REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDERAHMANE MIRA DE BEJAIA
FACULTE DES SCIENCES EXACTES

DEPARTEMENT D'INFORMATIQUE
ECOLE DOCTORALE RESEAUX ET SYSTEMES DISTRIBUES

Mémoire de Magistère

En vue de l'obtention du diplôme de magistère en informatique

Option : Réseaux et Systèmes Distribués

Thème

Routage avec *QoS* dans les réseaux mobiles Ad-Hoc

Présenté par

BOUDAA Abdelghani

Devant le jury composé de :

Président :	KHELFAOUI	Youcef	M.C	Université A. Mira, Bejaïa.
Rapporteur :	BOUKERRAM	Abdallah	M.C	Université F. Abbas, Sétif.
Examinateur :	ALIOUAT	Makhlouf	M.C	Université F. Abbas, Sétif.
Examinateur :	MELIT	Ali	M.C	Université de Jijel
Invité :	MOUSSAOUI	Ali	M.A	Centre universitaire de B.B.A

Promotion 2008-2009

DEDICACES

Je voudrais dédier le fruit de mon travail :

A la mémoire de mes parents et, mon beau-père

A mes enfants A/Hamid, Riadh, Ikram et Djamel-Eddine

A ma femme, pour sa patience et son soutien indéfectible

A ma belle-mère qui m'a tant soutenu

A mes sœurs ainsi qu'à leurs maris

A mes frères Hicham, Mokhtar et Adel ainsi qu'à leurs femmes

A tous mes Amis dont la liste est longue

A tous ceux qui m'ont aidé à achever ce mémoire.

REMERCIEMENTS

En premier lieu, je remercie le Bon Dieu de m'avoir donné la force et le courage pour accomplir ce travail et qui m'a procuré ce succès.

Je suis profondément reconnaissant à Mr Abdallah BOUKERRAM de Conférences à l'UFAS de Sétif, et Mr Ali MOUSSAOUI Maître Assistant au C. Universitaire de BBA pour avoir diriger ce travail, pour leurs orientations et conseils.

Je remercie également les membres du jury pour l'immense honneur qu'ils me font en acceptant d'évaluer ce travail, et leurs précieux temps qu'ils me consacrent.

Que toute personne ayant contribué, d'une manière ou d'une autre, à l'aboutissement de ce travail, trouve ici l'expression de mes sincères reconnaissances.

Je remercie sincèrement ma femme, dont l'aide et l'encouragement m'ont permis de continuer mes études et de préparer ce mémoire.

RESUME

Un réseau Ad-Hoc est un mode de fonctionnement particulier des réseaux sans fil, c'est donc une collection de nœuds mobiles à topologie variable et communicants entre eux sans point d'accès et sans administration centralisée. Ce type de réseaux est très facile à déployer et leurs utilisations semblent beaucoup plus étendues. Les communications multi-sauts y sont possibles grâce à des protocoles de routage spécifiques. Cependant, le processus de routage devient une tache complexe dans le cas ou certaines contraintes de qualité de services sont rajoutées au niveau des transmissions.

Nous avons étudié différents protocoles de routage, pour améliorer le routage dans les réseaux *Ad-Hoc* et afin de prendre en compte les contraintes stabilité des liens. Finalement, nous avons choisi d'utiliser le protocole de routage *AODV* (*Ad-hoc On-Demand Distance Vector*) car il présente un nombre important de caractéristiques qui s'adaptent aux réseaux sans fil.

Suite à une modification de ce protocole, plusieurs optimisations ont été apportées pour améliorer ses performances. *AODV* adopte la métrique de nombres de sauts qui n'est pas toujours la solution optimale, ainsi la route à tendance à contenir des liens fragiles et offre de faibles performances. Pour résoudre ce problème, on a utilisé la puissance du signal reçu pour la sélection d'itinéraires et pour la prédiction des ruptures de liens. En préférant les liens avec une plus grande puissance du signal, alors les chemins convergent progressivement à l'optimum, donc l'itinéraire sélectionné possède une forte stabilité et une plus grande durée de vie. *L'AODV* modifié est testé sous le simulateur « *OPNET Modeler »* qui stipule parfaitement bien que ce dernier apporte une qualité de service indéniable par rapport à *l'AODV* original.

Mots-clés : Réseaux Ad-Hoc, protocoles de routage, stabilité des liens, contrôle d'admission, qualité de service (*QoS).*

ABSTRACT

An Ad-Hoc network is a particular mode of operation of wireless networks, so it's a collection of mobile nodes in variable topology and communicating with each other without access point and without centralized administration. This type of network is very easy to deploy and use seem to be much broader. Communications multi hops are possible through specific routing protocols. However, the routing process becomes a complex task in case of certain quality of service are added at the transmission.

We have study different routing protocols to improve the routing in the Ad-Hoc networks and to take care of the links stability constraints. Finally, we have chosen the use *AODV* (*Ad-hoc On-Demand Distance Vector*) routing protocol because it has a large number of features that adapt to wireless networks (*MANET*).

In this work, the protocol was changed and several optimizations have been added to improve its performance. *AODV* adopts the metric of the number of hops which is not always the optimal solution, and the road tends to contain links fragile and offers poor performance. To resolve this problem, we used the signal received power for the selection of routes and the prediction of links failures. In using the links with greater strength of the signal, then the paths gradually converge to the optimum, so the selected route has a high stability and greater durability. We evaluated the modified *AODV* with the « *OPNET Modeler* » simulator and found that it takes a great improvement and quality of service compared to the original *AODV*.

Keywords: Ad-Hoc networks, routing protocols, links stability, admission control, quality of service (*QoS*).

TABLE DES MATIERES

Chapitre 3 : Qualité de service dans les réseaux Ad-Hoc

Chapitre 4 : Routage suivant les liens stables

Chapitre 5 : Etude et simulation du protocole *AODV-SI*

Liste des figures

Liste des tableaux

Introduction générale

Introduction générale

Au cours de la dernière décennie, le nombre de dispositifs qui ont un moyen de communication sans fil a énormément augmenté et la plupart de nos habitudes se basent sur ces derniers. En effet, tout le monde se sert de téléphones mobiles, certains agendas numériques (*Notebook*) ou des ordinateurs portables (*Laptop*) et les sociétés investissent de plus en plus dans des réseaux sans fils. En parallèle, plusieurs technologies comme le *Bluetooth*, *GPRS*, *WiMax* et *WiFi (Wireless Fidelity)* se sont développées et permettent actuellement le déploiement d'une infrastructure de communication sans fil performante.

Les réseaux sans fils fonctionnant en mode Ad-Hoc sont flexibles et faciles à être déployés. Il représente un ensemble de terminaux indépendants formant un *IBSS* (*Independent Basic Service Set*), dont le rôle consiste à permettre aux stations de communiquer sans l'aide d'une quelconque infrastructure. Cependant, il existe encore de nombreuses régions où cette infrastructure n'est pas disponible par exemple : dans les opérations d'aide en cas de catastrophe, les forces de secours se trouvent parfois dans l'obligation de créer temporairement un réseau autonome (Ad-Hoc). Dans une telle situation, les appareils mobiles fonctionnent de façon auto-organisée et sont en mesure de communiquer directement avec leurs voisins. Pour atteindre une destination éloignée, un nœud dépend de la capacité d'autres nœuds à propager son message.

L'un des importants et célèbres groupes de développement de réseaux mobiles Ad-Hoc est *MANET* [2] (*Mobile Ad-Hoc network Group*). Ainsi, de nombreux protocoles de routage ont été conçus pour la découverte et la maintenance des routes. La stratégie de routage est utilisée dans le but de découvrir les chemins qui existent entre les nœuds. Le but principal d'une telle stratégie est l'établissement de routes qui soient correctes et efficaces entre une paire quelconque d'unités de communication, ce qui assure l'échange des messages d'une manière continue.

Deux principales stratégies de routage sont utilisées dans les réseaux mobiles Ad-Hoc. Ainsi, il existe ceux qui sont basés sur l'état des liens et ceux, basés sur le vecteur de distance. Les deux méthodes exigent une mise à jour périodique des données de routage qui doivent être diffusées par les différents nœuds de routage du réseau.

Dans un réseau mobile Ad-Hoc, les problèmes sont encore plus compliqués à résoudre. Ils sont confrontés aux interférences radios, ont une plus petite bande passante que les réseaux filaires et ils ont une puissance de calcul et d'énergie limités. Enfin, la mobilité des nœuds et sa topologie dynamique compliquent la communication. Ces complications engendrent des obstacles pour le routage dans ces réseaux. Les limitations des réseaux Ad-Hoc rendent difficile la découverte et la maintenance des routes. La mobilité des nœuds et le changement fréquent de la topologie dans les réseaux Ad-Hoc réduisent énormément la découverte des routes fiables et stables.

