données dans la direction [v - α; v + α]. Le noeud R peut se déplacer durant T1- T0 dans un cercle de rayon x = (t1 - t0)*v dans une direction choisie uniformément dans l'intervalle [0; 2Π]. Le paramètre α ne dépend que des informations de mobilité du nœud destinataire R : α = arcsin v (t1- t0)/r.

Le problème dans cette méthode est le choix de α qui est en fonction des informations de mobilité du noeud destination. Or, ces informations peuvent être imprécises, et donc α sera mal estimé, et dans ce cas les noeuds intermédiaires qui se trouvent dans la direction Source\Destination seront aussi mal estimés.

3.13 Le protocole ZRP (Zone Routing Protocol)

Ce protocole propose une approche hybride combinant les deux approches proactive et réactive, Chaque noeud maintient proactivement une route vers chacun des noeuds à moins de K sauts de lui, K étant le rayon de la zone. Le routage interzone est réactif [27].

3.13.1 Principe de Zone

Une zone Z (k, n) pour un noeud n avec un rayon k, est définie comme l'ensemble de noeuds avec le nombre de sauts est moins d'une distance k :

$$Z(k, S) = \{i \mid H(S, i) \leq k\}$$

K est le rayon de la zone et H (i, j) est la distance en nombre des sauts entre le noeud i et le noeud j. Le noeud S s'appelle le noeud central de la zone de cheminement, alors que le noeud G tels que $H(S, G) = k$ s'appelle le noeud périphérique de S.

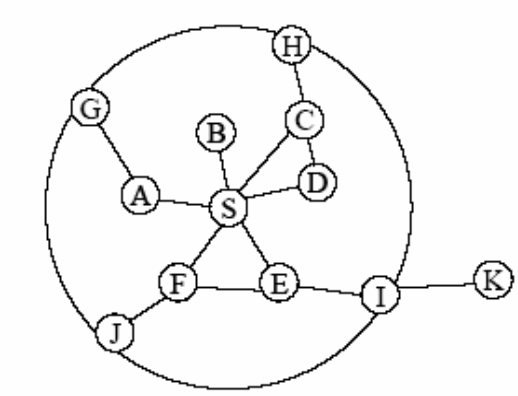


Figure 3.12 Principe de zone dans ZRP

Dans la figure 3.12 : **Z (2, S)** = {A, B, C, D, E, F, J, I, G}

3.13.2 Architecture du protocole

L'architecture du protocole est organisée en quatre composants principaux : le protocole de cheminement d'intra zone (IARP IntrAzone Routing Protocol), le protocole de

cheminement d'inter zones (IERP IntErzone Routing Protocol), le protocole de diffusion (BRP Bordercast Protocol), et le protocole niveau liaison pour découvrir les voisins ainsi l'échec de lien d'un voisin (NDP Neighbor Discovery Protocol). L'IARP fournit des itinéraires proactivement à ces noeuds situés à l'intérieur de la zone du cheminement de la source.

3.13.3 Recherche et Installation d'itinéraire

Quand un noeud source, a des paquets de données à transmettre à une destination particulière, il consulte sa table de routage. Si la destination se trouve en dessous de sa zone, alors un itinéraire existera dans sa table de routage. Autrement, le nœud source lance une recherche au découvert d'itinéraire vers la destination (utilisant le protocole IERP).

Le protocole IERP emploie une forme de diffusion sélective pour exploiter la structure fondamentale de la zone produite par l'IARP. Spécifiquement, la diffusion est basé sur l'envoie des paquets de demandes seulement aux noeuds périphériques (noeuds de frontière), en utilisant un genre spécial de transmission multicast (figure 7.14). Quand un noeud reçoit le paquet de demande d'itinéraire, il répond à la source si la destination est parmi les membres de sa zone, autrement il rediffuse le paquet de demande à ses noeuds périphériques. Par la suite le paquet de demande atteint un noeud ayant la destination en tant que membre de sa zone. Dans ce cas un paquet de réponse soit produit et envoyé à la source [48].

3.13.4 Maintenance d'itinéraire

Dans ZRP, la connaissance de la topologie locale peut être employée pour l'entretien d'itinéraire. L'échec de lien d'un itinéraire actif dans une zone peut être déviés par d'autres nœuds, ainsi les paquets de données entrants peuvent être redirigés autour du lien cassé par un chemin alternatif actif. De même, la topologie locale peut être employée pour raccourcir des itinéraires, par exemple, quand deux noeuds se sont déplacés dans la couverture radio de chacun.

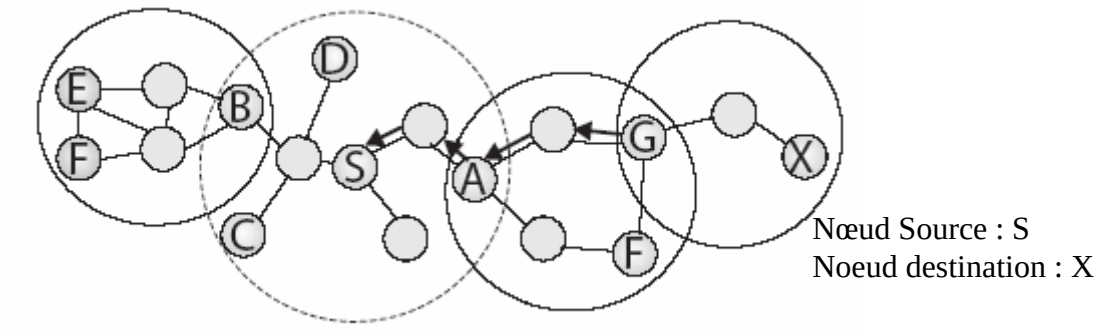


Figure 3.13 Découvert de route dans ZRP

3.14 Le protocole HSR (Hierarchical State Routing)

Le cheminement hiérarchique d'état HSR [28] est un protocole basé sur le cheminement multi-niveaux de groupes. Il maintient une topologie hiérarchique logique en employant le groupement périodique des noeuds. Les noeuds au même niveau logique sont groupés dans des clusters. Les chefs de groupes (clusterheads) élus par le niveau plus bas vont bien êtres des membres d'un niveau plus élevé. Ces nouveaux membres s'organisent à leurs tour dans des clusters, et ainsi de suite. Le but de ce groupement et de réduire les frais généraux de cheminement (c-à-d, stockage, traitement, et transmission de table de cheminement) à chaque niveau.

Un exemple d'une structure hiérarchique à trois nivaux est démontré dans la figure 3.14. Généralement, il y a trois genres de noeuds dans un cluster : clusterheads (dans l'exemple, les noeuds 1, 2, 3, et 4), nœuds passerelles (gateways) (dans l'exemple, les noeuds 6, 7, 8, et 11), et noeuds internes (par exemple, noeuds 5, 9, et 10). Un clusterhead agit en tant que coordonnateur local pour les transmissions dans le cluster.

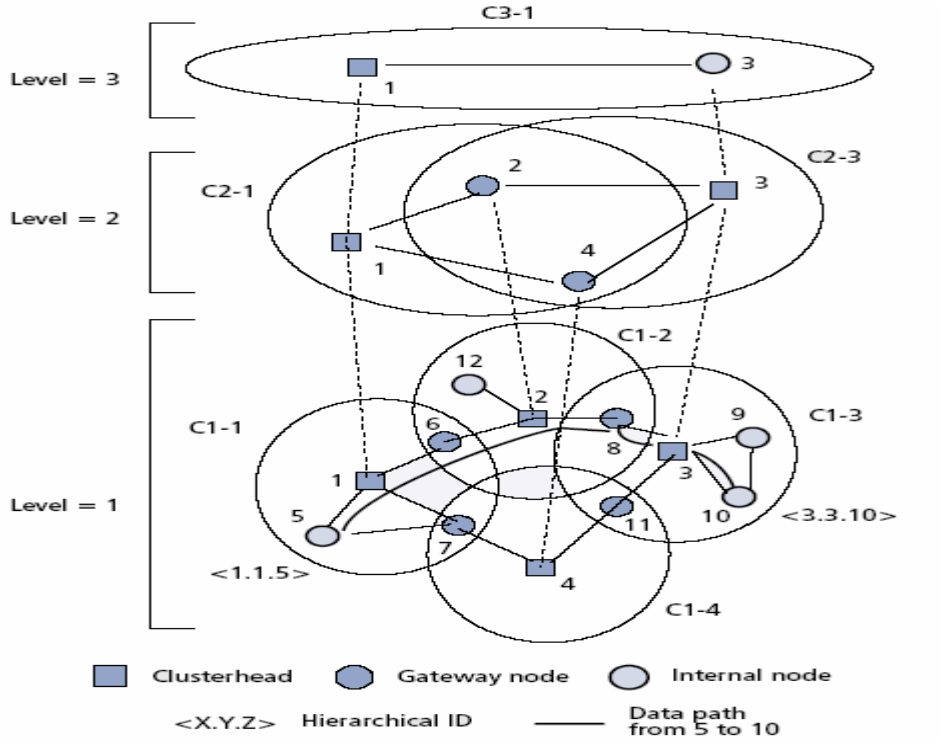


Figure 3.14 Le partitionnement du réseau en cluster

HSR est basé sur le cheminement d'état de lien. Au premier niveau de groupement (appelé aussi niveau physique), chaque noeud surveille l'état du lien de chaque voisin et les diffuses dans le cluster. Le clusterhead récapitule l'information d'état de lien de son cluster et la propage aux clusterhead des clusters voisins par l'intermédiaire des noeuds de passerelles (Geteways). La connaissance de la connectivité entre les clusterheads voisins mène à la formation des clusters du niveau 2. Par exemple, comme montré dans la figure 3.14, les clusterheads des cluster 1 et 2 devenus des membres du cluster C2 du niveau 2. Les entrées d'état de lien aux noeuds du niveau 2 contiennent les liens "virtuels" dans le C2. Un lien "virtuel" entre les noeuds voisins 1 et 2 comprend le chemin du niveau 1 du clusterhead 1 au clusterhead 2 via le noeud passerelle 6. Appliquant le procédé de groupement mentionné ci-dessus récursivement, de nouveaux chefs de clusters sont élus à chaque niveau, et deviennent des membres des clusters de plus haut niveau. Après obtention de l'information d'état de lien à un niveau, chaque noeud virtuel l'inonde aux noeuds des clusters plus bas. En conséquence, chaque noeud physique a l'information "hiérarchique" de la topologie par l'adresse hiérarchique de chaque noeud.

L'hiérarchie ainsi développée exige une nouvelle adresse pour chaque noeud, c'est l'adresse hiérarchique. Les Id's des noeuds montrées dans la figure 3.14 (au niveau = 1) sont des adresses physiques (par exemple, des adresse MAC). Dans HSR, l'identification hiérarchique (hierarchical *ID* HID) d'un noeud est définie comme une séquence des adresses MAC des noeuds sur le chemin du niveau supérieur de l'hiérarchie au noeud lui-même. Par exemple, dans la figure 3.14 l'adresse hiérarchique du noeud 5, HID (5), est < 1.1.5 >. L'avantage de ce schéma hiérarchique d'adresse est que chaque noeud peut dynamiquement et localement mettre à jour ses propres HID on recevant des mises à jour des noeuds des niveaux plus haut. L'adresse hiérarchique est suffisante pour livrer un paquet à sa destination n'importe où dans le réseau en utilisant les tables de HSR. Les noeuds passerelles peuvent communiquer avec plusieurs clusterheads et peuvent être atteints ainsi de la hiérarchie supérieure par l'intermédiaire de plusieurs chemins. En conséquence, un noeud passerelle a des adresses hiérarchiques multiples, semblables à un routeur dans l'Internet, équipé des adresses multiples de sous-adresse. Ces avantages viennent au coût de plus longues adresses (hiérarchiques) et des mises à jour fréquentes de la hiérarchie de cluster et des adresses hiérarchiques pendant que les noeuds se déplacent.

3.14.1 Recherche et Installation d'itinéraire

Quand un noeud source veut envoyer des données à un autre noeud dont l'adresse est connue, il extrait d'abord l'adresse hiérarchie de la destination, et en utilisant sa liste (ou celle du niveau hiérarchique supérieur) il obtient l'adresse hiérarchique de clusterhead du noeud destination. Le noeud source envoie alors, les données au clusterhead de la destination en

utilisant l'adresse hiérarchique obtenue. Lors de la réception, le clusterhead trouve l'adresse de la destination qui appartient à son groupe. Par la suite, le clusterhead de la destination envoie les données vers cette dernière. Une fois que les deux noeuds, la source et la destination, connaissent leurs adresses hiérarchiques, les messages peuvent être délivrés directement sans l'intervention des clusterheads.

3.15 Conclusion

La stratégie de routage est utilisée dans le but de découvrir les chemins qui existent entre les noeuds. Le but principal d'une telle stratégie est l'établissement de routes qui soient correctes et efficaces entre une paire quelconque d'unités, ce qui assure l'échange des messages d'une manière continue.

Nous avons présenté, dans ce chapitre, un panorama des protocoles les plus connus, proposés pour effectuer le routage dans les réseaux ad hoc. Nous avons décrit leurs principales caractéristiques et fonctionnalités qui permettent d'assurer l'acheminement des données.

Chapitre 4

Modélisation et simulation des réseaux Ad Hoc

La modélisation et la simulation sont des méthodes traditionnelles employées pour évaluer la conception des réseaux sans fil. La modélisation et l'analyse mathématique ont introduit quelques aperçus dans la conception de tels systèmes. Cependant les méthodes analytiques sont souvent non générales ou assez détaillées pour l'évaluation et la comparaison de divers systèmes sans fil et mobiles proposés et de leurs services. Ainsi, la simulation aide de manière significative pour obtenir les caractéristiques des performances cruciales.

Décrire en détail les concepts de la modélisation et la simulation dépasse le but de ce chapitre. Nous avons limité notre étude aux modèles de mobilité, ainsi qu'à l'impacte de ces derniers sur les performances des protocoles de routage. On va aussi présenter une introduction à la simulation et quelques paramètres de performance pour l'évaluation d'un tel protocole.

4.1 Conception et modélisation des réseaux Ad Hoc

Le but principal d'une étude basée sur la simulation d'un système MANET est d'obtenir des informations détaillées sur des performances de comportement, d'overheads, de la qualité du service, et beaucoup d'autres métriques concernant le système sous l'étude. L'évaluation des performances d'un tel système par l'intermédiaire de la modélisation et de la

simulation se compose de deux étapes préliminaires : (1) définir un modèle pour le système, et (2) adopter la technique appropriée de simulation pour estimer les métriques d'évaluation des performances du système [64].

4.1.1 Modèle de mobilité

La mobilité d'utilisateur est la valeur principale pour modéliser les réseaux sans fil. Des résultats précis de simulation exigeraient des détails précis pour être modélisés. La mobilité a un rôle central, et elle a un effet approprié à modéliser dans presque chaque analyse de simulation des systèmes sans fil. L'effet de la mobilité sur les politiques de système et les protocoles est approprié à plusieurs couches. L'effet de la mobilité présente des comportements adaptatifs des utilisateurs, des protocoles, et des applications. D'ailleurs, il peut arriver que les modèles de mobilité soient liés au scénario physique sous l'étude. Les modèles de mobilité peuvent parfois être liées à l'application. La plupart de protocoles d'accès au médium (protocoles de routage et de transport) proposés pour des scénarios de MANET sont adaptées et conçues aux besoins pour des modèles de mobilité choisis. Ils se comportent mieux qu'un protocole d'usage universel pour ces scénarios indiqués. L'évaluation des positions des mobiles peut être une tâche informatique appropriée dans la simulation d'un système mobile sans fil. Cela est dû à la mobilité et au nombre élevé d'événements liés aux positions des mobiles. Les MANET ont une architecture sans infrastructure, d'où la majorité des modèles de mobilité adoptés pour les systèmes cellulaires ne sont pas appropriés pour les réseaux MANET. Comme exemple, les modèles de Markov (promenades aléatoires) décrits par des probabilités de migration de cellule à cellule, dont la caractéristique et de décrire la mobilité des mobiles en termes de "nombre moyen des mobiles franchissant la frontière d'un secteur donné". Ce modèle de mobilité n'est pas approprié pour les réseaux Manet.

En général, deux types de modèles de mobilité peuvent être adoptés dans la simulation des réseaux mobiles sans fil, et spécifiquement MANET : Les traces de mouvements et les modèles synthétiques [67].

4.1.1.1 Traces de mouvement

Les traces de mouvement fournissent des informations précises et réalistes sur le modèle de mobilité et le comportement d'utilisateur, en particulier quand la mobilité d'utilisateur est liée à de vrais utilisateurs dans un scénario délimité (par exemple les rues d'une ville, etc..). Malheureusement, les traces exigent de grands fichiers journaux, selon le nombre de mobiles suivis et la durée de temps des échantillons. Les traces ont des

descriptions significatives sur la mobilité des utilisateurs, seulement si les échantillons de mouvement sont rassemblés pour des intervalles de temps significatifs. Si la fréquence des échantillons est faible, des solutions d'approximation (par exemple, interpolation) peuvent être employées. Ces dernières exigent un calcul additionnel, et peuvent avoir comme conséquence un comportement étrange. D'ailleurs, les traces peuvent être rassemblées seulement pour les systèmes existants. Ainsi il est difficile de trouver des traces de MANET parce que de grands scénarios pour ces derniers doivent être mis en application et des applications d'utilisateur doivent être définies. Les traces de mouvement résolvent le problème de définir la distribution initiale et le placement des positions des mobiles d'une manière déterministe. Une autre caractéristique intéressante des traces de mouvement est leur capacité de capturer le vrai effet de corrélation entre la mobilité d'utilisateur et le vrai utilisateur d'application. Il peut arriver que le mouvement d'utilisateur soit conduit par les besoins d'application. En outre, les utilisateurs peuvent se déplacer en montrant un comportement corrélé de groupe.

4.1.1.2 Modèles synthétiques

Les modèles synthétiques sont définis pour représenter la mobilité des utilisateurs d'une manière réaliste et sans employer des traces. Beaucoup de modèles synthétiques ont été définis et adoptés en tant que modèles analytiques. La caractéristique principale qui qualifiait ce type de modèle était la docilité mathématique au lieu du réalisme. De tels modèles ont également survécu dans beaucoup d'études de simulation.

Plusieurs modèles sont définis dans la littérature afin de simuler le comportement des noeuds. Ces modèles sont répartis en deux classes, selon le mode de déplacement des noeuds. Dans la première classe (modèles de mobilité par entité), les nœuds se déplacent indépendamment les uns des autres. Tandis que dans la deuxième (modèles de mobilité par groupe), les noeuds se déplacent en groupe [68].

4.1.1.2.1 Modèles de mobilité par entité

Dans cette classe, les noeuds se déplacent de manière indépendante les uns des autres dans une zone de simulation, généralement bornée. Plusieurs modèles individuels ont été proposés dans la littérature. Nous en présentons ici les principaux.

Marche Aléatoire (Random Walk RW)

Souvent dans la réalité, les noeuds se déplacent de manière imprédictible. Le modèle de Marche Aléatoire a été développé afin de simuler de telles situations. Dans ce modèle, un noeud se déplace d'un point A à un point B, en choisissant aléatoirement sa vitesse et sa

direction de mouvement, dans les intervalles [speedmin, speedmax] et [0, 2Π] respectivement où speedmin et speedmax sont la vitesse minimale et maximale respectivement. Le déplacement de ce noeud dans une durée constante t ou dans une distance constante d. À la fin de ce déplacement, le noeud choisira de nouveau, une vitesse et une direction de mouvement vers un autre point. Le modèle de Marche Aléatoire est sans mémoire, car il ne maintien aucune connaissance antérieure, ni sur la localisation, ni sur la vitesse ou la direction des nœuds (utilisateurs mobiles) [68,69].

Chemin de but aléatoire (Random Waypoint RWP)

Ce modèle aléatoire inclut un temps de pause entre les changements de la direction et/ou de la vitesse des mobiles. Un mobile reste dans un endroit pendant une certaine période (c-à-d, un temps de pause). Une fois cette période expirée, le mobile choisit une destination aléatoire dans le secteur de simulation et une vitesse qui est uniformément distribuée entre [minspeed, maxspeed]. Le mobile se déplace alors vers la destination nouvellement choisie et avec la vitesse choisie. À l'arrivée, le mobile fait une pause pendant une période de temps indiquée avant de commencer le processus de déplacement une autre fois (Figure 4.1).

Nous notons que le modèle de mouvement d'un mobile employant ce modèle est semblable au modèle de mobilité de marche aléatoire si le temps de pause est zéro.

Le modèle RWP est également un modèle de mobilité largement répandu.

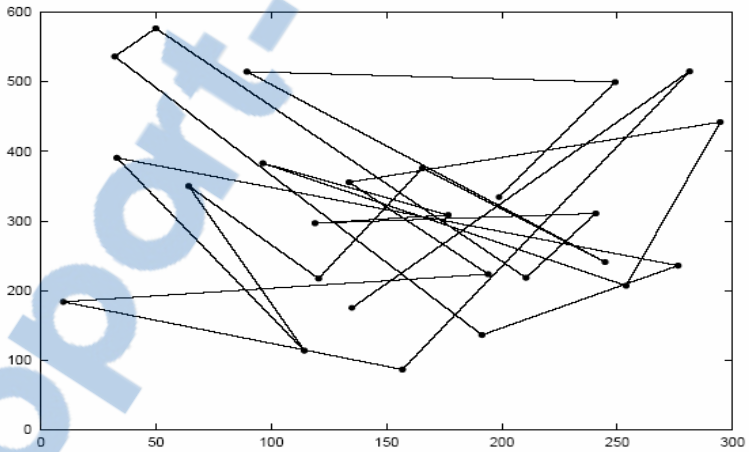


Figure 4.1 Le modèle de déplacement des mobiles employant le modèle de chemin de but aléatoire.

Modèle de direction aléatoire (Random Direction)

Le modèle RWP peut créer un problème de concentration ou de groupement des nœuds dans une même zone de simulation. Cela est due à la forte probabilité qu'un nœud choisie une direction proche du centre de la zone de simulation (figure 4.1).

Pour éviter ce genre de comportement, un noeud suivant le modèle de Direction Aléatoire et comme dans le cas du modèle de Marche Aléatoire, choisit une direction aléatoire vers laquelle il va se déplacer. Le noeud continue son parcours suivant la direction choisie jusqu'à atteindre les bornes de la zone de simulation. Le noeud marque ensuite un temps de pause d'une durée spécifique, ensuite choisit une nouvelle direction et réitère le processus de nouveau [70].

Modèle Gauss Markov

Le modèle de mobilité de Gauss-Markov a été conçu pour s'adapter à différents niveaux d'aspect aléatoire par l'intermédiaire d'un paramètre d'ajustement. Initialement à chaque nœud sont assignées une vitesse et une direction. Pour un intervalle fixe de temps, n, le mouvement se produit en mettant à jour la vitesse et la direction de chaque mobile. Spécifiquement, la valeur de la vitesse et de la direction au n^{ème} instant est calculée à base de la valeur de la vitesse et la direction de l'instant (n-1) ^{éme} et une variable aléatoire, en utilisant les équations suivantes :

$$s_n = \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}}$$

$$d_n = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)}d_{x_{n-1}}$$

Où S est la vitesse du mobile et d est sa position ; α est un paramètre d'ajustement où $0 \leq \alpha \leq 1$; $\bar{s}$ et $\bar{d}$ sont des constantes représentant, respectivement, la vitesse moyenne et l'angle moyen de déplacement lorsque $n \rightarrow \infty$; S_{n-1} et d_{n-1} sont des variables aléatoires de distribution Gaussienne.

Le modèle Gauss-Markov élimine les arrêts soudains et les virages aigus effectués par les mouvements des nœuds dans les cas des modèles déjà cités en dessus.

Région de Simulation Illimitée (A Boundless Simulation Area)

Dans le modèle de mobilité de région de simulation illimité, il existe une relation entre la direction et la vitesse précédente du mouvement d'un mobile avec sa direction et sa vitesse courante. Le vecteur de vitesse V = (v ; q) est employé pour décrire la vitesse v d'un mobile aussi bien que sa direction q ; La position du mobile est représentée comme (x;y). Le vecteur de vitesse et la position, sont mis à jour à chaque intervalle de temps Δt selon les formules suivantes :

$$v(t+\Delta t) = min[max(v(t)+\Delta v, 0), V_{max}];$$

$$\theta(t+\Delta t) = \theta(t) + \Delta\theta;$$

$$x(t+\Delta t) = x(t) + v(t) * \cos\theta(t);$$

$$y(t+\Delta t) = y(t) + v(t) * \sin\theta(t);$$

Où V_{max} est la vitesse maximale définie dans la simulation, ΔV est le changement dans la vitesse qui est défini uniformément dans l'intervalle [$-A_{max}* \Delta t$, $A_{max}* \Delta t$], A_{max} est l'accélération maximale d'un noeud donné ; Δθ le pas de changement de direction qui est uniformément distribuer dans l'intervalle [$-\alpha* \Delta t$, $\alpha* \Delta t$], α est l'angle maximum de changement dans la direction d'un nœud.

Le modèle Région de Simulation Illimitée se différencier par la manière de traiter les limites de la zone de simulation. Dans les autres modèles, un noeud s'arrête dès qu'il atteint une borne de la zone de simulation. Or, dans le modèle de Région de Simulation Illimitée, un noeud atteignant une borne de simulation, continue son déplacement et réapparaît sur la borne opposite. D'où la nomination de Région de Simulation Illimitée.

4.1.1.2.2 Modèles de mobilité par groupe

Dans les Modèles de mobilité par entité citée en dessus, les mouvements des mobiles sont complètement indépendants l'un de l'autre. Dans un réseau ad hoc, cependant, il y a beaucoup de situations où il est nécessaire de modéliser le comportement de mouvement de groupe de mobiles ensemble. Par exemple, un groupe de soldats dans un scénario militaire travaille ensemble d'une façon coopérative pour accomplir une tache commune. Afin de modéliser de telles situations, un modèle de mobilité de groupe est nécessaire pour simuler cette caractéristique coopérative. Nous présentons dans ce qui suit quelques modèles.

Modèle de mobilité aléatoire exponentiellement corrélée (Exponential Correlated Random Model)

Dans ce modèle, une fonction de mouvement est employée pour créer des mouvements des mobiles. La position (mobile ou groupe) au temps t, $\vec{b}$ (t) est employé pour définir la prochaine position (mobile ou groupe) au temps t + 1, $\vec{b}$ (t + 1) :

$$b(t+1) = b(t).e^{-\frac{1}{\tau}} + (\sigma\sqrt{1-(e^{-\frac{1}{\tau}})^2}).r$$

Le facteur τ ajuste le taux de changement du mobile, de l'endroit précédent à son nouvel endroit, et r est une variable gaussienne aléatoire de variance σ. Malheureusement, il n'est pas facile de créer un modèle de mouvement donné, en choisissant des valeurs appropriées pour (τ, σ) dans le modèle aléatoire corrélé exponentiel de mobilité.

Modèle de graviter (Gravity Model)

Ce modèle peut être employé dans les scénarios où les mobiles peuvent tendre à se déplacer vers certaines destinations, appelés point d'attraction. A chaque mobile est assigné une charge positive, ainsi aux points d'attraction est assignée une charge négative. Les charges opposées s'attirent, alors que les mêmes charges se repoussent. Les mobiles sans la charge n'ont aucun effet de graviter.

Modèle de mobilité avec groupe de points de référence (Reference Point Group Mobility RPGM)

C'est le modèle de mobilité de groupe le plus général. Spécifiquement, le modèle de colonne, le modèle nomade de la Communauté, et le modèle de poursuite peuvent être mis en application en tant que cas spéciaux du modèle de RPGM. Un centre logique pour le groupe est défini. Le centre du groupe caractérise le déplacement des membres du groupe. Chaque noeud se déplace autour d'un point de référence prédéterminé, dont son déplacement dépend du centre logique du groupe. Lorsqu'un point de référence se déplace entre l'instant t et l'instant t + 1, sa localisation est mise à jour en fonction du centre logique du groupe. Une fois les points de référence calculés, ils sont combinés avec un vecteur aléatoire afin de représenter le déplacement aléatoire de chaque noeud par rapport à son point de référence.