Dans les réseaux Ad-Hoc, les recherches actuelles sont dirigées vers les algorithmes de routage. En effet, la plupart des protocoles existants ne se préoccupent que d'un seul paramètre pour trouver le chemin d'une source vers une destination : le nombre de sauts. Or, dans la plus part des situations, cette mesure se traduit par des chemins avec des liaisons fragiles entrainant des pertes de paquets et des changements de routes fréquents. Une des solutions possibles pour tenir compte de la qualité des routes est de se baser sur les informations de la couche physique, à savoir, la puissance du signal reçue ou la distance qui nous sépare de l'émetteur (longueur du lien).

Pour fournir un routage efficace avec une certaine qualité de service dans les réseaux Ad-Hoc, il faut tout d'abord comprendre les différents mécanismes de base des réseaux sans fils et les caractéristiques de ce type de réseaux. Nous devons notamment comprendre les différents algorithmes et protocoles de routage afin de parvenir à identifier la meilleure approche du point de vue de la transmission des paquets.

Notre objectif est de modifier ou d'effectuer une extension d'un protocole existant afin de prendre en compte la stabilité des liens, l'augmentation du trafic délivré et la réduction des pertes de paquets de données. Nous allons implémenter l'extension du protocole dans le simulateur *OPNET Modeler* qui permet de modifier rapidement les protocoles standards existants (*AODV*, *DSR*, *TORA*, *OLSR*). Les simulations effectuées vont nous permettre de vérifier si les résultats obtenus du protocole sont satisfaisants.

Notre travail dans ce mémoire est organisé comme suit :

- Dans le premier chapitre, nous présentons les étapes d'évolution des réseaux sans fils pour aboutir ensuite à la définition et la description des caractéristiques spécifiques des réseaux Ad-Hoc.

- Dans le deuxième chapitre, nous décrirons les stratégies de routage et les différents modes de communication dans les réseaux Ad-Hoc : nous étudions un panorama de protocoles de routage multi-sauts selon des classifications fonctionnelles (réactif, proactif, hybride et géographique).

- Dans le troisième chapitre, sont étudiées les catégories de solutions existantes de routage avec qualité de service (*QoS).* Nous donnerons, en ce sens, des exemples de protocoles de routage suivant les différentes catégories.

- Dans le quatrième chapitre, nous expliciterons notre proposition pour l'amélioration de la stabilité des itinéraires dans les réseaux Ad-Hoc. Nous décrirons les extensions faites au protocole *AODV Standard* ainsi que la méthode d'estimation de la stabilité des itinéraires.

- Le nouveau protocole simulé avec l'outil *OPNET Modeler, fait l'objet du dernier chapitre.* Les résultats de performances obtenus ainsi que des surcoûts engendrées par la solution proposée, en comparaison avec les résultats du protocole de routage classique selon bien entendu, les mêmes scénarios et paramètres de simulation y sont discutés.

- Une conclusion générale reprenant les points forts de ce le travail, ponctuée de perspectives futures éventuelles, termine ce mémoire.

Chapitre 1
Introduction aux Réseaux Ad-Hoc

1.1 Historique et évolution des réseaux sans fil

Trois grandes phases caractérisent l'histoire des télécommunications [35]. La première phase s'étend entre *1678* et *1898*, elle concerne les découvertes théoriques et la mise en évidence des ondes radio. La seconde phase commence au début du *20ème* siècle et s'étend jusqu'à la fin de la *2ème* guerre mondiale, elle est caractérisée par le développement et l'évolution des équipements et des techniques mais pour des usages réservées à certaines catégories (militaire, transport,...). La dernière phase est caractérisée par le progrès technique qui va faire entrer les systèmes de communications sans fil et mobiles dans le domaine grand publique, comme le *GSM* [1] *(Global System for Mobile Communication).*

Le développement des réseaux locaux sans fil et leur utilisation a débuté dans des projets à cadre purement militaire mais l'application de ces réseaux s'est étendu à d'autres domaines et s'est révélé d'un grand apport lors des catastrophes naturelles, des incendies où il est indispensable de disposer rapidement d'un réseau pour organiser les opérations de secours. Les travaux de recherche sur les réseaux mobile Ad-Hoc ou *MANET* [2] (*Mobil Ad-Hoc NET Works*) ont débuté au début des années *70* par le *DARPA* (*Defense Avance Research Projects Agency*). Ce dernier a fondé le projet *PRNets* [34] (*Packet Radio Networks*), qui est constitué de plusieurs terminaux sans fil disposant d'une architecture distribuée et partageant le canal de diffusion par le billet des protocoles *ALOAH* et *CSMA* pouvant ainsi communiquer entre eux sur un champ de bataille. Par la suite, en *1983* les *SURAN* (*Survivable Radio Networks*) furent développés par la *DARPA* dont l'objectif est de surpasser les limitations des *PRNets* 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 [34].

Les applications civiles ou généralisés des réseaux Ad-Hoc ne sont apparues que beaucoup plus tard, vers la fin des années *90*. Plusieurs facteurs ont contribué à leurs évolutions rapides. En effet, la miniaturisation des composants a permis de réduire la taille des produits électroniques sans fils en général, comme le *GPS (Global Positioning System)*, téléphone sans fil et le matériel informatique en particulier. Cette miniaturisation s'est accompagnée avec une baisse de consommation et une évolution d'autonomie [35]. Un autre facteur important qui a contribué à la divulgation de la technologie sans fil est la baisse des prix ainsi que la disponibilité des bandes radios *ISM* (*Industrial Scientific and Medical*) et *UNII* (*Unlicensed National Information Infrastructure*) à *800 MHz*, *2.4 GHz* et *5GHz*, qui peuvent être utilisées gratuitement et sans aucune autorisation.

Les différentes catégories des réseaux sans fil existent suivant leur étendue sont :

- Pour les petits réseaux personnels d'une dizaine de mètres de portée *WPAN* (*Wireless Personal Area Network*), les principales normes sont : *IEEE 802.15* ou *Bluetooth* et *ETSI HiperPAN* (*European Telecommunications Standards Institute —High Performance Personal Area Network*).

- Pour les réseaux locaux *WLAN* (*Wireless Local Area Network*), nous avons : *IEEE 802.11* ou *WiFi* et *ETSI HiperLAN* (*European Telecommunications Standards Institute — High Performance Local Area Network*).

- Pour les réseaux atteignant plus de dix kilomètres *WMAN* (*Wireless Metropolitan Area Network*), on trouve : *IEEE 802.16* ou *WIMAX* et *ETSI HiperMAN* et *HiperACCESS*.

- En fin, pour les très grands réseaux c'est-à-dire le réseau universel *WWAN* (*Wireless Wide Area Network*), on a : *IEEE 802.20*, *GSM*, le *GPRS* (*General Packet Radio Services*) ou *l'UMTS* (*Universal Mobile Telecommunication System*).

Nous ne considérons dans cette étude que des solutions pouvant servir de base à la construction de réseaux faciles à mettre en œuvre à partir de *PDA (Personal Digital Assistant)*, micro portable, de bâtiments soient les réseaux locaux sans fil *WLAN*. La norme *802.11* dans les *WLAN* (*Wireless Local Area Network*) offre deux modes de fonctionnement, le mode infrastructure et le mode Ad-Hoc. Le mode infrastructure (voir figure 1.1, dans la page suivante) est défini pour fournir aux différentes stations des services spécifiques, sur une zone de couverture déterminée par la taille du réseau. Les réseaux d'infrastructure sont établis en utilisant des points d'accès qui jouent le rôle de station de base pour l'ensemble de stations. Les réseaux en mode Ad-Hoc auxquels nous nous sommes intéressés sont ceux décrits et étudiés par le groupe *MANET* (*Mobile Ad-Hoc Networks*) de *l'IETF* (*Internet Engineering Task Force*). Une définition de ces réseaux est donnée formellement dans *RFC2501* [3].

1.2 Définition des réseaux mobiles Ad- Hoc

Un réseau mobile Ad-Hoc est une collection de périphériques équipés d'une technologie de transmission sans fil et dotés de protocoles permettant la mise en réseaux de ceux-ci. La particularité de ce type de réseau est que chaque nœud peut communiquer avec n'importe quel autre nœud du réseau.

En effet dans la figure 1.1, si le nœud *A* veut communiquer avec le nœud *B* qui n'est pas à porter radio, alors il passera par une série de nœuds intermédiaires (*C* est un nœud intermédiaire) qui joueront le rôle de relais entre la source et la destination.

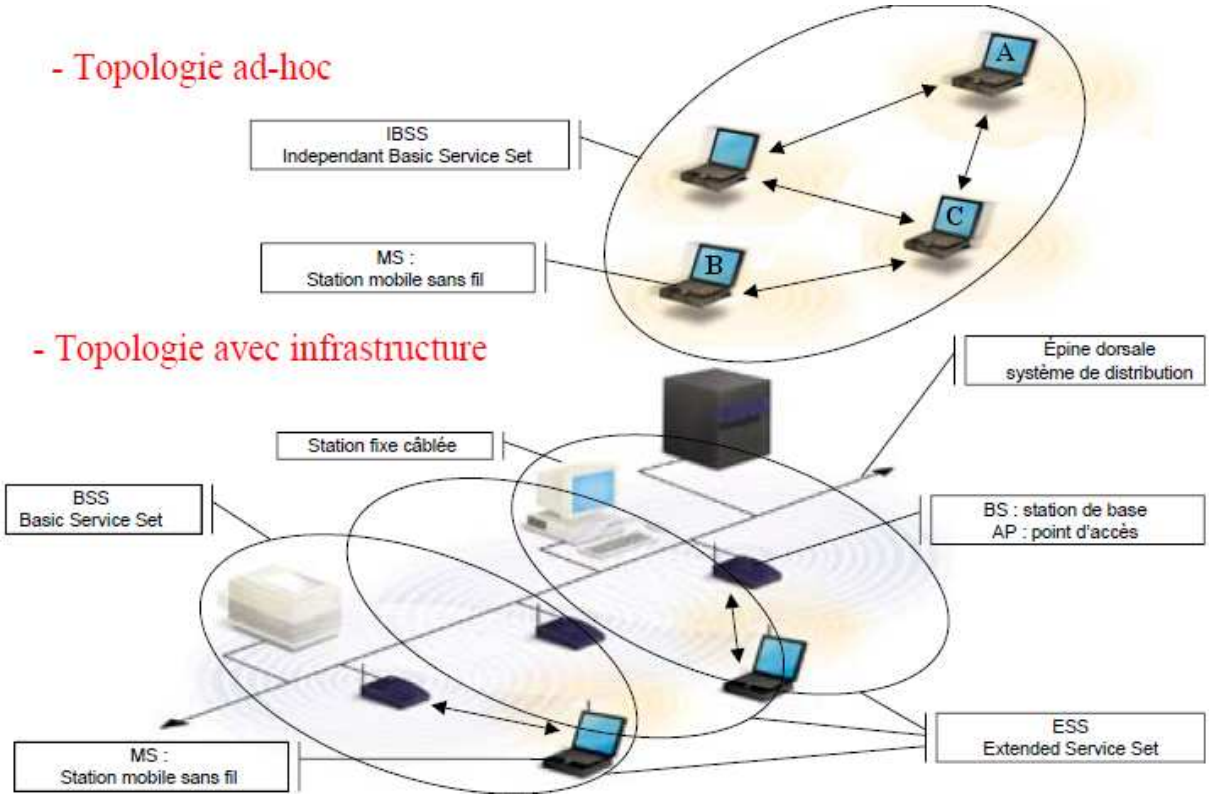


Figure 1.1 - Mode Ad-Hoc et infrastructure de la norme *IEEE 802.11*

Un réseau Ad-Hoc est adaptatif et s'auto-organise, c'est à dire il se forme et se déforme à la volée sans intervention d'une entité administrative où serait centralisé la gestion, comme c'est le cas pour un point d'accès en mode infrastructure. Par conséquence, les nœuds Ad-Hoc doivent être capables de détecter la présence des éventuels voisins et d'effectuer les négociations nécessaires pour mettre en place une communication et un partage d'informations et de services.

La démocratisation des prix des technologies de transmission sans fil et l'émergence de protocoles standards (tels que *802.11*, *Bluetooth*...) ont contribué à l'expansion des réseaux Ad-Hoc. Initialement prévu pour la mise en réseau d'ordinateur, les technologies *WiFi* inondent aujourd'hui le marché *High-tech* et se retrouvent communément dans les téléphones cellulaires, ordinateurs de poche (*PDA)*, et les consoles de jeux portables. Parmi les protocoles de routage les plus célèbres que nous allons décrire dans la section 2.4, nous avons: le protocole *DSR*, le protocole *AODV*, le protocole *OLSR*, le protocole *ZRP*, le protocole *CBRP*, le protocole *LAR*, le protocole *DREAM*…etc.

1.3 Domaines d'applications des réseaux Ad-Hoc

Le premier domaine d'application des réseaux Ad-Hoc fut le domaine militaire où les différents groupes d'unités communiquaient ensemble par liaison radio. L'utilisation des réseaux Ad-Hoc par les militaires s'explique par le caractère particulier de cette technologie qui est adapté aux situations hostiles.

Puisque ce réseau ne nécessite pas d'infrastructure, il est possible de l'utiliser dans le cadre des sinistres comme les tremblements de terre ou les incendies, ainsi les équipes de sauvetage peuvent communiquer bien que les infrastructures de communication classiques soient détruites. De plus, la fiabilité de tel réseau a été prouvée puisque les communications peuvent emprunter plusieurs chemins différents [4].

Les réseaux Ad-Hoc peuvent également être utilisés dans le cadre commercial pour former des réseaux locaux. En effet, la simplicité de la mise en place (pas de câblage et de travaux d'installation dans le bâtiment) se traduit par des économies intéressantes pour l'entreprise. Dans le cas de regroupement pour les jeux sous réseau ou de réunions comme des conférences, l'organisation est simplifiée par la non nécessité de câblage ou de travaux d'aménagements.

Dans le cadre de l'informatique omniprésente, les réseaux Ad-Hoc peuvent relier entre eux les différents composants informatiques et tous les équipements de la maison constituant ainsi non plus un réseau local *WLAN* (*Wireless Local Area Network*) mais plutôt un réseau personnel *WPAN* (*Wireless Personal Area Network*) et le routage est utilisé pour faire communiquer tous les éléments en utilisant de faibles puissance et donc de faible portée ainsi on réduit la consommation énergétique et on diminue les risques pour la santé des utilisateurs.

En fin, les réseaux Ad-Hoc deviennent très utiles pour effectuer des mesures en milieu à très grand risque pour la vie humaine (volcan, fond mairain, espace …). Pour ces cas d'utilisations, ils sont appelés les réseaux de senseurs [5] et ils sont destinés à mesurer les propriétés physiques des environnements (la température, la pression, …), ils sont dispersés à grande échelle, ils effectuent les mesures et les transmettent par l'intermédiaire d'un routage Ad-Hoc le long du réseau ainsi formé. La caractéristique principale d'une telle application est la forte contrainte d'énergie, de mémoire et la capacité de traitement très réduite de ces dispositifs.

1.4 Propriétés et spécificités des réseaux Ad-Hoc

Un réseau Ad-Hoc mobile est considéré comme un système autonome dynamique composé de nœuds mobiles interconnectés par des liens sans fil, sans l'utilisation d'une infrastructure fixe et sans administration centralisée. Les nœuds sont libres de se déplacer aléatoirement et s'organisent arbitrairement. Par conséquent, la topologie du réseau peut varier de façon rapide et imprévisible.

Un réseau Ad-Hoc peut être autonome ou connecté à une infrastructure fixe. La route entre un nœud source et un nœud destination peut impliquer plusieurs sauts sans fil, d'où l'appellation de « réseaux sans fil multi-sauts ». Un nœud mobile peut communiquer directement avec un autre nœud s'il est dans sa portée de transmission. Au delà de cette portée, les nœuds intermédiaires jouent le rôle de routeurs (relayeurs) pour relayer les messages saut par saut.

Les réseaux Ad-Hoc héritent les mêmes propriétés et problèmes liés aux réseaux sans fil. Ces problèmes spécifiques ayant une influence importante sur les solutions à mettre en place pour assurer la qualité de service. Notamment, le fait que le canal radio soit limité en termes de capacité de la bande passante, plus exposé à la perte des données comparée au médium filaire, et sujet à des variations dans le temps.

Le canal est confronté aux problèmes de « station cachée » c'est-à-dire que les nœuds qui ne s'entendent pas, à cause d'un obstacle qui empêche la propagation des ondes, peuvent provoquer des collisions [8] et aux problèmes de « station exposée » c'est-à-dire que ce problème est l'inverse du précédent, lors d'une communication radio entre deux mobiles, leurs voisins respectifs ne peuvent émettre car ils sont bloqués par le protocole d'accès au medium (*MAC*) ainsi beaucoup de mobiles se retrouvent alors immobilisés même si leurs transmissions ne brouilleraient pas le canal de transmission. Le débit utile des mobiles est donc diminué considérablement. En outre, les liens sans fil sont asymétriques et pas sécurisés.

D'autres caractéristiques spécifiques aux réseaux Ad-Hoc conduisent à ajouter une complexité et des contraintes supplémentaires qui doivent être prises en compte lors de la conception des algorithmes et des protocoles réseaux, à savoir :

- **L'absence d'une infrastructure centralisée** : chaque nœud travaille dans un environnement pair à pair distribué, et agit en tant que routeur pour relayer des communications, ou génère ses propres données. La gestion du réseau est ainsi distribuée sur l'ensemble des éléments du réseau.

- **La mobilité des nœuds et maintenance des routes** : la mobilité continue des nœuds crée un changement dynamique de topologie. Par exemple, un nœud peut joindre un réseau, changer de position ou quitter le réseau. Ce déplacement a naturellement un impact sur la morphologie du réseau et peut modifier le comportement du canal de communication. Ajoutons à cela la nature des communications (longues et synchrones, courtes et asynchrones). Les algorithmes de routage doivent ainsi résoudre ces problèmes et supporter la maintenance et prendre en charge en un temps limité la reconstruction des routes tout en minimisant *l'overhead* généré par les messages de contrôle.