Modèle de mobilité de poursuite (Pursue Mobility Model)

Le modèle de mobilité de poursuite essaye de représenter un modèle où les mobiles poursuivre une cible particulière. Par exemple, des officiers de police essayant d'attraper un criminel échappé. Le modèle de mobilité de poursuite se compose d'une équation simple de mise à jour pour la nouvelle position de chaque mobile :

Nouvelle_position = Ancienne_position + Acceleration (Cible_ancienne_position) + Vecteur_aléatoire

Où Acceleration (Cible_ancienne_position), est une information du déplacement du noeud poursuivi et random_vector est un pas de déplacement aléatoire de chaque noeud. La valeur de Vecteur_aléatoire est obtenue par un modèle de mobilité par entité. Le taux d'aléa du mouvement d'un nœud est limité afin que ce dernier poursuive effectivement la cible.

Modèle de mobilité de colonne (Column Mobility Model)

Ce modèle définit un modèle de mobilité semblable à une colonne des soldats marchant dans en ligne. Chaque mobile a un point de référence dans la colonne et se déplace aléatoirement autour de ce point. Tous les mouvements des points de référence (c-à-d, la colonne) se déplacent suivant un vecteur prédéfini.

4.1.2 L'impact de choix des modèles de mobilité sur les performances des protocoles

Dans [68], les auteurs ont prouvé par des tests de simulation que le choix du modèle de mobilité affecte significativement les performances des protocoles de routage.

La simulation du protocole DSR [34] avec plusieurs modèles de mobilités a donné les résultats des figures 4.2, figure 4.3 et la figure 4.4 . Ces derniers illustrent grandement l'effet du choix des modèles de mobilités sur les performances des protocoles de routage.

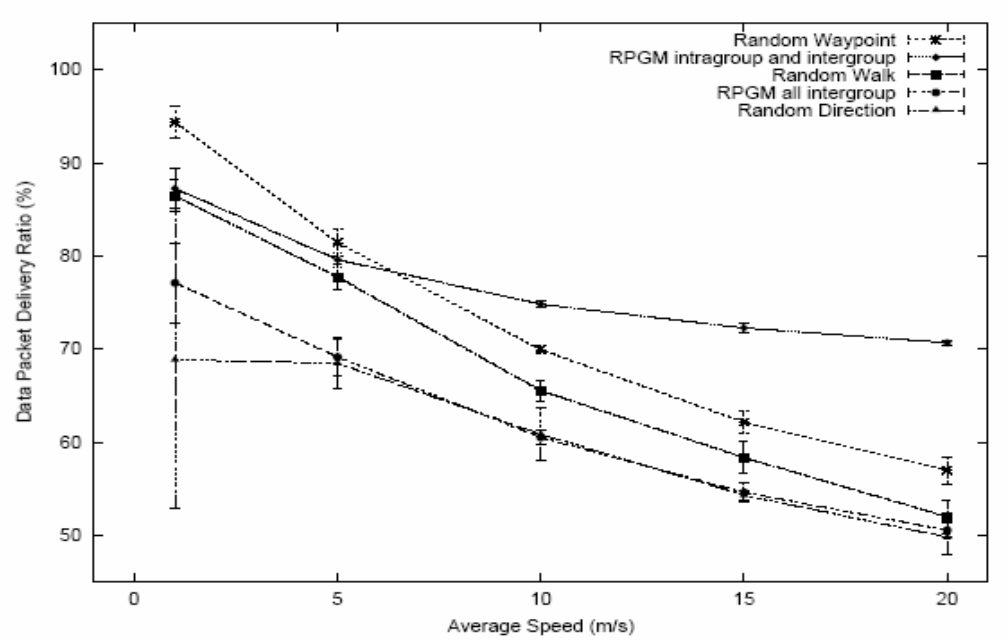


Figure 4.2 délais de bout en bout par rapport à la vitesse

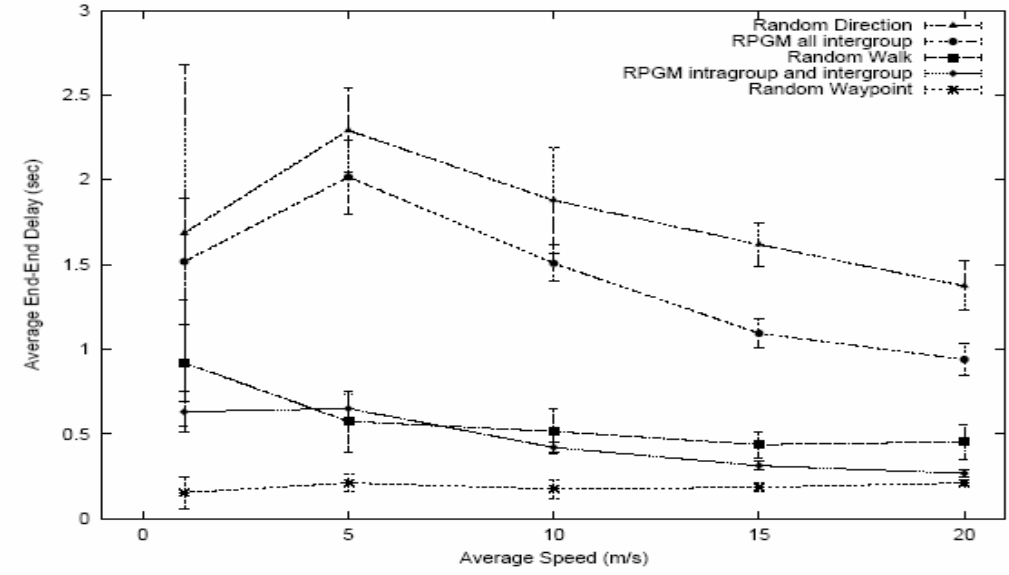


Figure 4.3. Nombre de sauts par rapport à la vitesse

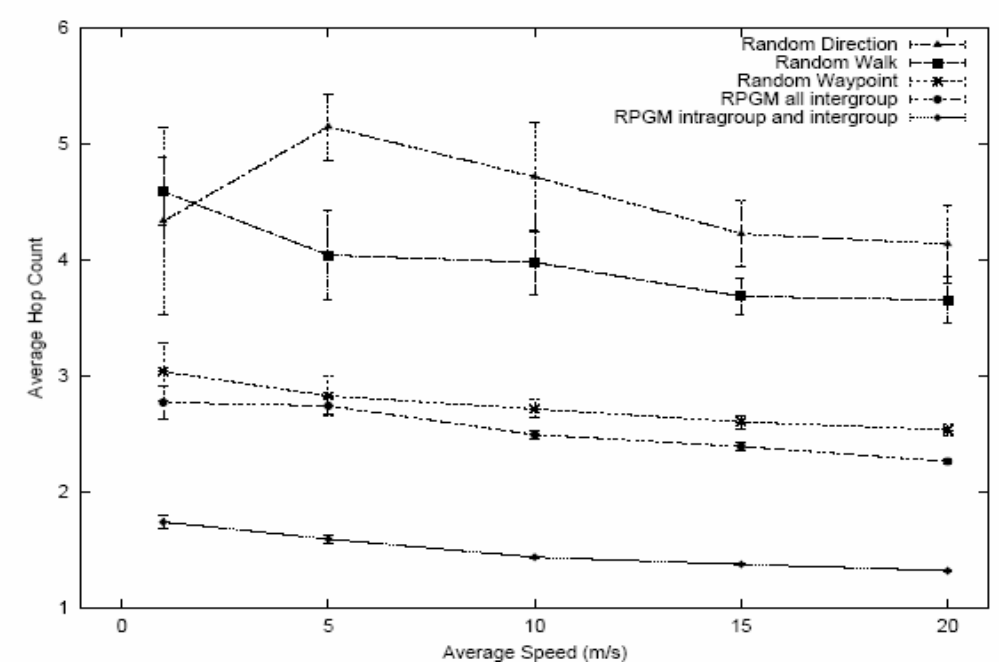


Figure 4.4 surcharge des paquets de contrôle par rapport à la vitesse

4.1.3 Simulation des réseaux Ad Hoc

La simulation à événements discrets est l'une des méthodes les plus flexibles pour l'évaluation des performances des systèmes complexes tels que MANET. Le but d'une étude de simulation est la construction d'un simulateur qui imite les transitions d'état de système. Il permet ainsi de rassembler et d'analyser les données pendant l'exécution de la simulation et estime les métriques de performances du système sous l'analyse. Une étude orthodoxe de simulation est basée sur plusieurs étapes dont les caractéristiques et le nombre peut changer à savoir la nature du système analysé et les objectifs de l'étude. Les étapes principales pour n'importe quelle étude de simulation sont (1) formulation du problème, (2) caractérisation de charge de travail, (3) définition et validation du modèle, (4) construction et vérification du simulateur, (5) conception des expériences, et (6) analyse des résultats de simulation [64].

Les réseaux de MANET sont caractérisés par les topologies dynamiques, exigeantes des protocoles adaptatifs de cheminement et de multi-sauts, traitant les liens bidirectionnels et unidirectionnels. Les liens ont la contrainte de largeur de bande comparée aux réseaux filaires, et ils offrent des capacités et des délais variables. Puisque les nœuds mobiles ont la contrainte d'énergie, les protocoles de MANETs doivent prendre en considération le traitement de la réduction d'énergie (exemple, la gestion de période de mise en veuille et la réduction adaptative de puissance). Dû à la mobilité des noeuds, le passage à l'échelle des

réseaux MANET, est un problème difficile à résoudre. Ce problème est compliqué en plus par l'exécution distribuée des protocoles dans ce type de réseau. Ceci rend difficile de garantir le comportement, la fiabilité, l'équité et l'efficacité de réseau dans chaque condition. La réduction d'overheads pour maintenir la fonctionnalité appropriée de réseau est un problème commun : l'utilisation des ressources critiques, comme l'énergie de batterie, la mémoire et la largeur de bande pour la transmission des paquets de contrôle devraient être réduites au maximum.

4.1.3.1 Métriques de performance

Un grand ensemble de métriques de performances, a pu être défini pour évaluer les réseaux sans fil, afin d'évaluer les caractéristiques critiques du système considéré. Quelques métriques peuvent être considérées appropriée ou significative seulement pour une couche donnée de protocole. D'autres métriques peuvent être générales [65, 66].

Les métriques de performances peuvent être rudement divisées en trois catégories qui sont les suivantes :

Métrique de performances d'utilisation : inclut la latence, le délai, la qualité de service, les priorités, fiabilité, et métrique de rentabilité etc.

Métrique d'utilisation de ressource : inclut les overheads, l'utilisation, l'équité, et l'efficacité etc.

Métrique de système : Les métriques de système incluent la stabilité, la scalabilité, et les métriques de contexte (par exemple, changements de topologie, partitionnement du réseau, durabilité des clusters, mobilité, densité, charge, longueur de chemin, etc.)

pour l'évaluation des protocoles de routage, les métriques intéressante peuvent être la moyenne du débit de bout à bout, le délai moyen de bout à bout, l'utilisation moyenne du lien, la probabilité moyenne de perte des paquets, l'efficacité énergétique, et les overheads de protocole, etc. .....

Dans ce qui suit nous allons présenter quelques métriques utilisées dans l'évaluation des protocoles de routages et de l'accès au médium.

Débit et utilisation (Throughput and Utilization) : Le but de n'importe quel protocole de transmission est de maximiser le nombre de bits transmis et réduire au minimum le délai moyen d'accès. Le débit T est défini comme la taille moyenne S d'un bloc de donnée indiquée

(paquet), divisée par le délai d'accès moyen correspondant D, c-à-d, T = S / D. Cet index est lié à l'indice d'utilisation U, qui peut être défini comme la fraction de la capacité du canal C utilisée pour la transmission réussie des données.

Overheads : Chaque ressource dans le système qui n'est pas strictement nécessaire pour transmettre la charge utile de la communication, peut être considérée comme frais généraux et devrait être réduite au maximum (exemple, largeur de bande, énergie, etc..). L'overhead dans les protocoles de routage est le nombre de paquets de contrôle ou la taille totale des paquets de contrôle. Ce dernier inclut l'overhead des paquets de données, par exemple dans les protocoles source_route les paquets de contrôle sont en entiers des paquets de contrôle. Le nombre total de paquets (ou la taille totale) envoyés pendant la simulation entière est rapporté.

Fiabilité (Reliability) : Ce concept définit une mesure de fiabilité de système en ce qui concerne beaucoup d'échecs qui peuvent être prévus, par exemple, congestion du réseau et échec de chemins. La fiabilité peut être évaluée comme mesure de probabilité d'échecs, et comme une mesure de délai de rétablissement d'échec.

Scalabilité (passage à l'échelle) : Un système extensible est obtenu quand les protocoles et la gestion des ressources réagissent et s'adaptent d'une manière opportune aux changements des facteurs de système comme la charge et le nombre de mobiles. Un système scalable est un système dans lequel les performances passent à l'échelle sans diminution. Si une diminution se produit, il serait intéressant de trouver des informations sur le point de saturation, c-à-d, la limite que le système peut soutenir, ainsi que le temps de rétablissement des états de saturation. Un exemple typique est donné par les problèmes de congestion dans les réseaux.

Consommation d'énergie : Les mobiles ont des batteries limitées, ainsi la consommation efficace de l'énergie est exigée pour chaque tâche accomplie, y compris l'entretien de système, la transmission, et la réception des données.

Temps d'accès : C'est le temps passé par une trame (ou un paquet) dans la file d'attente de la couche MAC (routage au niveau transport). Il est défini à partir de l'instant où la trame est enfilée (ou retiré de la file d'attente) jusqu'à ce que sa transmission soit accomplie avec succès. Puisque le délai dépend de la définition de protocole et également de la charge de système et du modèle de trafic, les comparaisons devraient être effectuées dans des mêmes conditions.

Capacité Du canal : C'est la quantité maximale de données qui peut être transmises au-dessus d'un canal simple. Le débit binaire nominal peut être réduit en présence du bruit et d'interférence.

4.2 Conclusion

Dans ce chapitre nous avons présenté les méthodes de modélisation des réseaux ad hoc, nous avons limité notre présentation aux modèles de mobilité que nous avons jugés intéressant et important de connaître ainsi que l'impact de choix de ces modèles sur l'évaluation des performances de tout protocole. Nous avons aussi présenté la méthode classique pour l'évaluation des performances des protocoles qui est la simulation. Ainsi une brève description de quelques paramètres de performances.

Dans le chapitre suivant nous allons présenter les détails de notre proposition avec les résultats de simulations, ainsi qu'un bref aperçu sur le simulateur OPNet que nous avons adopté pour valider notre proposition.

Chapitre **5**

Mobilité et rupture de route dans les réseaux Ad Hoc

A cause de la forte mobilité des nœuds dans le contexte des réseaux ad hoc, les routes sont souvent instables et les informations d'état de lien utilisées par les protocoles de routage traditionnels peuvent alors devenir rapidement obsolètes. Par conséquent, les ruptures de routes sont beaucoup plus fréquentes et doivent être recalculées très souvent. Ce qui provoque évidemment des interruptions de connexions et une perte de qualité au niveau des applications.

Dans ce chapitre, nous proposons des nouvelles métriques pour élire des chemins stables entre les entités communicantes. Nous allons aussi proposer des mécanismes de prédiction de rupture de route, ainsi que des mécanismes pour maintenir ces routes localement ou de bout en bout.

5.1 Stabilité d'itinéraire

Un chemin stable est un chemin constitué de liens stables. De ce fait, nous allons définir en premier lieu la notion de lien stable dans notre proposition ou bien les métriques de choix d'un lien stable. Ensuite nous décrivons le protocole d'installation d'un chemin. En cas de rupture d'un lien de chemin, La phase de maintenance assure la réparation de ce dernier localement et en cas ou cette tentative échoue, la solution de bout en bout (de la source à la destination) sera prise.

Dans ce chapitre, nous allons présenter deux métriques de choix de liens stables. La première métrique est extraite des protocoles SSA et ABR et dans la deuxième, nous définissons notre propre métrique.

5.2 Concept de lien stable

Nous allons présenter dans cette section les métriques de choix d'un lien stable :

5.2.1 Métriques extraites des protocoles SSA et ABR

Dans ce cas, les métriques de choix d'un lien stable sont extraites des protocoles SSA et ABR. Chaque nœud diffuse périodiquement un message Hello à ses voisins. Chaque nœud reçoit un paquet Hello de son voisin, enregistre et met à jours sa table SST (Signal Stability Table) qui a comme colonnes *(ID, Set, Clicks, Ticks)*, comme le montre l'algorithme suivant :

Algorithme 5.1. M-A-J de la table SST

```
Ticks++
Si  puissance de réception (message Hello) >  SS_threshold  Alors
Debut
        Clicks++
        Set = S
Fin
Sinon
Début
        Clicks = 0
        Set = W
Fin
```

Ticks : le nombre total des messages Hello reçus par ce nœud.

SS_threshold : Un seuil défini expérimentalement, représente la puissance minimale pour qu'un signal soit considéré comme un signal fort.

Set : Une colonne utilisée pour enregistrer la qualité du signal. Elle prend la valeur S (Strongly) si la puissance du signal est supérieur à SS_threshold, sinon elle prend la valeur W (Weakly).

Clicks : Est le nombre de fois que le nœud voisin est observé avec une forte puissance de signal.

Chaque nœud qui doit participer dans l'élection d'un chemin, calcule la stabilité du lien **ST** avec le nœud qui l'a sollicité en consultant l'entrée de sa table SST. Si le Set est à S et la valeur du Clicks est supérieure à un seuil ***Click_threshold*** alors le lien est stable et ST prend la valeur Vraie, sinon ST prend la valeur Fausse. Exemple : dans la figure 5.1, l'itinéraire ABCDE sera élu comme itinéraire stable entre les deux nœuds A et E.

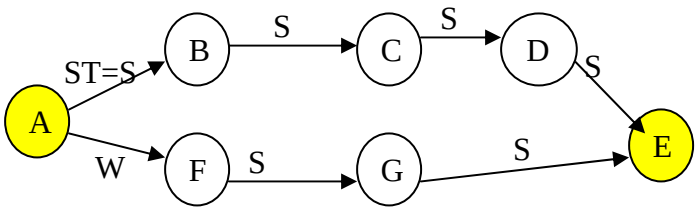


Figure 5.1 : Choix d'itinéraire utilisant la stabilité du lien ST.

Un chemin est dit stable s'il est formé uniquement par des liens stables, ce qui n'est pas toujours réalisable. Pour palier à ce problème, un autre critère pour élire des chemins, en cas d'absence de chemin stable, est décrit par le chemin avec la valeur maximale de RT (Ticks Route). Cette dernière est la plus petite valeur des Ticks d'un chemin. Exemple : Figure 5.2.

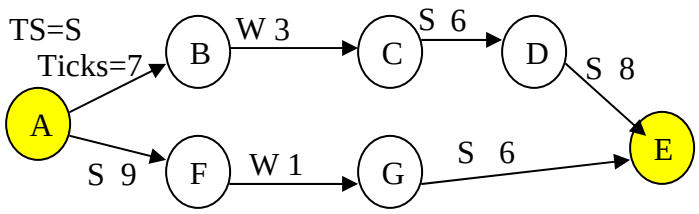


Figure 5.2 : Choix d'itinéraire utilisant le RT des itinéraires comme métrique.

Le TR du chemin ABCDE est de 3 et le TR du chemin AFGE est de 1. Donc, le chemin ABCDE sera élu dans le cas où aucun itinéraire stable n'est trouvé.

5.2.2 Métriques basées sur les zones de propagation du signal

Chaque nœud diffuse un signal avec une vitesse fixe et une puissance qui diminue de plus en plus au fur et à mesure qu'il s'éloigne de son émetteur jusqu'à ce qu'il ne soit plus décodé. Cette surface de propagation du signal est divisée dans notre proposition en zone Z_1, Z_2,…, Z_n (Figure 5.3). Chaque zone Z_i est délimitée par l'intervalle $[P_{min}, P_{max}]$, P_{min} et P_{max} qui sont des puissances de réception ($P_{min} << P_{max}$).

Z_1 :] P_1, P_0], Z_2 :] P_2, P_1], Z_3 :] P_3, P_2] … Z_n :] P_n, P_{n-1}] (Z_n : ≤ P_{n-1})

Où :

P_0 est la puissance d'émission initiale du signal.

P_1, P_2, …,P_n sont des puissances choisies de telle sorte que $P_0 >> P_1 >> P_2 >> \ldots >> P_{n-1} >> P_n$, ces puissances délimitent les intervalles de puissances des zones.

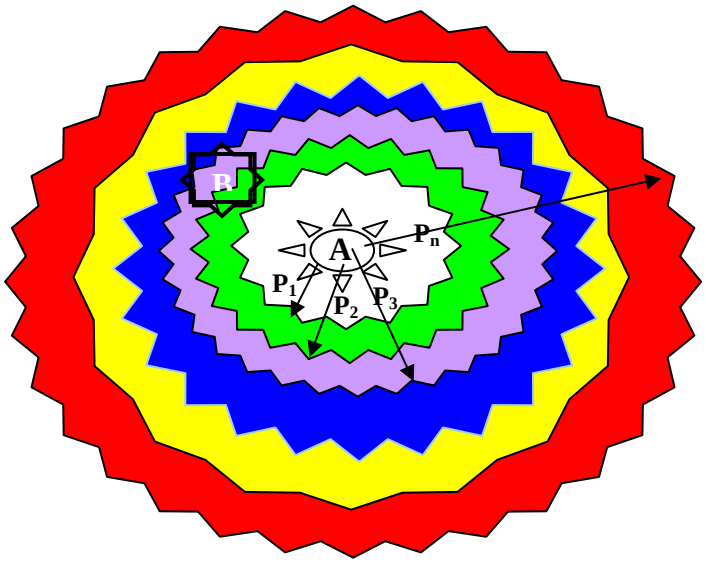


Figure 5.3 : Propagation du signal diffusé par un mobile.

5.2.2.1 Métriques de stabilité de lien

Chaque nœud diffuse un message Hello à chaque durée de temps prédéfinie. Quatre concepts dans notre proposition peuvent affecter le choix d'un lien stable :

a- Un nœud B se trouve dans la portée de A, bouge par rapport à A par une vitesse de mouvement en zone Vm calculée comme suit :

Vm = Nb_Zone_Franchis/Fenêtre_Temps

Le Nb_Zone_Franchis est le nombre de zone que B a franchi dans un temps Fenêtre_Temps. Si le nœud B se déplace sur le même rayon par rapport à A, sa vitesse de mouvement Vm est égale à 0. Hors qu'en réalité on ne trouve pas un nœud qui se déplace autour d'un nœud et sur le même rayon.

b- Si le mobile B franchi les zones dans le sens $Z_{i+1} \rightarrow Z_i$ alors le mobile se rapproche de A et dans le cas contraire il s'éloigne de A. Ainsi, si le mobile est observé pendant une durée sur le même rayon il est considéré comme immobile par rapport à A. Dans notre proposition, le sens du mobile est représenté par une variable **Sens** qui peut prendre deux valeurs : –1 (si le mobile s'éloigne) ou +1 (si le mobile se rapproche ou il est immobile).

c- Le nombre de fois (nombre de message Hello) que le mobile B est observé par le mobile A est représenté par une variable **Nb_Hello**. Cette dernière est initialisée à 1 pour la première fois que le mobile B est observé. Ainsi, à chaque réception du message Hello de B, le mobile A incrémente la variable **Nb_Hello** par 1.

d- La dernière zone où le mobile B est observé (la puissance de réception du dernier message Hello) est représentée par la variable **Zone_Locale**.

5.2.2.2 Fonction de Stabilité de lien

Les quatre concepts cités ci dessus peuvent être regroupés dans une seule formule qui constitue la Fonction de Stabilité de lien FS comme suit :

$$\mathbf{FS_{AB}} = a^* 1/Vm + b^* Sens + c^* Nb\_Hello + d^* 1/ Zone\_Locale$$

Les constantes a, b, c et d constituent des poids qui donnent l'importance à un critère par rapport aux autres. Ces paramètres peuvent être donnés soit manuellement (par des tests de simulation) soit trouvés automatiquement selon le modèle de mobilité. Exemple : Dans un modèle avec une forte mobilité, le poids de la vitesse de mouvement en zone sera le plus grand. Dans le cas où le modèle est avec une faible mobilité, le paramètre du **Sens** sera le plus grand. Ainsi dans le cas où le modèle représente une variation entre faible et forte mobilité, le poids du paramètre **Nb_Hello** sera le plus grand, etc.

Plus la valeur de la fonction FS est grande plus le lien est stable, et plus elle est faible plus le lien et moins stable.

La fonction de stabilité FS peut être décrite sous différents formats, par exemple les termes 1/Vm et Nb_Hello peuvent s'écrire Nb_Hello /Vm, etc. Le choix de l'écriture de la fonction, telle quelle est mentionnée dans la formule ci-dessus, est de donner, à chaque concept à part, un poids différent des autres. Avec ces poids, on peut bien représenter le modèle de mobilité que les nœuds réalisent.

5.2.2.3 Zone de rupture

Un Nombre des Dernières Zones N_D_Zone (exemple : 2 ou 3 zones) constitue la zone de rupture de lien. Quelque soit sa vitesse ou sa direction, un nœud B, se trouve sur la zone de rupture d'un nœud A, a une probabilité élevée qu'il sort de la porté de A dans un délai très court. Pour cette raison, le calcul de la fonction FS sera décrit par l'algorithme suivant :

Algorithme 5.2 Calcul de la Fonction de Stabilité FS

```
/* Nb_Zone : est le nombre total des zones.
        Si Zone_Locale > (Nb_Zone  –  N_D_Zone) Alors
                FS_AB = 0
        Sinon
                FS_AB = a* 1/Vm + b* Sens + c* Nb_Hello + d* 1/ Zone_Locale.
```

5.3 Etablissement d'itinéraires

Nous allons présenter dans cette section les détails de la procédure de recherche et d'installation d'itinéraire, ainsi que l'algorithme de choix d'itinéraires parmi les itinéraires possibles.

5.3.1 Recherche et installation d'itinéraire

Avant de décrire les détails de la procédure d'installation d'itinéraire, en va décrire les formats de paquet de demande d'itinéraire RREQ et de réponse d'itinéraire RREP.

Format des paquets RREQ et RREP :

Le paquet RREQ contient les champs suivants **(IDSRC, IDDEST, NSEQ, LNI, TTL).** Dans le cas de la première métrique, Le champ LNI contient la Liste des identificateurs des Nœuds Intermédiaires, et les champs TS et Ticks sont utilisés pour enregistrer l'état de stabilité des liens parcourus. Dans le cas de la deuxième métrique, le

champ LNI contient les identificateurs des nœuds intermédiaires, ainsi que les valeurs FS des liens parcourus sont utilisées pour enregistrer leurs états de stabilité.

Le paquet RREP est un paquet unicast, qui contient les champs suivants : **(Stable_Route, multipath).** Le champ Stable_Route est utilisé pour enregistrer le chemin le plus stable élu par la destination. Le champ multipath est utilisé pour enregistrer tous les autres itinéraires alternatifs élus par la destination.

Recherche et Installation d'itinéraire :

Dans notre proposition, l'établissement d'itinéraire se fait sur demande. Chaque nœud demandeur lance une recherche d'itinéraire par la diffusion du message RREQ à ses voisins. Initialement, ce dernier contient : ID IDSRC, l'ID du destinataire IDDEST, un numéro de séquence unique NSEQ, le nombre TTL de saut maximal que le paquet RREQ qu'il peut atteindre et une liste LNI vide.