- **L'hétérogénéité des nœuds** : un nœud mobile peut être équipé d'une ou plusieurs interfaces radio ayant des capacités de transmission variées et opérant dans des plages de fréquences différentes. Cette hétérogénéité de capacité peut engendrer des liens asymétriques dans le réseau. De plus, les nœuds peuvent avoir des différences en terme de capacité de traitement, de logiciel, de taille et de mobilité (lent, rapide). Dans ce cas, une adaptation dynamique des protocoles s'avère nécessaire pour supporter de telles situations.

- **La contrainte d'énergie** : Les équipements mobiles disposent de batteries limitées [4], et dans certains cas très limitées tels que les *PDA (Personal Digital Assistant),* et par conséquent d'une durée de traitement réduite. Sachant qu'une partie de l'énergie est déjà consommée par la fonctionnalité du routage. Cela limite les services et les applications supportées par chaque nœud.

- **La taille des réseaux Ad-Hoc** : elle est souvent de petite ou moyenne taille (une centaine de nœuds) ; le réseau est utilisé pour étendre temporairement un réseau filaire, comme pour une conférence ou des situations où le déploiement du réseau fixe n'est pas approprié (par exemple : catastrophes naturelles). Cependant, quelques applications des réseaux Ad-Hoc nécessitent une utilisation allant jusqu'à des dizaines de milliers de nœuds, comme dans les réseaux de senseurs [5]. Des problèmes liés au passage à l'échelle tels que: l'adressage, le routage, la gestion de la localisation des senseurs et la configuration du réseau, la sécurité,...etc. doivent être résolus pour une meilleure gestion du réseau.

1.5 La norme IEEE 802.11

L'*IEEE 802.11* [55] est donc la norme pour les *WLAN*, connue aussi sous la dénomination *WiFi* (*Wireless Fidelity*), qui correspond en réalité à un label délivré par le consortium d'industriels intitulé la «*WiFi Alliance*". Ce label garantit l'interopérabilité des équipements issus de différents fabricants, mais par abus de langage, le terme *WiFi* est également employé pour désigner les normes *IEEE 802.11*.

Par rapport au modèle *OSI*, le *IEEE 802.11* ne concerne qu'une partie de la couche de liaison de données *2* et la couche physique *1* et reste donc entièrement compatible avec les couches supérieures (Figure 1.2). Le standard est vu de manière transparente par les applications et les différents protocoles des couches supérieures (*HTTP*, *FTP*, *TCP/IP*...).

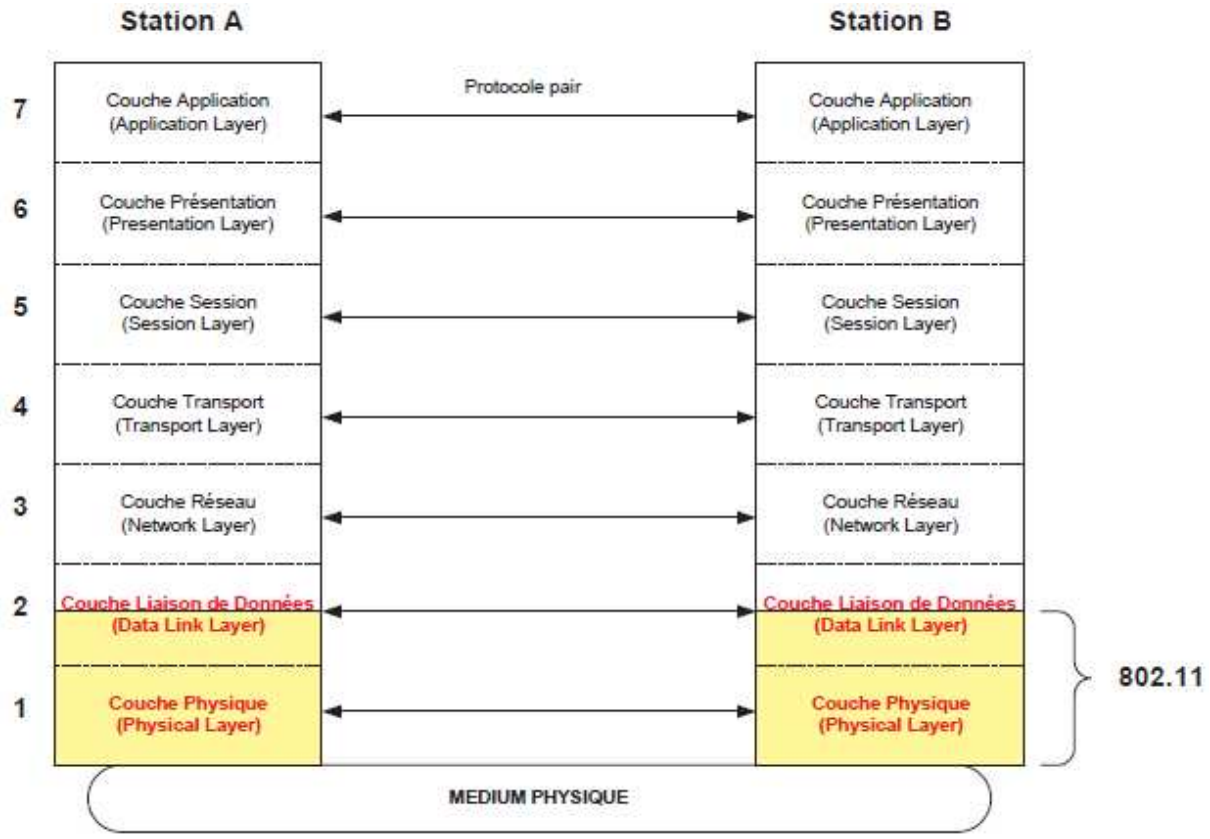


Figure 1.2 - La position du standard *IEEE 802.11* [55] dans le modèle *OSI*, d'après *ISO 94*

Au niveau de la couche de liaison de données, le *IEEE 802.11* supporte deux mécanismes d'accès au médium : le *PCF* (*Polling Coordination Function*) et le *DCF* (*Distributed Coordination Function*).

Le standard *IEEE 802.11* [55] intervient, comme on a dit, au niveau de la couche physique du modèle *OSI* (Figure 1.2, voir page précédente) et il utilise plusieurs bandes de fréquences (*2,4 GHz* ou *5 GHz*), ainsi que différentes techniques de modulation et de codage de l'information. Le standard *IEEE 802.11* se décline donc en différentes normes :

- *IEEE 802.11a* : Cette norme pour les *WLAN* utilise la bande des *5 GHz* avec un débit théorique de 54Mb/sec.

- *IEEE 802.11b* : Ce standard a connu un grand succès en *1999*. Il offre des débits jusqu'à *11 Mb/sec* et une portée radio sept fois supérieur à *802.11a* dans un espace dégagé. Il est utilisé dans la bande de fréquence des *2,4 GHz* ce qui le rend plus sensible aux interférences des appareils électroménagers ayant la même bande (four à micro onde).

- *IEEE 802.11g* : Cette une amélioration de la norme *IEEE 802.11b* ainsi elle offre un débit théorique de *54 Mb/sec* et une compatibilité ascendante avec *IEEE 802.11b*.

Parmi d'autres normes, il faut mentionner *802.11e* qui améliore la couche *MAC* [6] pour incorporer de la *QoS* (pour le support de la voix et de la vidéo sur les réseaux *802.11*) et *802.11n* qui offre des débits de l'ordre de *100 Mb/sec* grâce aux technologies *MIMO* (*Multiple Input Multiple Output*).

L'architecture *802.11* [55] est cellulaire. Un groupe de terminaux munis d'une carte d'interface réseau *802.11*, s'associent pour établir des communications directes et forment un *BSS (Basic Set Service).* Comme illustré à la figure 1.1 dans la page 8, le standard *802.11* offre deux modes de fonctionnement, le mode infrastructure et le mode Ad-Hoc. Le mode infrastructure est défini pour fournir aux différentes stations des services spécifiques sur une zone de couverture déterminée par la taille du réseau. Les réseaux d'infrastructure sont établis en utilisant des points d'accès, ou *AP (Access Point),* qui jouent le rôle de station de base pour une *BSS* (*Basic Set Service*).

Lorsque le réseau est composé de plusieurs *BSS* (*Basic Set Service*), chacun d'eux est relié à un système de distribution, ou *DS (Distribution System),* par l'intermédiaire de leur point d'accès (*AP*) respectif. Un système de distribution correspond en règle générale à un réseau Ethernet utilisant du câble métallique. Un groupe de *BSS* interconnectés par un système de distribution (*DS*) forment un *ESS (Extented Set Service),* qui n'est pas très différent d'un sous système radio de réseau de mobiles.

Le système de distribution (*DS*) est responsable du transfert des paquets entre les différentes stations de base (*AP*). Dans les spécifications du standard, le système de distribution (*DS)* est implémenté de manière indépendante de la structure hertzienne et utilise un réseau Ethernet métallique. Il pourrait tout aussi bien utiliser des connexions hertziennes entre les points d'accès (*AP*).

Le standard *802.11* [55] s'attache à définir les couches basses du modèle *OSI* (*Open System Interconnexion*) pour une liaison sans fil utilisant des ondes électromagnétiques (Tableau 1.1), c'est-à-dire :

- La couche physique (notée parfois couche *PHY*).
- La couche liaison de données : constituée de deux sous-couches ; le contrôle de la liaison logique (*Logical Link Control*, ou *LLC*) et le contrôle d'accès au support (*Media Access Control*, ou *MAC*).

La couche physique définit la modulation des ondes radioélectriques et les caractéristiques de la signalisation pour la transmission de données, tandis que la couche liaison de données définit l'interface entre le bus de la machine et la couche physique, notamment une méthode d'accès proche de celle utilisée dans le standard Ethernet et les règles de communication entre les différentes stations.

Couche 2 *OSI* *Liaison de données*	***802.11* Logical Link Control (*LLC*)**				
	***802.11* Medium Access Control (*MAC*)**				
Couche 1 *OSI* *Physique*	**Type *PHY***	**802.11 PHY (*FHSS/DSSS*)**	**802.11a/g PHY (*OFDM*)**	**802.11b *PHY* (*DSSS*)**	**802.11n *PHY* *(MIMO)***
	Débit	*2Mbps*	*54Mbps*	*11Mbps*	*540Mbps*
	Fréquence	*2.4 GHz*	*2.4 et 5 GHz*	*2,4 GHz*	*2.4 et 5.1GHz*

Tableau 1.1 - Quelques normes de la famille *802.11*

1.5.1 La couche physique de 802.11

La couche physique est chargée de la transmission des données d'un émetteur jusqu'au récepteur. La norme *802.11* [55] définit six techniques de transmission qui utilisent les ondes électromagnétiques et lumineuses comme support physique et qui se différencie par la modulation utilisée.

Le standard *IEEE 802.11* admet donc les spécifications de la couche physique suivantes [7]:

- ✓ *FHSS (Frequency Hopping Spread Spectrum)*
- ✓ *OFDM (Orthogonal Frequency Division Multiplexing)*
- ✓ *DSSS (Direct Sequence Spread Spectrum)*
- ✓ *HR-DSSS (High Rate DSSS)*
- ✓ *MIMO (Multiple Input Multiple Output)*
- ✓ Infrarouge.

La technique *FHSS* (étalement de spectre par saut de fréquence) utilise *79* canaux qui ne se recouvrent pas, chacun avec une largeur de bande de *1 MHz*, à partir de la fréquence *2,4 GHZ*.

La technique *OFDM* utilise le multiplexage orthogonal en répartition de fréquence c'est-à-dire une transmission est répartie simultanément sur plusieurs ondes porteuses à fréquence distinctes (*52* fréquences sont utilisées dont *48* aux données et *4* à la synchronisation) pour offrir un débit de *54 Mb/sec*.

Les techniques *DSSS* et *HR-DSSS* (étalement de spectre par séquence directe) divisent la bande des *2,4 GHz* en *13* ou *14* canaux de *22 MHz*. Les canaux adjacents ont des bandes passantes qui se recouvrent partiellement et sont donc sujette à des perturbations mutuelles (Figure 1.3), ce qui se traduit par un signal très bruité.

La technologie *MIMO* permet l'utilisation de la diversité spatiale grâce à l'installation de plusieurs antennes.

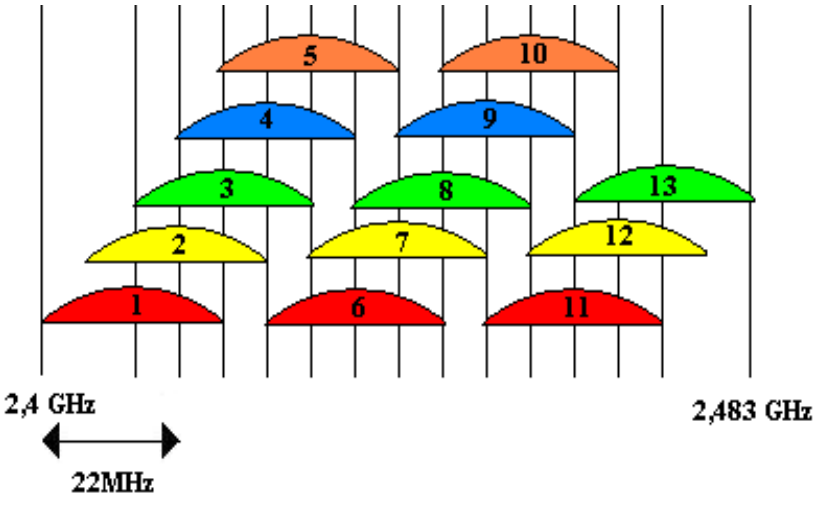


Figure 1.3 - Allocation du spectre par séquence directe *DSSS*

La dernière technique basée sur l'infrarouge utilise les ondes électromagnétiques comprises entre *850* et *950 nanomètres* (longueur d'onde supérieure à celle de la lumière visible). En revanche, les infrarouges peuvent pénétrer à travers le verre, mais pas à travers des obstacles opaques, ce qui représente un avantage en termes de sécurité. Mais, comme les réseaux infrarouges sont sensibles aux interférences lumineuses, la coupure du faisceau lumineux implique l'interruption de la transmission. La portée des transmissions est limitée à *10 mètres* (technique très peu utilisée) [7].

1.5.2 Le protocole d'accès au medium

Au-dessus de la couche physique, la norme définit un unique protocole d'accès au médium [6] (la couche *MAC*), afin de gérer les accès concurrents à un même médium partagé. Ce protocole fait partie de la famille des protocoles de gestion des accès multiples par détection de porteuse avec évitement de collisions (*Carrier Sense Multiple Access with Collision Avoidance- CSMA/CA*). Il associe un mécanisme de détection de porteuse avant transmission à un mécanisme d'attente aléatoire permettant de limiter le nombre et l'impact des collisions. En plus, le standard définit un mécanisme supplémentaire *RTS/CTS* (*Request To Send/Clear To Send*) pour éviter les collisions et le problème des nœuds cachés.

1.5.2.1 La sous-couche *LLC* (*Logical Link Control*)

La sous-couche *LLC* (spécification *IEEE 802.2*), qui est indépendante des mécanismes d'accès au support physique, représente une partie de la couche de liaison de données. Elle présente les caractéristiques de fiabilité grâce au séquencement et à la retransmission des données en cas de détection d'erreurs.

1.5.2.2 Description du protocole *MAC*

La couche *MAC* [6] de la norme *IEEE 802.11* [55] définit deux modes d'accès au médium :

- **Le mode centralisé** (*Point Coordination Function, PCF*): Ce mode est avec infrastructure représenté par la figure 1.4 (a) de la page suivante, ainsi les nœuds mobiles communiquent directement avec un point d'accès fixe. Ces points d'accès sont analogues aux stations de bases des réseaux cellulaires et sont généralement reliés entre eux par un réseau filaire ou hertzien. Un point d'accès est le point de passage obligé pour qu'un mobile puisse communiquer avec un autre mobile. Les mobiles peuvent se déplacer tout en restant dans la zone de couverture du point d'accès.

- **Le mode distribué** (*Distributed Coordination Function, DCF*) : Les nœuds mobiles communiquent directement entre eux sans avoir recours à une tierce station, à condition qu'ils soient à porté radio (figure 1.4 b). C'est ce dernier mode qui est utilisé dans le cas des réseaux ad hoc. Dans la figure 1.4 (c), si un nœud *A* sort de la portée radio d'un autre nœud *B*, il ne pourra pas communiquer avec *B*.

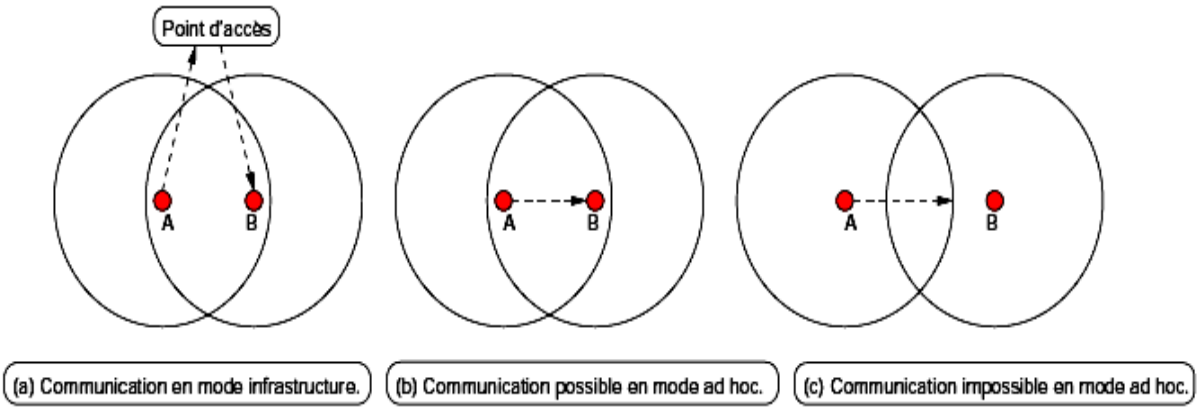


Figure 1.4 - Modes d'opération dans *IEEE 802.11*

1.5.2.3 Principe de base du *DCF*

La couche *MAC* [6] *802.11* d'un réseau Ad-Hoc utilise la méthode d'accès au medium appelée *DCF* (*Distributed Coordination Function*). *DCF* est basé sur la technique *CSMA/CA* (*Carrier Sense Multiple Access/Collision Avoidance)* et offre d'autres fonctions qui sont confiées normalement aux couches supérieures, comme la fragmentation, les retransmissions et les accusés de réception.