Chaque nœud reçoit le paquet RREQ, compare IDDEST avec son adresse. Deux cas possible :

- S'ils sont les mêmes alors c'est lui la destination en question.
- Sinon, il insert dans la liste LNI du paquet RREQ son ID et l'état du lien avec le nœud qui lui a diffusé le message. Dans le cas de la première métrique, l'état d'un lien est représenté par ses valeurs de TS et de Ticks et dans la deuxième métrique par sa valeur du FS. Dans le dernier cas, si la valeur du FS est égale à 0, le paquet RREQ est détruit sans aucun traitement (exemple : Dans la figure 5.4, le nœud C, qui a reçu un paquet RREQ de B, détruit ce paquet sans aucun traitement parce que le FS de ce lien =0). Le nœud intermédiaire enregistre ensuite dans son cache l'IDSRC et le NSEQ, décrémente le nombre TTL de 1 et rediffuse le message à ses voisins si son TTL est supérieur à 0. Pour éviter les boucles, chaque noeud consulte l'entête du paquet RREQ, s'il trouve que l'IDSRC et le NSEQ sont déjà enregistrés dans son cache, il détruit le paquet RREQ sans aucun traitement. Ce processus se continue jusqu'à que le paquet RREQ atteint la destination.

Après la réception des paquets RREQ, la destination, et selon l'algorithme décrit dans la section suivante (Algorithme de choix d'itinéraire), élit un itinéraire primaire (le plus stable), ainsi que d'autres itinéraires alternatifs. La destination répond par un message RREP envoyé à la source sur le chemin le plus stable dans le sens inverse. La destination enregistre la liste de tous les chemins alternatifs élus dans le champ Multipath du paquet RREP et le chemin le plus stable (parmi les chemins élus) dans le champ Stable_Route. Chaque nœud intermédiaire reçoit le paquet RREP, ajoute dans sa table une entrée vers la destination. Il enregistre, comme prochain nœud, le nœud qui lui a diffusé le message, et comme précédent nœud, le nœud qui figure dans le paquet RREP juste avant lui (dans le champ Stable_Route du paquet RREP reçu). Il enregistre la portion du chemin qui vient juste après lui jusqu'à la destination dans son cache pour utilisation dans la procédure de maintenance locale

(exemple : sur la figure 5.4 le nœud F, après réception du message RREP, enregistre dans son cache la portion du chemin FGE dans son cache). Il envoie le paquet à son prédécesseur qui figure dans le champ Stable_Route du paquet RREP. Ce processus se continue jusqu'à ce que le message RREP atteint la source.

Après la réception du message RREP, la source ajoute dans sa table de routage une entrée vers la destination, et comme prochain nœud le nœud qui lui a diffusé le message. Elle enregistre les autres itinéraires alternatifs dans son cache pour une utilisation ultérieure (en cas où l'itinéraire primaire soit rompu).

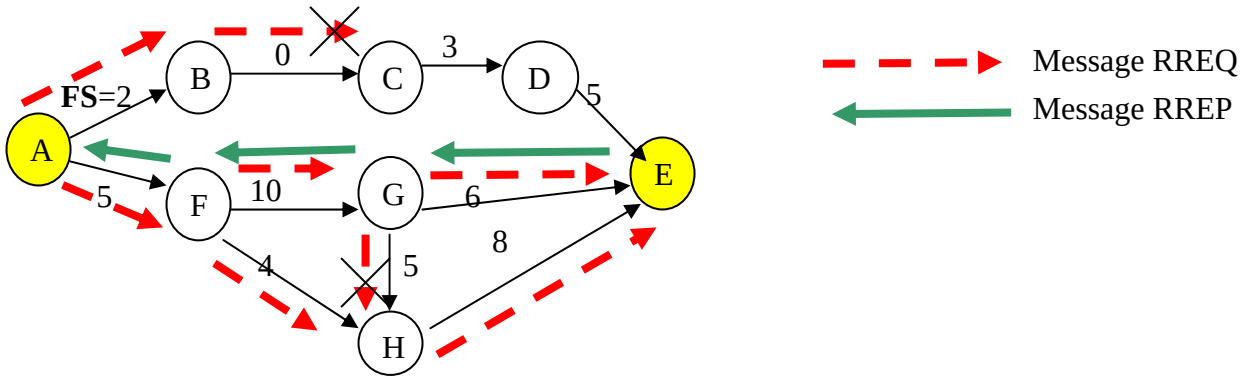


Figure 5.4 : Installation et recherche d'itinéraire utilisant la métrique basée sur les zones de propagation du signal.

5.3.2 Algorithme de choix d'itinéraire

Après la réception du premier message RREQ, La destination attend une durée de temps pour réceptionner plusieurs RREQ's. Selon L'algorithme ci après, la destination choisit un itinéraire stable sur lequel la source va envoyer les données, ainsi que d'autres itinéraires alternatifs pour une utilisation en cas d'échec de l'itinéraire en cour d'utilisation.

5.3.2.1 Cas de la première métrique

Un itinéraire est stable selon la première métrique si toutes les valeurs TS de ses liens sont à vraies.

La destination, et selon l'algorithme ci-après, choisit les itinéraires stables parmi les itinéraires des paquets RREQ enregistrés. Dans le cas où elle ne trouve aucun itinéraire stable, elle choisit des itinéraires non stables triés par ordre décroissant des valeurs de TR (Tick's Route).

La destination doit retourner un message RREP qui contient l'itinéraire à utiliser par la source (le plus stable ou celui qui a la plus grande valeur de TR) et d'autres itinéraires alternatifs.

Algorithme 5.3 Choix d'itinéraire utilisant la première métrique

```
Liste   Ensemble_stable : liste des ID's des noeuds
Liste   Ensemble_Non_Stable : liste des ID's des nœuds
Booléen   Chemin_Stable
Entier      Nb_Tick

Pour tous les itinéraires des RREQ's Faire
Début \* Début de parcours des entêtes des paquets RREQ *\
        Chemin_Stable = Vrai
        Tick_Nb  = 0

        Pour tous les nœuds de l'itinéraire Faire
        Début
                Si ST est à Faux Alors
                Debut
                    Chemin_Stable = Faux
                Fin
                Si Tick < Tick_Nb Alors
                Debut
                     Tick_Nb = Tick
                Fin
        Fin_Pour

Si Chemin_Stable Alors
            Ajouter le chemin à (Ensemble_stable)
        Sinon
           Ajouter le chemin à (Ensemble_Non_Stable, TR = Nb_Tick)
Fin     \* Fin de parcours des paquets RREQ *\
```

```
Si Taille (Ensemble_stable) >1 Alors
Début
    Ajouter_le_premier_chemin (Ensemble_Stable, RREP.Stable_Route)
    Ajouter_le_reste_des_chemins (Ensemble_Stable, RREP.Multipath)
Fin
Sinon
    Si Taille (Ensemble_stable) = 1 Alors
    Début
        Ajouter_le_premier_chemin (Ensemble_Stable, RREP.Stable_Route)
        Trier_par_ordre_décroissant_de_TR (Ensemble_Non_Stable)
        Ajouter_Ensemble_Non_Stable (RREP.Multipath)
     Fin
     Sinon
     Début
        Trier_par_ordre_décroissant_de_TR (Ensemble_Non_Stable)
        Ajouter_le_premier_chemin (Ensemble_Non_Stable,
                    RREP.Stable_Route)
        Ajouter_le_reste_des_chemins (Ensemble_Non_Stable,
                    RREP.Multipath)
     Fin
```

5.3.2.2 Cas de la deuxième métrique

La destination doit retourner un message RREP qui contient l'itinéraire le plus stable et d'autres itinéraires alternatifs selon la deuxième métrique.

Algorithme 5.4 Choix d'itinéraire utilisant la deuxième métrique

```
Liste   Ensemble_stable : liste des ID's des noeuds et FSM : Entier
                        /* FSM est la valeur minimale de tous les FS de l'itinéraire
Entier fs
Pour tous les itinéraires des RREQ's Faire
Début
        fs = FS du premier nœud de la liste LNI du RREQ
        /* fs est la valeur minimale de tous les FS de l'itinéraire
        Pour tous les nœuds de l'itinéraire Faire
        Début
                Si fs < FS Alors
                   fs = FS
        Fin
         FSM = fs
        Ajouter le chemin à (Ensemble_Stable, FSM)
Fin
Trier_par_ordre_décroissant_de_FSM (Ensemble_Stable)
Ajouter_le_premier_chemin (Ensemble_Stable, RREP.Stable_Route)
Ajouter_le_reste_des_chemins (Ensemble_Stable, RREP.Multipath)
```

5.4 Maintenance d'itinéraire

Si un nœud A, dans un réseau, ne reçoit plus de message Hello de son voisin B. cela signifie que ce dernier a émigré hors de la portée du nœud A. Dans ce cas, le nœud A consulte sa table de routage à la recherche d'une entrée, qui a comme prochain nœud, le nœud B. si une telle entrée est trouvée, le nœud A, lance une procédure de maintenance (rétablissement) de chemin. Cette dernière peut prendre un délai important avant qu'un autre chemin ne soit établi.

Pour cette raison, nous avons préféré prédire la rupture au lieu de la détecter. En utilisant la première métrique, et selon la définition de cette dernière, ce n'est pas possible de prédire la rupture des routes. Or dans la deuxième métrique, si un nœud A perçoit que le noeud voisin B rentre dans sa zone de rupture, il consulte sa table de routage à la recherche des entrées, qui ont comme prochain nœud, le nœud B. Si une telle entrée est trouvée, Le nœud A déclenche une procédure de rétablissement de route.

Dans notre approche, la procédure de rétablissement se compose de deux phases : une phase de réparation Locale et une autre de réparation de bout en bout (de la source à la destination).

5.4.1 Réparation locale

Le nœud, qui déclenche la procédure de maintenance, diffuse à ses voisins un paquet **Repair**. Ce dernier contient les champs suivants **(IDSRC, List_IDDEST, NSEQ, LNI, TTL).** Il a les mêmes champs avec le paquet RREQ à l'exception que ce paquet contient **List_IDDEST** au lieu de IDDEST. Le champ List_IDDEST contient tous les nœuds de la portion du chemin qui vient après lui (qui a envoyé le paquet Repair). Le champ **TTL** est initialisé à un petit nombre pour assurer que le paquet se propage localement (exemple : TTL = 2 ou 3). IDSRC est l'identificateur du nœud émetteur du paquet Repair, et le NSEQ est généré par ce nœud pour éviter les boucles.

Si un nœud reçoit le paquet Repair, il consulte la liste List_IDDEST. Si son Id n'appartient pas à cette liste, il décrémente le nombre de TTL par 1. Si le TTL est supérieur à 0, il insert son Id dans la liste LNI et rediffuse le message à ses voisins. Dans le cas où son Id appartient à la liste List_IDDEST, il répond par un message RREP comme le montre l'algorithme suivant :

Algorithme 5.5 Envoi de la réponse d'itinéraire dans le cas de la réparation locale

```
Liste Nœuds : liste des ID's des nœuds
Nœuds = liste_des_noeuds_qui_viennent_aprés_le_noeud (List_IDDEST, Id_self)
RREP.Stable_Route = Concaténé (IDSRC, LNI, Nœuds)
RREP.Multipath = Nil
```

Chaque nœud intermédiaire traite le paquet RREP comme dans la procédure d'installation de chemin décrite en dessus (met à jour sa table de routage, enregistre la portion du chemin qui vient juste après lui dans son cache, envoie le paquet RREP au nœud qui figure dans la liste Stable_Route juste avant lui).

Après la réception du paquet RREP, le nœud émetteur du paquet Repair met à jour sa table de routage et envoie les données provenant de la source sur la nouvelle portion du chemin vers la destination.

Si aucun message RREP n'est reçu par le nœud émetteur du paquet Repair dans un délai de temps prédéfini, il déclenche la procédure de réparation de bout en bout.

5.4.2 Réparation de bout en bout (de la source à la destination)

Dans le cas où la procédure de réparation locale échoue, le nœud, qui a détecté ou prédit la rupture de route, envoie un message ERROR unicast à la source pour l'informer de l'échec de l'itinéraire.

Après la réception du paquet ERROR, La source choisit un itinéraire parmi les itinéraires alternatifs stockés dans son cache. Pour s'assurer que ce dernier est toujours valide, elle envoie un message Trace (Stable_Route) unicast vers la destination. Chaque nœud intermédiaire qui reçoit ce message, le rediffuse au prochain nœud qui figure après lui dans la liste Stable_Route sans aucun traitement. Après réception du paquet Trace, la destination répond par un message RREP (Stable_Route, Nil). Les nœuds intermédiaires recevant le paquet RREP, le traite comme dans la procédure *d'installation du chemin* décrite ci-dessus.

Après la réception du paquet RREP (Stable_Route, Nil), La source commence l'envoi des données sur l'itinéraire Stable_Route.

Dans le cas contraire, si la source ne reçoit pas le paquet RREP dans un délai prédéfini, elle extrait de son cache un autre itinéraire alternatif et envoie une autre fois un message TRACE vers la destination.

Dans le cas où la source ne trouve aucun chemin alternatif dans son cache, elle lance une nouvelle procédure de recherche d'itinéraire.

5.5 Evaluation des performances

Dans ce qui suit, nous allons présenter les résultats de simulation de notre proposition en comparaison avec le protocole DSR. Nous présentons en premier lieu l'outil de simulation que nous avons adopté pour évaluer les performances de notre proposition qui est l'OPNET [72].

5.5.1 Environnement de simulation OPNet

L'outil de simulation OPNET [72] fournit un environnement complet de développement pour l'analyse de spécifications, de simulation et l'analyse des performances des réseaux de communication. Une gamme étendue des systèmes de communication du LAN simple aux réseaux satellites globaux est soutenue. Des simulations à événements discrets sont employées comme moyens d'analyser les performances des systèmes et de leurs comportements. Les caractéristiques principales d'OPNET sont les suivantes :

5.5.1.1 Cycle de modélisation et de simulation

Il y a cinq phases dans le cycle de modélisation et de simulation (voire Figure 5.5). OPNET fournit les outils spécifiques qui aident l'utilisateur dans trois de ces cinq phases (conception des modèles, l'exécution d'une simulation et l'analyse des données de sortie).

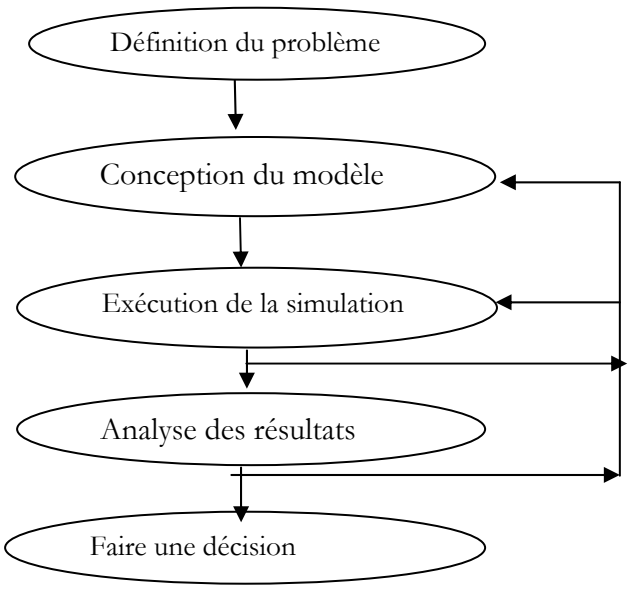


Figure 5.5 : Cycle de modélisation et de simulation

5.5.1.2 Modélisation hiérarchique

OPNET utilise une structure hiérarchique. Chaque niveau de la hiérarchie décrit différents aspects du modèle à simuler. Le modeleur d'OPNET est basé sur une série d'éditeurs hiérarchiques qui mettent en parallèle directement la structure des réseaux réels, équipements et protocoles.

Editeur de projet

L'éditeur de projet (Figure 5.6) représente graphiquement la topologie d'un réseau de communication. Les réseaux se composent de noeuds et de liens des objets, configurables par l'intermédiaire des fenêtres de dialogue. Pour créer un réseau, il suffit de faire glisser et déplacer les objets noeuds et liens de la palettes des objets de l'éditer pour concevoir un réseau. L'éditeur de projet nous permet l'emploie des objets de la bibliothèque étendue d'OPNET, ou la personnalisation des palettes selon nos besoins pour contenir nos propres modèles de noeud et de lien. Il fournit un contexte géographique, des caractéristiques physiques reflétées convenablement dans la simulation des réseaux filaires et sans files.

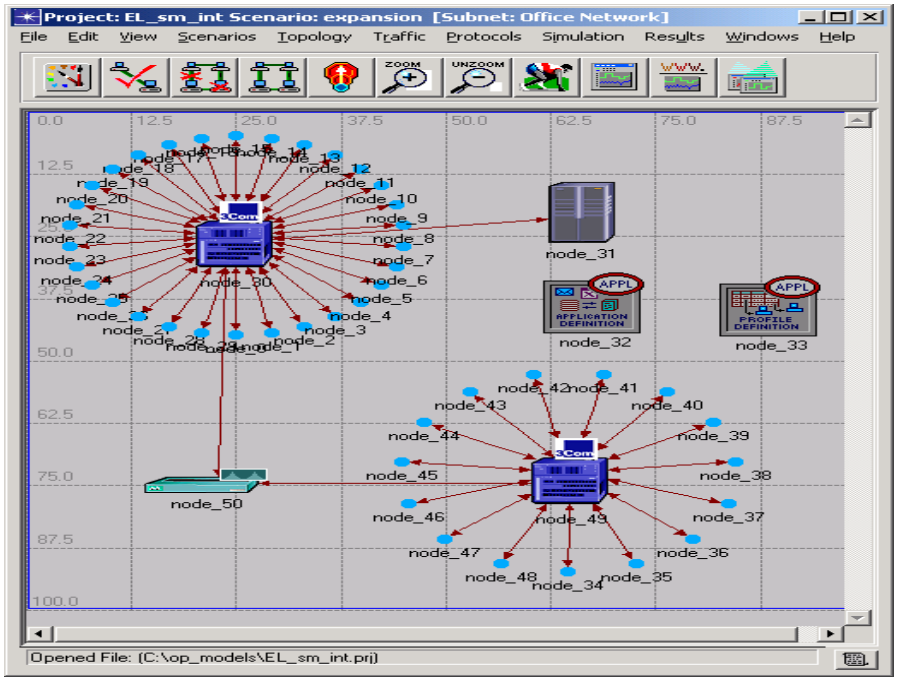


Figure 5.6 : Editeur de projet

Editeur de nœud

Cet éditeur affiche une représentation modulaire d'un élément de la bibliothèque ou d'un élément créé par l'utilisateur (Figure 5.7). Chaque module envoie et reçoit des paquets d'autres modules. Les modules représentent des applications, des couches protocolaires ou des ressources physiques (buffer, port, …). Des modèles de processus sont assignés aux

modules (développés dans l'éditeur de processus) pour réaliser tous les comportements demandés ou prévus.

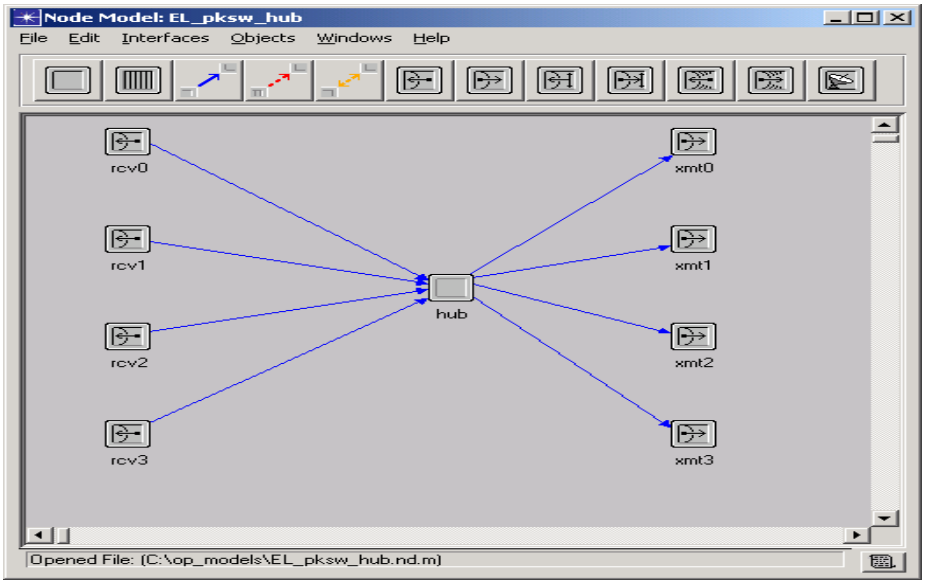


Figure 5.7 : Editeur de noeud

Editeur de processus

L'éditeur de processus est utilisé pour créer les modèles de processus (Figure 5.8). Ces derniers sont employés pour décrire l'écoulement logique et le comportement des modules constituant un nœud. La communication entre les processus est soutenue par des interruptions. Les modèles de processus sont exprimés avec un langage appelé Proto-C, qui se compose du diagramme d'état-transition, d'une bibliothèque des procédures (kernel procédures), et du langage de programmation standard de C. L'éditeur de processus d'OPNET emploie une approche puissante de diagramme d'état-transition pour supporter les spécifications de n'importe quel type de protocole, de ressource, d'application ou d'algorithme. Les états et les transitions définissent graphiquement la progression d'un processus en réponse aux événements. Dans chaque état, une logique générale peut être indiquée en utilisant une bibliothèque des fonctions prédéfinies et même de la flexibilité du langage C. Le processus peut créer de nouveau processus (processus enfant) pour exécuter des tâches secondaires (il s'appelle ainsi le processus parent).

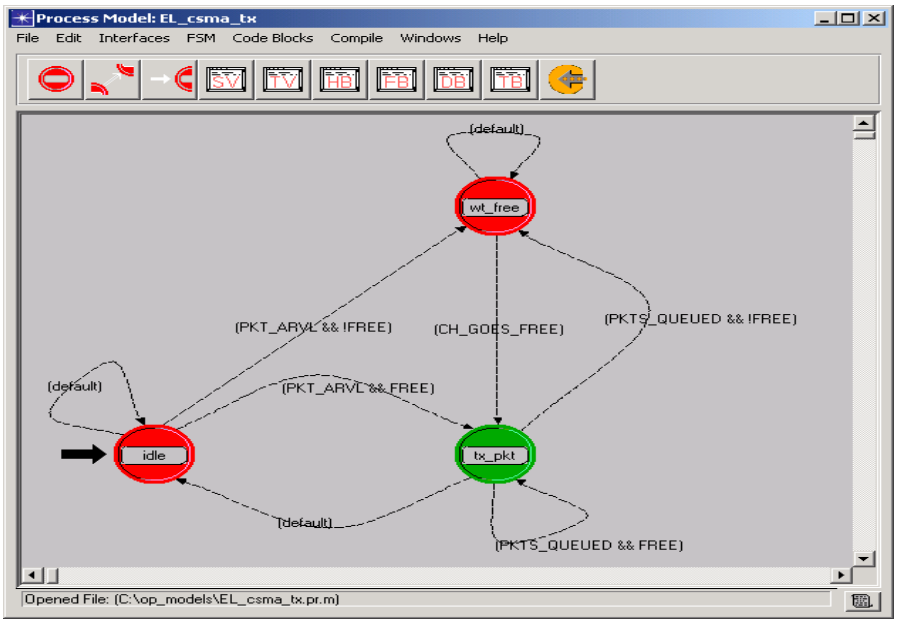


Figure 5.8. Editeur de processus

5.5.1.3 Exécution de la simulation

Après avoir défini tous les modèles du système de réseau, Nous pouvons le valider par la simulation afin d'étudier les performances et le comportement du système. Généralement, il y a deux étapes pour l'exécution de simulation et la collection de l'information :

Spécification des Données à collecter

Les modèles développés doivent toujours décider quelle information devrait être extraite à partir de la simulation. Ceux-ci peuvent prendre différentes formes comprenant des animations visuelles, séries dépendantes de valeurs de temps (vecteur), et des rapports paramétrés (scalaires).

Construction et Exécution De Simulation

L'exécution de la simulation est l'étape finale dans une "itération" d'une expérience modelante. En général, en se basant sur les résultats observés pendant cette étape, des changements sont faits aux spécifications du modèle ou aux paramètres de la simulation. OPNET fournit un certain nombre d'options pour exécuter les simulations, y compris l'exécution interne et externe, et la capacité de configurer les attributs qui affectent le comportement de la simulation.

5.5.2 Modèle de simulation

Nous allons décrire dans cette section les détails les plus importants, du modèle que nous avons conçu, pour évaluer notre proposition.

Modèle nœud

L'objet prédéfini d'OPNET ou le nœud dont lequel nous avons modifié sa couche réseau est le « Manet Station » (figure 5.9). Le modèle processus de notre proposition, tel que montré sur la figure 5.10, est présenté sous forme de diagramme d'état-transition contenant deux états qui sont les suivants :

Etat Init : Pour initialiser l'état et les paramètres (valeurs des variables) du nœud dans l'état initial (par exemple : initialiser toutes les valeurs des statistiques à récolter au cours de la simulation à 0, création de la table de routage, table de voisinages, etc.).

Etat wait : Cet état contient deux transitions vers lui-même :
La transition *PACKET_ARRIVAL* : la condition pour exécuter cette transition est l'arrivée d'un paquet de données ou de contrôle. L'exécution de cette transition fait appel à la fonction *My_Proposition_Packet_Arrival ()* que nous avons défini pour traiter le paquet reçu.

La transition *HELLO_TIMER_EXPIRY* : Cette transition est exécutée à chaque laps de temps (prédéfini dans l'état Init). L'exécution de cette transition fait appel à la fonction *send_hello_message ()* que nous avons défini pour diffuser un message Hello aux voisins.

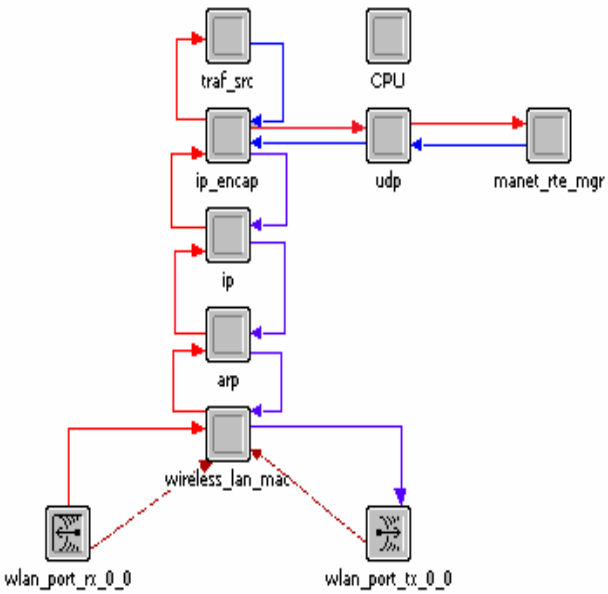


Figure 5.9. Modèle du noeud Manet Station

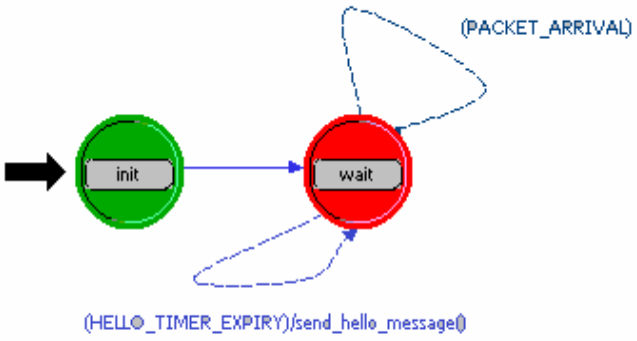


Figure 5.10 : Le modèle de processus de notre proposition

Modèle réseau

Le modèle réseau que nous avons conçu pour simuler et évaluer notre proposition se compose de 35 nœuds (les nœuds avec le modèle modifié cité ci-dessus) placer aléatoirement dans une zone de simulation de 2000 M sur 2000 M. Le modèle de mobilité que nous avons choisi est le modèle RWP (Random Waypoint), avec la vitesse des mobiles variées entre 0 et 25m/s, et le temps de pause est de 4 secondes.