Dans un environnement sans fil, les collisions ne sont pas détectées du fait qu'un émetteur ne peut pas transmettre et écouter en même temps. Pour éviter les collisions, le *CSMA/CA* utilise différents mécanismes, tels que l'écoute du support physique, l'algorithme de *Backoff* (section 1.5.2.5) pour gérer l'accès au canal, la réservation du canal et les trames d'acquittement positif.

Dans *l'IEEE 802.11*, l'écoute du support se fait à la fois au niveau de la couche physique avec le *PCS* (*Physical Carrier Sens*) et au niveau de la couche *MAC* avec le *VCS* (*Virtual Carrier Sense*). Le *PCS* détecte la présence d'autres stations *802.11* en analysant toutes les trames passant sur le support hertzien et en détectant l'activité sur le support grâce à la puissance relative du signal des autres stations. Le *VCS* est un mécanisme de réservation basé sur l'envoi de trames *RTS/CTS* (*Request To Send/Clear To Send*) entre une station source et une station destination avant tout envoi de données.

La transmission d'un paquet par une station *802.11* utilise la méthode appelée *Two-Way Handshake Scheme*. Dans cette technique, chaque nœud (souhaitant émettre des données) doit écouter le canal avant de tenter d'obtenir l'accès. Si le canal est occupé le nœud doit attendre la fin de la transmission en cours pour avoir le droit d'accès au médium. Lorsque le canal devient libre, il faut qu'il le reste pour une période *DIFS* (*DCF Inter-Frame Space*), si le canal reste libre durant le *DIFS* alors les nœuds qui veulent émettre choisissent un temps de temporisation appelé *Backoff*. Lorsque la temporisation expire, si le canal est inoccupé, ils peuvent commencer l'envoi de ses paquets. Dans le cas de plusieurs nœuds qui veulent accéder au canal (Figure 1.5**)**, celui qui a choisi la temporisation la plus courte est donc celui qui gagne le droit d'accès les autres stations mettent à jour leur *NAV* (*Network Allocation Vector*), en incluant le temps de transmission de la trame de données, le *SIFS* ainsi que la durée de l'acquittement *ACK* (*acknowledgement*). Si les données envoyées ont été reçues de manière intacte (la station destination vérifie le *CRC* (*Cyclic Redundancy Check*) de la trame de données), la station destination attend pendant un temps équivalent à un *SIFS* (*Short InterFrame Space*) et émet un *ACK* (*acknowledgement*) pour confirmer la bonne réception des données. Si par contre, cet *ACK* (*acknowledgement*) n'est pas détecté par la station source ou si les données n'ont pas été reçues correctement ou encore si cet *ACK* n'a pas été reçu correctement, alors on suppose qu'une collision s'est produite et les données sont retransmises après un certain temps aléatoire. En plus, pour éviter la monopolisation du canal, La station *802.11* doit attendre un temps *Backoff* [8] aléatoire entre la transmission de deux nouveaux paquets (paquets ne font pas partis de la même session). Le mécanisme de *Backoff* limite les risques de collision mais ne les supprime pas complètement, il faut utiliser dans ce cas le mécanisme de prévision de collision. La figure 1.5 montre le mécanisme de base de *two-way handshake*.

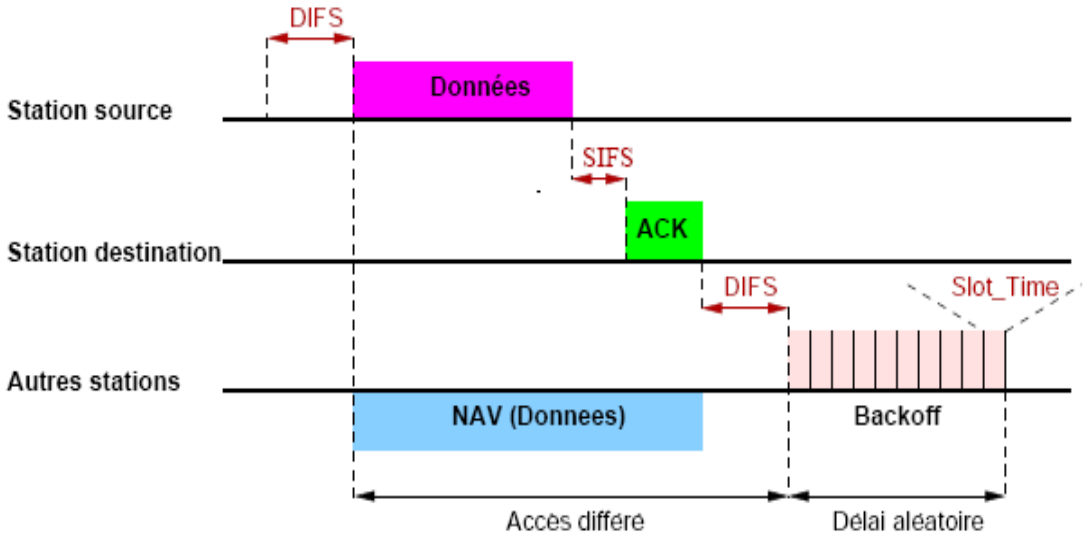


Figure 1.5 - La méthode de base de transmission des données (*Two-Way Handshake Scheme*)

1.5.2.4 Prévention de collision

La méthode de base de transmission d'un paquet par une station *802.11* (*Two-Way Handshake Scheme*) n'utilise pas le mécanisme de réservation. Le *VCS* (*Virtual Carrier Sense*) permet de réserver le support entre deux stations avant tout envoi de données (*Four-Way Handshake Scheme*). Lorsqu'une station source veut transmettre des données, elle envoie tout d'abord une trame de réservation *RTS* (*Request To Send*). Toutes les stations entendant cette trame, lisent le champ de durée du *RTS* et mettent à jour leurs *NAV* (*Network Allocation Vector*).

Dans la figure 1.6, lorsque la station destination reçoit le *RTS*, elle répond, après avoir attendu pendant un *SIFS*, en envoyant une trame *CTS* (*Clear To Send*). Les autres stations entendant le *CTS*, lisent le champ de durée du *CTS* et mettent à nouveau à jour leur *NAV*. Après la réception du *CTS* par la station source, celle-ci est assurée que le support est réservé pour sa transmission des données qui peut alors débuter.

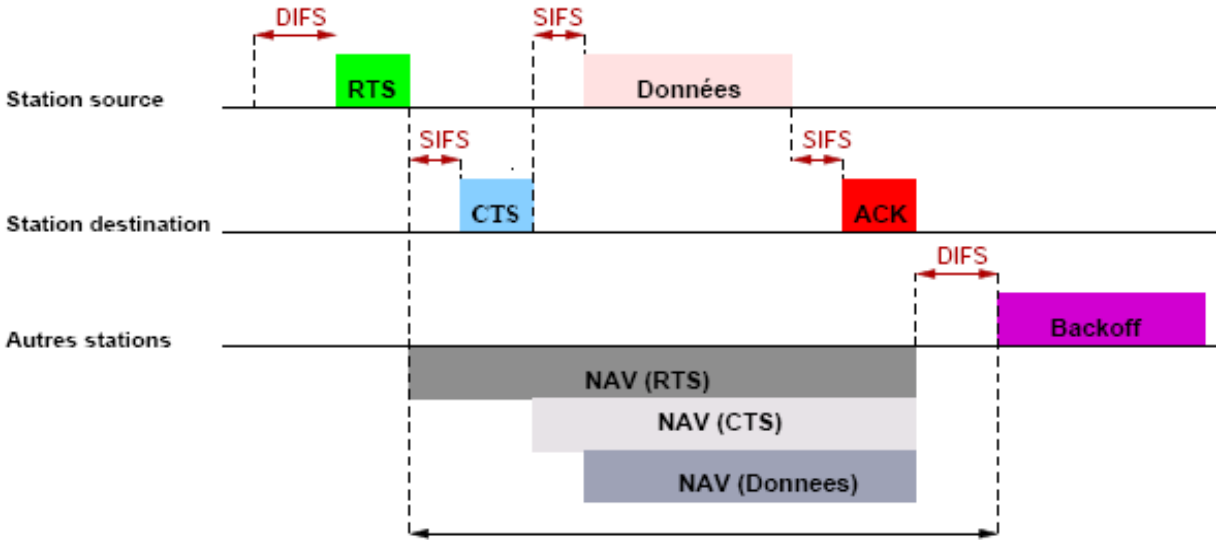


Figure 1.6 – Transmission de données en utilisant les trames *RTS*/*CTS*.
(*four way handshake scheme*)

Ce mécanisme permet ainsi à la station source de transmettre ces données ainsi que de recevoir le *ACK* sans qu'il n'y ait aucune collision. Comme les trames *RTS*/*CTS* réservent le support pour la transmission d'une station, elles sont utilisées habituellement lorsque l'on a de grosses trames à envoyer pour lesquelles une retransmission serait trop coûteuse suivant la métrique bande passante. Les stations peuvent choisir d'utiliser le mécanisme *RTS*/*CTS* que lorsque la trame à envoyer excède une variable *RTS_Threshold*.