5.5.3 Résultat de simulation

L'exécution de la simulation de notre modèle a été effectuée pendant une période de 300 secondes. Le nœud source commence l'envoi des données après la 100ème seconde pour permettre aux nœuds de s'initialiser, connaître leurs voisins et donner des valeurs aux métriques des liens initiaux.

Dans notre proposition, nous avons adopté deux métriques pour élire des chemins stables et durables. De ce fait, l'exécution de la simulation de notre proposition est lancée pour chaque métrique à part et avec différents paramètres comme suit :

La première métrique est inspirée des protocoles ABR et SSA. Dans notre simulation, nous supposons que la puissance d'un signal ne dépend que de la distance entre l'émetteur et le récepteur. Nous avons exécuté la simulation pour $SS_{threshold} = 200$ et pour $SS_{threshold} = 300$.

La deuxième métrique dépend des paramètres de la fonction de stabilité FS

$FS_{AB} = (a*1/Vm) + (b*Sens) + (C*NbHello) + (d*1/Local\_Zone)$

a,b,c et d sont des paramètres expérimentaux dépendant du modèle de mobilité. Les paramètres a et d sont multipliés par des nombres inférieurs à 1, d'où ils prennent des valeurs significativement plus grandes que les valeurs des paramètres b et c.

5.5.3.1 Métriques de performances

Les métriques de performances que nous avons choisi pour évaluer les performances de notre proposition sont les suivantes :

Rapport de la livraison de paquets (packet delivery ratio)

Ce paramètre mesure le taux de livraison de paquets ou bien le nombre de paquet de données qui est réceptionné par la destination avec succès. Ce paramètre est très important

pour mesurer les performances d'un protocole de routage et peut influencer significativement sur la comparaison des protocoles.

Délai de recherche d'itinéraire (search time of a route)

Avec ce paramètre, nous pouvons mesurer le temps consacré à une procédure de recherche d'itinéraire. Surtout dans les applications multimédia, le délai de réponse à une requête de communication ainsi que le temps de rétablissement d'itinéraire rompu ont une grande importance sur le choix de protocole de routage.

Le trafic de contrôle de routage généré ou l'overheads (Routing Traffic generated)

Chaque ressource dans le système, qui ne sera pas strictement nécessaire pour transmettre la charge utile de la communication, peut être considérée comme frais généraux et devra être réduite au maximum. Ce paramètre nous permet de calculer le trafic généré par les entités mobiles ou bien le nombre de paquets de contrôle de routage générés.

5.5.3.2 Analyse des résultats de simulation

Dans ce qui suit, nous présentons les résultats de l'évaluation des performances de notre proposition sous forme de graphes illustratifs. Pour faire une décision sur la qualité de ces performances, nous les comparons avec les performances d'un autre protocole très populaire retenu par l'IETF qui est le protocole DSR (Heureusement que ce protocole est déjà implémenter dans OPNET 11.5). La comparaison est faite entre notre proposition, avec la première métrique et la deuxième métrique chacune à part, et le protocole DSR.

La figure 5.11 illustre le rapport de livraison de paquets de données entre deux entités communicantes. Nous pouvons observer, dans ce graphe, clairement que le rapport de livraison des paquets dans notre proposition dans les deux cas (cas de deux métriques) est plus grand que dans le DSR. Ainsi le rapport de livraison des paquets dans le cas de la deuxième métrique de notre proposition avec les paramètres (a=100, b=5, c=0.1, d=100) est le même dans le cas de la première métrique avec les paramètres (SSthreshold = 200 et click_threshold = 5). Nous pouvons également observer que le rapport de livraison des paquets, dans le cas de la deuxième métrique avec les paramètres (a=200, b=1, c=0.1, d=200), assure une grande livraison des paquets comparés aux autres.

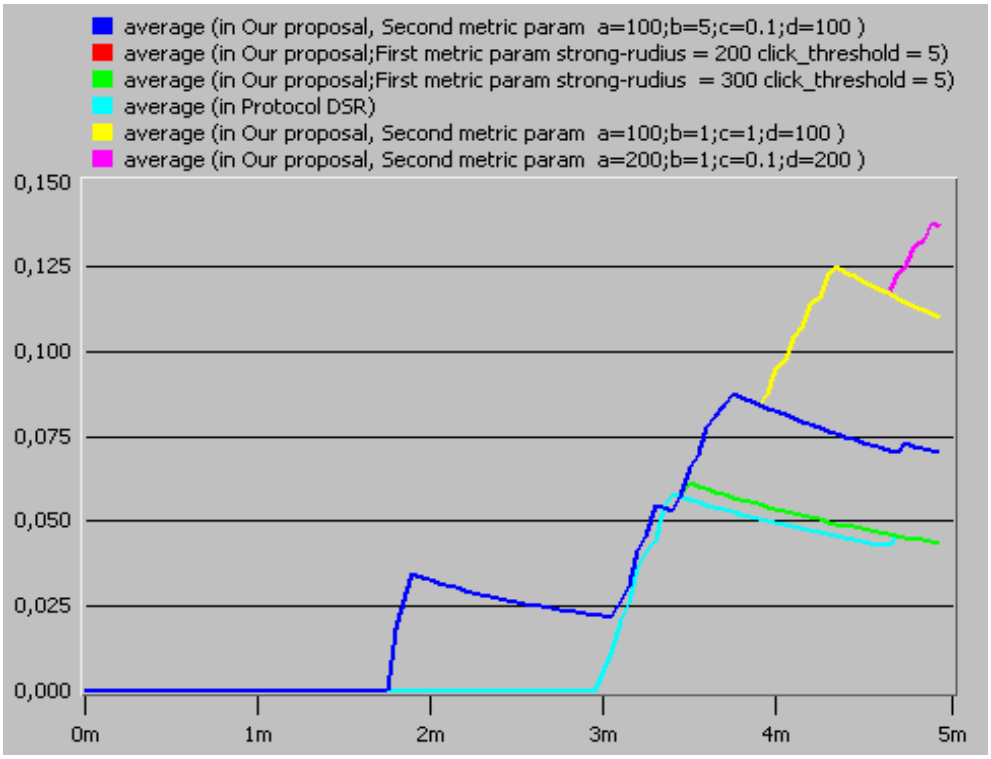


Figure 5.11 : Le rapport de livraison de paquets de données

La figure 5.12 illustre le délai de recherche d'itinéraire entre deux entités mobiles communicantes, ainsi que le temps de rétablissement d'un itinéraire rompu. Nous constatons que le temps de recherche dans le protocole DSR est le plus grand comparé au temps de recherche dans notre proposition dans les deux métriques. Ceci est en raison de la politique de la réparation locale et au principe du multi-chemin qui accélère la réparation et l'installation d'itinéraire. Ainsi notre proposition cherche et installe les itinéraires les plus stables et les plus durables comparé au DSR qui installe le premier itinéraire (le plus court) trouvé. Nous pouvons aussi observer que le délai de recherche employant la première métrique est plus grand que le temps de recherche en utilisant la deuxième métrique. Cela prouve que les itinéraires établis dans le cas de la deuxième métrique sont plus stables et plus durables comparés aux itinéraires établis en utilisant la première métrique.

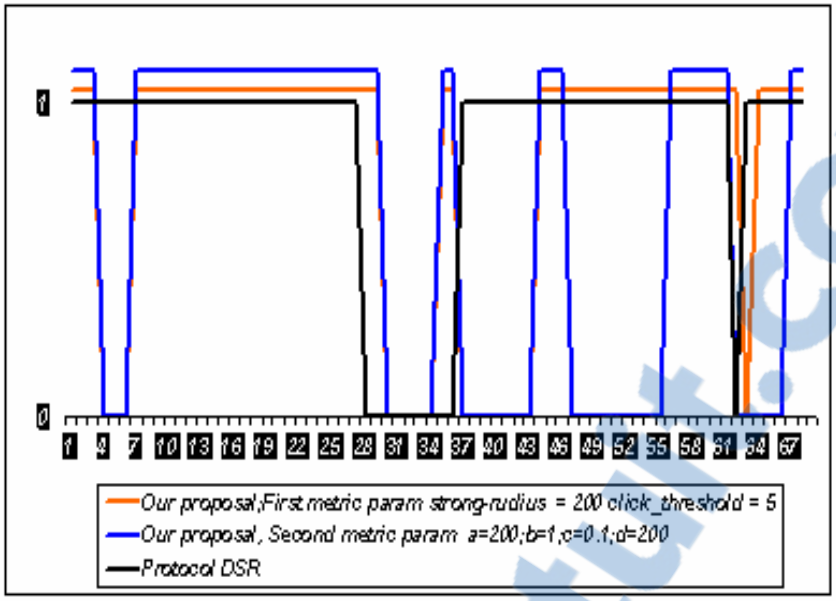


Fig. 5.12 : Le délai de recherche d'itinéraire

La figure 5.13 illustre le trafic de contrôle de routage généré. Nous pouvons observer que notre proposition dans les deux métriques présente un très grand trafic généré par rapport au protocole DSR. Cela est dû à la diffusion périodique des messages Hello dans notre proposition.

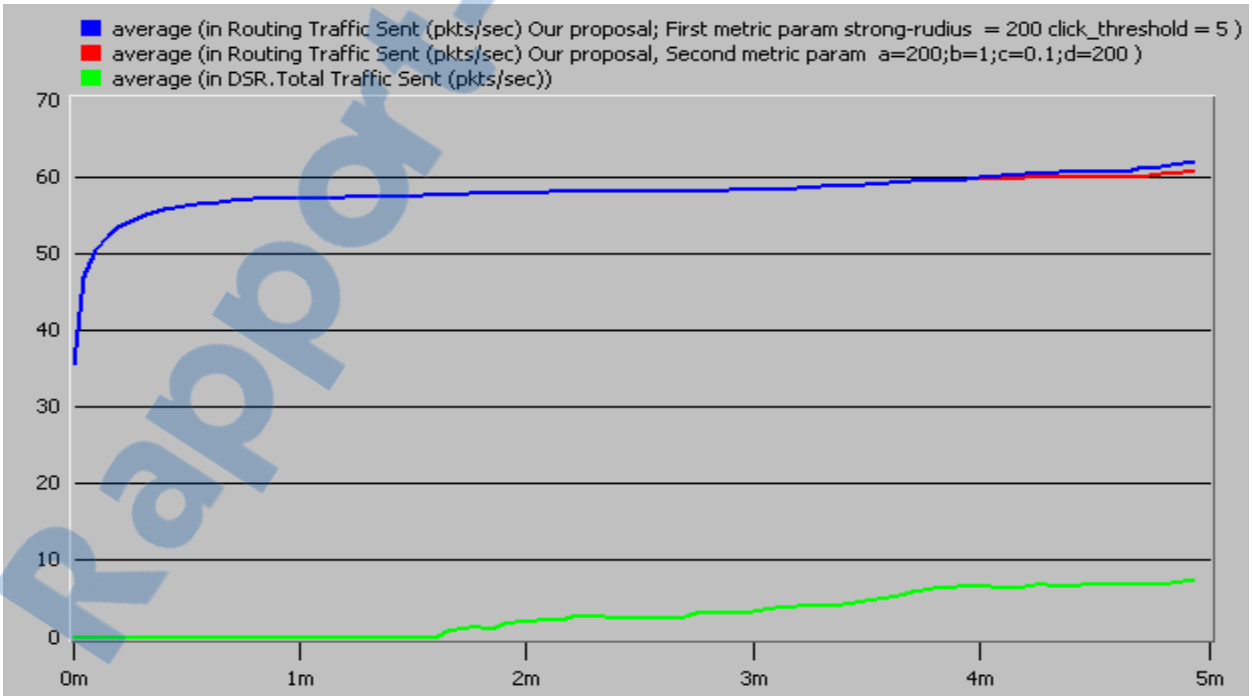


Figure 5.13 : le trafic de contrôle de routage généré

5.6 Conclusion

Du fait de la forte dynamique des nœuds dans le contexte des réseaux ad hoc, les routes sont souvent instables et les ressources au niveau des terminaux sont en constante variation, du fait de leur participation au service de routage. Les informations d'état de lien utilisées par les protocoles de routage traditionnels peuvent alors devenir rapidement obsolètes à cause du mouvement rapide des nœuds. Par conséquent, les ruptures de routes sont beaucoup plus fréquentes et doivent être recalculées très souvent, ce qui provoque évidemment des interruptions de connexions et une perte de qualité au niveau des applications.

Pour maintenir des communications stables dans les réseaux ad hoc avec une certaine qualité de service rendue aux applications, nous avons proposé dans ce chapitre des nouvelles métriques pour élire des chemins stables entre les entités communicantes. Nous avons aussi proposé des mécanismes de prédiction de rupture de route, ainsi que des mécanismes pour maintenir ces routes localement ou de bout en bout (de la source à la destination).

Après l'analyse des résultats de simulation pour trois paramètres de performances que nous avons jugés les plus importants pour évaluer notre proposition et après la comparaison des résultats de simulation avec les performances du protocole DSR. Nous pouvons conclure que notre proposition améliore considérablement le délai d'établissement d'itinéraire et le délai de réparation d'itinéraires rompus. Nous pouvons conclure aussi que notre proposition améliore la qualité des itinéraires installés, en itinéraires plus stables et plus durables (surtout dans le cas de la deuxième métrique). Le prix de cette amélioration est le trafic de contrôle généré qui est significativement plus grand que le trafic de contrôle généré par le protocole DSR.

Conclusion et perspectives

Les réseaux sans fil ad hoc sont des réseaux ne disposant d'aucune infrastructure préexistante et formée de nœuds mobiles interconnectés par des liaisons sans fil. Leurs architectures évoluent au gré de l'apparition et du mouvement des noeuds. L'absence d'infrastructure se traduit par la nécessité de mettre en place des solutions adaptées reposant sur la participation de l'ensemble des nœuds formant le réseau ad hoc.