1.5.2.5 Algorithme de *Backoff*

Dans *802.11* [55], le temps est découpé en *Slot_Time* qui correspond à la période nécessaire pour détecter la transmission d'un paquet par une autre station (dépend de la couche physique et du délai de propagation). Les stations utilisent l'algorithme de *Backoff* [8] pour savoir quand est ce qu'elles vont pouvoir à nouveau accéder au support. L'algorithme de *Backoff* définit une fenêtre de contention *CW* (*Contention Window*) qui correspond au nombre de time-slots qui peuvent être sélectionnés pour le calcul du temps d'attente (appelé temporisateur de *Backoff*).

Lorsque le support est libre pendant une période supérieure à un *DIFS* (*DCF Inter-Frame Space*), les stations décrémentent leurs *Timer*s jusqu'à ce que le support soit occupé ou jusqu'à ce que le *Timer* atteigne la valeur *0* (pour accéder au support). Pour chaque transmission d'un paquet, le *Backoff* est sélectionnée uniformément entre [*0, CW-1*]. La valeur de *CW* dépend du nombre d'échecs de transmission d'un paquet, c'est-à-dire, pour chaque paquet stocké pour une transmission, la fenêtre *CW* prend une valeur initial *CWmin* et elle sera doublé à chaque échec de transmission, jusqu'à une valeur maximal *CWmax* (les valeurs de *CWmin* et *CWmax* sont spécifiées par la couche physique). La valeur de *CW* conserve *CWmax* pour le reste des tentatives.

Cet algorithme est appelé le schéma exponentiel du *Backoff*. Grâce à cet algorithme, les stations ont la même probabilité d'accéder au support car chaque station doit y accéder à nouveau après chaque transmission. Le mécanisme de *Backoff* est illustré par la figure 1.7.

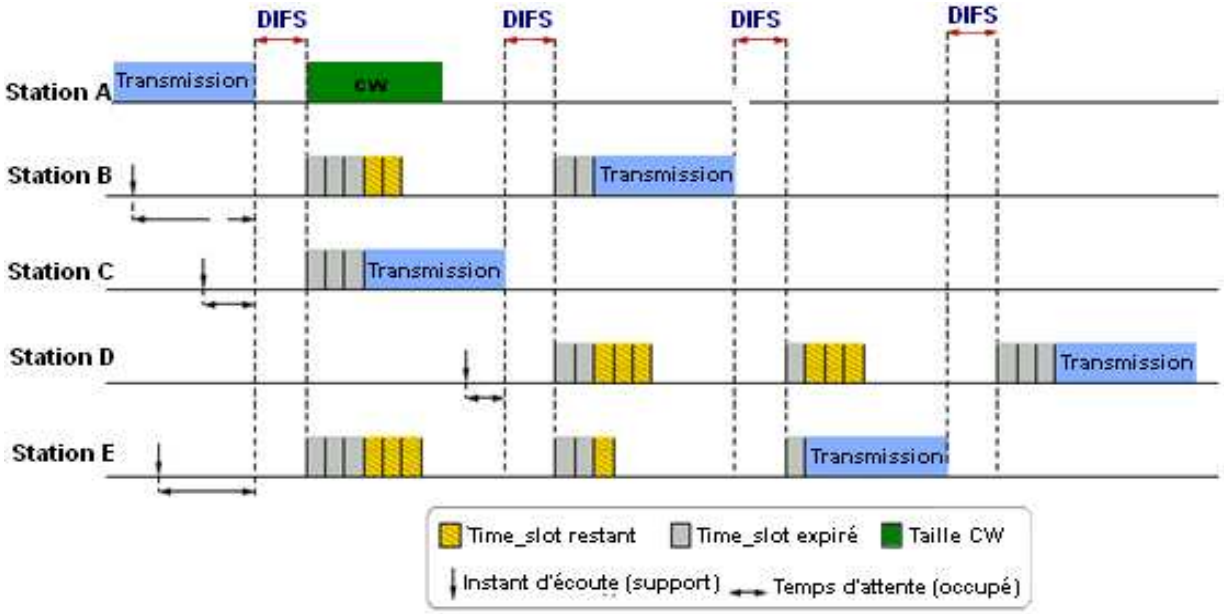


Figure 1.7 – L'algorithme de *Backoff*

1.6 Conclusion

Dans ce chapitre, nous avons présenté l'historique et l'évolution des réseaux sans fil, puis nous avons vu la définition d'un réseau mobile Ad-Hoc et ses domaines d'applications. Ensuite, nous avons exposé les propriétés et spécificités des réseaux Ad-Hoc ainsi que les problèmes liés aux caractéristiques du medium de transmission sans fil.

Nous avons exposé également, les principaux mécanismes utilisés par la norme *IEEE 802.11* [55] pour nous permettre, par la suite, l'évaluation des ressources dans un contexte Ad-Hoc. Dans le chapitre suivant, nous allons présenter les différentes classifications ainsi qu'une sélection des protocoles de routage pour les réseaux mobiles Ad-Hoc.

Chapitre 2

Etude des Protocoles de Routage dans les Réseaux Ad-Hoc

Le routage est un mécanisme clé des réseaux Ad-Hoc. Le routage (*routing protocol*) est le mécanisme d'ouverture et d'entretien d'une communication entre deux nœuds. Il est donc très important d'avoir un protocole de routage efficace si on veut pouvoir tirer parti du potentiel des réseaux Ad-Hoc [11]. L'opération est alors supportée par la source, le destinataire et les relais supportant l'échange.

Dans un premier temps, la source doit trouver le chemin jusqu'au destinataire. Elle peut s'appuyer sur une connaissance préalable du chemin ou demander à d'autres entités un chemin partiel ou complet. Si la source utilise une information incomplète, une chaîne de relais peut se créer jusqu'à joindre le destinataire. Ce dernier s'appuie alors sur les informations reçues pour retrouver le chemin vers la source et ainsi construire le chemin.

Trouver un chemin n'est qu'une partie du problème, il faut pouvoir assurer la stabilité des communications car la mobilité des nœuds peut entraîner de nombreuses reconfigurations des chemins. Ainsi, durant la communication, l'ensemble des relais d'une communication va changer plus ou moins fréquemment.

De plus, par la nature même des réseaux Ad-Hoc, les déconnexions peuvent être un événement normal (par opposition à un événement anormal, dû à une erreur dans un réseau filaire). En effet, un nœud peut se retrouver sans possibilité de joindre le destinataire, simplement par le fait qu'il ne possède pas de voisins ou que le graphe du réseau joignable n'inclut pas le destinataire. Cette erreur oblige la source et/ou certains relais à temporiser des envois et/ou à informer les applications de la source de cet événement.

Dans ce chapitre, nous allons présenter les classes les plus connues de protocoles de routage : proactifs, réactifs, hybrides et géographique. Nous allons exposer ensuite, un ensemble d'exemples de protocoles pour chaque classification mentionnée précédemment.

2.1 Les stratégies de routage

La mise en place de protocoles de routage pour relier les nœuds entre eux est un problème très difficile. L'environnement est dynamique et évolue donc au cours du temps, la topologie du réseau peut changer fréquemment. Le réseau doit être robuste aux changements de topologie (pannes d´´equipements, extensions temporaires du réseau) et à l´´evolution de la demande (mobilité des utilisateurs ou variabilité des applications).

Les stratégies optimales de routage [10] doivent pouvoir faire face à ces types de dynamiques afin que les protocoles de routage résultants ne dégradent pas les performances du système:

- **La minimisation de la charge du réseau** : les stratégies de routage déterminent les goulots d'´etranglement dans le réseau et donc les limites de la capacité offerte aux utilisateurs.

- **Offrir un support pour pouvoir effectuer des communications multipoints fiables :** le fait que les chemins utilisés pour router les paquets de données puissent évoluer, ne doit pas avoir d'incident sur le bon acheminement des données. L'élimination d'un lien, pour cause de panne ou pour cause de mobilité devrait, idéalement, augmenter le moins possible les temps de latence.