Dans ce mémoire, nous avons présenté un panorama des protocoles de routage, les plus connus, ainsi que leurs principales caractéristiques et fonctionnalités qui permettent d'assurer l'acheminement des données dans le réseau mobile.

Nous avons donné un aperçu sur les classifications les plus connues des protocoles de routage et les caractéristiques de chaque classe. Les classifications citées n'ont pas de structure sous forme d'arbre, bien que certaines caractéristiques de ces classifications soient en général communes entre les classes. D'autres classes citées sont d'une nature d'opposition.

Le travail de ce mémoire a porté sur le routage dans les réseaux ad hoc qui constitue un thème de recherche important. En effet, du fait de la forte dynamicité des nœuds dans le contexte des réseaux ad hoc, les routes sont souvent instables et les informations d'état de lien utilisées par les protocoles de routage traditionnels peuvent alors devenir rapidement obsolètes. Par conséquent, les ruptures de routes sont beaucoup plus fréquentes et doivent être recalculées très souvent, ce qui provoque évidemment des interruptions de connexions et une perte de qualité au niveau des applications. Pour ces raisons, nous avons proposé des nouvelles métriques pour élire des chemins stables entre les entités communicantes. Nous avons aussi proposé des mécanismes de prédiction de rupture de route, ainsi que des mécanismes pour maintenir ces routes localement ou de bout en bout (de la source à la destination).

Pour démontrer l'efficacité de ces nouvelles approches, nous l'avons simulé et comparer avec le protocoles DSR en utilisant l'outil Opnet 11.5.

L’analyse des résultats de simulation pour trois paramètres de performances : le rapport de livraison de paquets de données, le délai de recherche d’itinéraire et le trafic de contrôle de routage généré, nous a démontré que notre proposition améliore considérablement le délai d’établissement d’itinéraire et le délai de réparation d’itinéraires rompus. Ainsi qu'elle améliore la qualité des itinéraires installés, en itinéraires plus stables et plus durables (surtout dans le cas de la deuxième métrique). Le prix de cette amélioration est le trafic de contrôle généré qui est significativement plus grand que le trafic de contrôle généré par le protocole DSR.

Les mécanismes que nous avons proposés, peuvent être implémentés sur des protocoles de routages existants dont le but de tester l'efficacité de ces mécanismes sur les déférents protocoles.

Bibliographie

1- J. Liu and I.Chlamtac, "Mobile ad hoc networking," Chapter 1, IEEE Press 2004.

2- The IEEE 802.16 Working Group Web Site: http://www.ieee802.org/16.

3- Supplement to IEEE Standard for Information technology-Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band. June 2003.

4- IEEE Standard for Information technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band. June 2003.

5- Information technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. June 2003.

6- ETSI—BRAN, "ETSI HIPERLAN 1 Standards,"
http://www.etsi.org/frameset/home.htm?/technicalactiv/Hiperlan/hiperlan1.htm.

7- ETSI—BRAN, "ETSI HiperLAN 2 Standards,"
http://www.etsi.org/frameset/home.htm?/technicalactiv/Hiperlan/hiperlan2.htm.

8- The official Bluetooth web site: http://www.bluetooth.com/.

9- IBM Zurich Research Laboratory web site:
http://www.zurich.ibm.com/cs/wireless/bluetooth.html.

10- J.A. Freebersyser and B. Leiner, " Ad-hoc networking, " chapter A DoD Perspective on Mobile Ad Hoc Networks in Charles E. Perkins Ad-hoc networking. London, Addison Wesley, 2001.

11- D. Dhoutaut, « Etude du standard IEEE 802.11 dans le cadre des réseaux ad hoc : de la simulation à l'expérimentation ,» Thèse de doctorat , Projet INRIA ARES, Laboratoire CITI, INSA de Lyon, Décembre 2003.

12- M. S. Corson and J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, IETF, January 1999.

13- L.M. Feeney, "A taxonomy for routing protocols in mobile ad hoc networks" Tech. Rep., Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden http://www.sics.se/ lmfeeney/, October 1999.

14- E.M. Royer and C.K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," IEEE Personal Communications Magazine, 46–55, April 1999.

15- M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position based routing in mobile ad-hoc networks," IEEE Network Magazine, vol. 15, no. 6, pp. 30-39, Nov. 2001.

16- G. S. Malkin and M. E. Steenstrup. "Distance-Vector Routing," in M. Steenstrup (Ed.), Routing in Communications Networks, pp. 83–98. Prentice-Hall, 1995.

17- J. Moy. "Link-State Routing," in M. Steenstrup (Ed.), Routing in Communications Networks, pp. 135–157. Prentice-Hall, 1995.

18- M. Jiang, J. Li, and Y.C. "Tay Cluster Based Routing Protocol," IETF Draft, Aug. 1999, http://www.ietf.org/internet-drafts/draft-ietf-manet-cbrp-spec-01.txt.

19- C. E. Perkins and P Bhagwat, "Highly dynamic destination sequenced distance-vector routing (dsdv) for mobile computers, " in ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, 1994, pp. 234-244.

20- V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," in Proceedings of IEEE INFOCOM, 1997.

21 – P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot and T. Clausen, "Optimized Link State Routing Protocol (OLSR) ," IETF Draft, Oct. 2001.

22- B. Bellur. and R.G. Ogier, "A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks," Proceedings IEEE INFOCOM, Mar. 1999.

23 - C.K. Toh, "A Novel Distributed Routing Protocol to Support Ad-Hoc Mobile Computing," in Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communication ,pp. 480–486, March 1996.

24- Y.B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," In Proceedings of the 4th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pp. 66–75, Dallas, Texas, October 1998.

25- R. Dube, C.D. Rais, K-Y. Wang, and S. K. Tripathi, "Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks," IEEE Personal Communications, February 1997.

26- R. V. Boppana and S. Konduru, "An adaptive distance vector routing algorithm for mobile ad hoc networks" in Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, 2001, vol. 3, pp. 1753-1762.

27- Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," IETF Draft, draft-ietf-manet-zone-zrp-02.txt, June 1999

28- G. Pei M. Gerla, and X. H. LANMAR, "A Wireless Hierarchical Routing Protocol with Group Mobility " in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), pp. 1538–1542, New Orleans, September 1999.

29- H. Labiod N. Nikaein and C. Bonnet, "Ddr-distributed dynamic routing algorithm for mobile ad hoc networks," Tech. Rep., Institut Eurecom, France, 2000.

30- G. Pei, M. Gerla, and T.W. Chen. "Fisheye State Routing in Mobile Ad Hoc Networks" in Proceedings of the 2000 ICDCS Workshops, pp. D71–D78, Taipei, Taiwan, April 2000.

31- C. N. Sekharant-N.S.V. Rao Steven G. Batsell S. Radhakrishnan, Gopal Racherlat, "Dst - a routing protocol for ad hoc networks using distributed spanning trees" IEEE Wireless Communications and Networking Conference, pp: 100-104, 1999.

32- C. E. Perkins, E. M. Royer and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, July 2003, http://www.ietf.org/rfc/rfc3561.txt,

33- S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," in Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiC'98, Dallas, TX, October, 1998.

34- D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," in Ad Hoc Networking, C. E. Perkins (Ed.), pp. 139–172, Addison-Wesley, 2001

35- W. Su and M. Gerla, "IPv6 flow handoff in ad-hoc wireless networks using mobility prediction," In Proceedings of IEEE GLOBECOM'99, Rio de Janeiro, Brazil, December 1999.

36- J. Y. YU and P. H. J. CHONG, "A Survey of Clustering Schemes for Mobile Ad Hoc Networks," IEEE Communications Surveys and Tutorials, Vol. 7, No. 1, pp. 32--48, First Quarter 2005.

37- E. Kaplan, "Understanding GPS" Artech House, 1996.

38- S. Capkun, M. Hamdi, and J. Hubaux, "Gps-free Positioning in Mobile Ad Hoc Networks" Proc. Hawaii Int'l. Conf. System Sciences, Jan. 2001.

39- J. Hightower and G. Borriello "Location Systems for Ubiquitous Computing" IEEE Computer, vol. 34, no. 8, Aug. 2001, pp. 57–66.

40- J. C. Navas and T. Imielinski, "Geographic Addressing and Routing" Proc. 3rd ACM/IEEE Int. Conf. Mobile Comp. Net., MobiCom'97, Sept. 1997.

41- M. Mauve , J. Widmer and H. Hartenstein, " A Survey on Position-Based Routing in Mobile Ad Hoc Networks," IEEE Network Magazine, vol. 15, no.6, November/December 2001, pp. 30-39.

42- M. Marina and S. Das. "On-demand Multipath Distance Vector Routing in Ad Hoc Networks" in Proceedings of the International Conference on Network Protocols (ICNP), Riverside, CA, November 2001.

43- S.J. Lee and M. Gerla. "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," in Proceedings of the IEEE International Conference on Communications (ICC), pp. 3201–3205, Helsinki, Finland, June 2001.

44- P. Annamalai "Comparative Performance Study of Standardized Ad Hoc Routing Protocols and OSPF-MCDS," Master thesis, Blacksburg, Virginia, October 2005.

45- M. Abolhasan, T. Wysocki and E. Dutkiewicz "A review of routing protocols for mobile ad hoc networks" Elsevier, Ad Hoc Networks 2 (2004), PP 1–22.

46- A. Quintero, S. Pierre and B. Macabéo "A routing protocol based on node density for ad hoc networks" Elsevier, Ad Hoc Networks 2 (2004), PP335–349.

47- E.M. Bending-royer "Mobile ad hoc networking" Chapter 10, IEEE Press 2004.

48- A. Iwata, C. C. Chiang, G. Pei, M. Gerla, and T.W. Chen. "Scalable routing strategies for ad hoc wireless networks," IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks, pp.1369-79, August 1999.

49- G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," ACM/Kluwer Wireless Networks, 8(2-3):275–288, 2002.

50- C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," IEEE Personal Communications Systems (PCS) Magazine, special issue on Mobile Ad Hoc Networks, vol 8, no1, pp 16–29, February 2001.

51- Y-C. Hu and D. Johnson, " Caching strategies in On-demand Routing Protocols for Wireless Ad Hoc Networks," In Proceedings of the Sixth Annual IEEE/ACM International Conference on Mobile Computing and Networking (MobiCom 2000), pp. 231–242, Boston, MA, August 2000.

52- M. K. Marina and S. R. Das, "Performance of Route Caching Strategies in Dynamic Source Routing" In Proceedings of the Int'l Workshop on Wireless Networks and Mobile Computing (WNMC) in conjunction with Int'l Conf. on Distributed Computing Systems (ICDCS), pages 425-432, 2001.

53- Y. C. Hu and D. Johnson, "Ensuring Cache Freshness in On-demand Ad Hoc Routing Protocols" In Proceedings of Int'l Workshop on Principles of Mobile Computing (POMC), pages 25–30, 2002.

54- M.K. Marina and S.R. Das, "Ad Hoc Networks Technologies and Protocols ," chapter 3, Springer Science 2005.

55- ETSI. Broadband Radio Access Networks (BRAN) ; HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1 ; Functional specification. Technical report, July 1998.

56- P. Jacquets, P. Minet, P. Mühlethaler, and N. Rivierre. "Increasing reliability in cable-free radio LANs - Low Level Forwarding in HIPERLAN" Wireless Personal communication, 4(1), 1997.

57- A. Qayyum, L. Viennot, and A Laouiti, "Multipoint Relaying : An Efficient Technique for Flooding in Mobile Wireless Networks". Rapport de recherche de l'INRIA RR-3898, Equipe : HIPERCOM, Mars 2000.

58- A. Qayyum, "Analysis and evaluation of channel access schemes and routing protocols for wireless networks," Thèse de doctorat en Informatique, Université PARIS SUD-PARIS XI, 2000.

59- E. Gafni and D. Bertsekas, "Distributed Algorithms for Generating Loopfree Routes in Networks with Frequently Changing Topology," IEEE Transactions on Communications, 29(1): pp 11–18, 1981.

60- J. Broch, D. Maltz, D. Johnson, Y-C. Hu and J. Jetcheva. "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols" In Proceedings of IEEE/ACM MobiCom, pages 85–97, 1998.

61- S. R. Das, R. Castaneda, and J. Yan, "Simulation-based Performance Evaluation of Routing Protocols for Mobile Ad hoc Networks," ACM/Baltzer Mobile Networks and Applications (MONET), 5(3): pp 179–189, 2000

62- Ki-Il Kim and Sang-Ha Kim "Establishing Measurement-Based Reliable Path in Mobile Ad Hoc Networks," IEEE Communications Society / WCNC, 2005.

63- A.Boukerche and L.Bononi, "Mobile ad hoc networking," Chapter 14, IEEE Press 2004.

64- R. Jain, "The Art of Computer Systems Performance Evaluation," Wiley, New York, 1991.

65- S. Corson and J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, Jan. 1999.

66- M. Gerla, K. Tang, and R. Bagrodia, "TCP Performance in Wireless Multi-hop Networks" in Proceedings of IEEE WMCSA'99, New Orleans, pp: 41-50, LA, February 1999.

67- M. Sanchez and P. Manzoni, "A java-based ad hoc networks simulator," In Proceedings of the SCS Western Multiconference Web-based Simulation Track, Jan. 1999.

68- T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," Wireless Communications and Mobile Computing, Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol 2, no5, September 2002.

69- C. Bettstetter, "Smooth is Better than Sharp: a Random Mobility Model for Simulation of Wireless Networks," in Proceedings of ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'01), Rome, Italy, July 2001

70- E. Royer, P.M. Melliar-Smith, and L. Moser, "An analysis of the optimum node density for ad hoc mobile networks," In Proceedings of the IEEE International Conference on Communications (ICC), 2001.

71- D. Lang, "A comprehensive overview about selected Ad Hoc Networking Routing Protocols, " Technical Report I0311, 2003, TU Munich, March 2003.

72- OPNET Simulator (Version 11.5). Website : www.opnet.com.