- **Assurer un routage optimal :** l'architecture Ad-Hoc, lorsqu'elle utilise des équipements nombreux, contourne ce problème c'est à dire l'accès s'établit de proche en proche, via les autres membres du réseau qui font office de relais. Si la construction des chemins optimaux est un problème dur, la maintenance de tels chemins peut devenir encore plus complexe, on doit assurer une maintenance de routes sans coût supplémentaire.

- **Le temps de latence :** à chaque passage de relais, les informations accusent un léger décalage temporel, augmentant le temps de latence global, a fortiori si le réseau comporte de nombreux éléments.

2.2 Modes de communication dans les réseaux Ad-Hoc

Avant de regrouper les protocoles de routage dans différentes classes, nous allons donner quels sont les principaux modes de communication dans ces réseaux et particulièrement dans les réseaux Ad-Hoc :

- la communication point à point ou *unicast*, pour laquelle il y a une source et une seule destination ;

- la communication multipoint ou *multicast*, qui permet d'envoyer un message à plusieurs destinataires ;

- la diffusion ou *broadcast*, qui envoie un message à tous les nœuds du réseau.

Ces trois modes de communication sont schématisés par la figure 2.1 suivante :

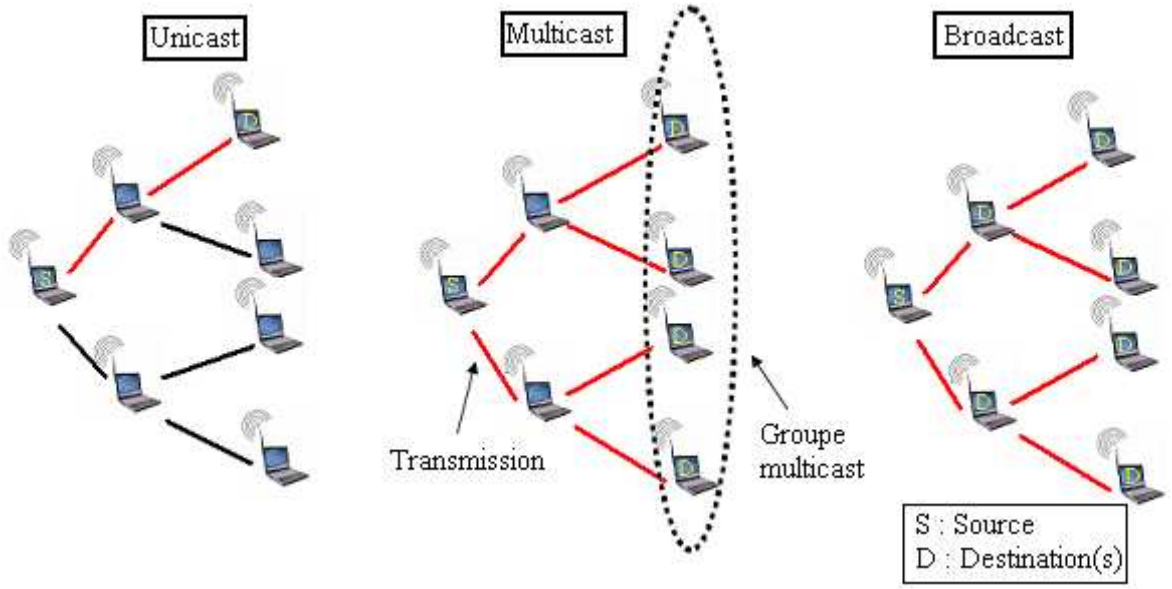


Figure 2.1 – Modes de communication dans les réseaux mobiles

2.3 Familles des protocoles de routage

De nombreux protocoles de routage ont été développés pour les *MANETs* faisant face aux contraintes spécifiques de ce type de réseaux. La fonction principale de ces protocoles de routage est de fournir le chemin le plus court, en terme nombre de sauts, entre une source et une destination de telle façon que le nombre de messages générés soit minimum. Ces protocoles de routage se décomposent en quatre familles [45] suivant qu'ils soient proactifs, réactifs, hybrides ou géographiques.

Les protocoles proactifs, présentés dans la section 2.3.1, maintiennent à jour les tables de routage en recherchant régulièrement des routes. A l'inverse, les protocoles réactifs, appelés aussi protocoles à la demande (*on-demand*), traités dans la section 2.3.2, sont caractérisés par le fait qu'une route n'est recherchée que lorsqu'un message doit être envoyé vers une destination. Les protocoles hybrides, abordés dans la section 2.3.3, utilisent à la fois une approche réactive et proactive. Enfin, les protocoles géographiques, présentés dans la section 2.3.4, enrichissent les protocoles précédents en incluant des informations géographiques dans les tables de routage qu'ils maintiennent [45].

La figure 2.2 ci-dessous illustre les différentes classifications de protocoles de routage dans les réseaux Ad-Hoc mobiles.

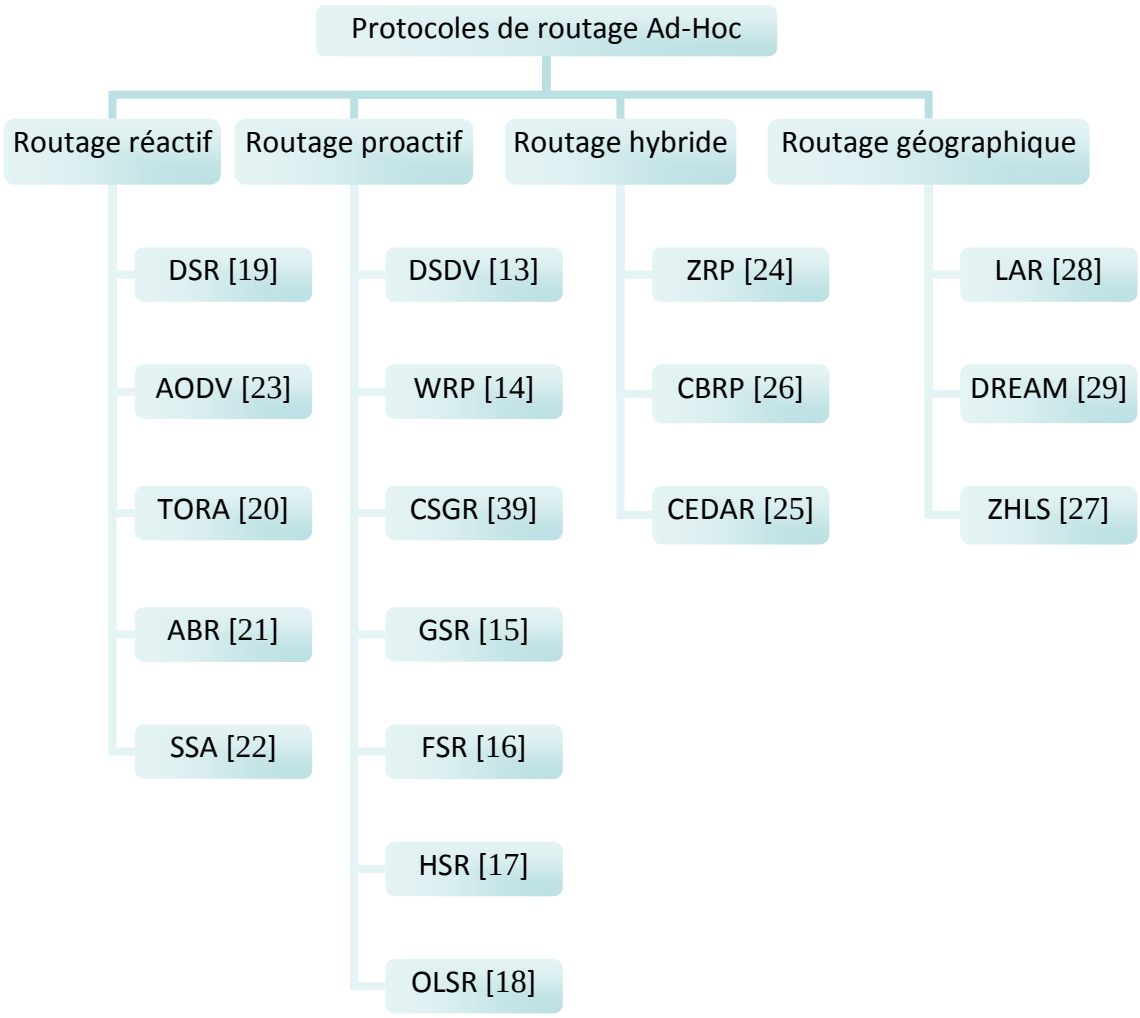


Figure 2.2 - Classification des protocoles de routage

2.3.1 Routage proactif

Le routage proactif, appelé aussi « *table-driven* », essai de maintenir les meilleurs chemins existants vers toutes les destinations possibles au niveau de chaque nœud du réseau. Les protocoles de routage proactifs ont l'avantage de la disponibilité immédiate des routes vers tous les nœuds du réseau. Ainsi, le délai d'acheminement des paquets est très court. Les routes sont sauvegardées mêmes si elles ne sont pas utilisées. La sauvegarde permanente des chemins de routage est assurée par un échange continu des messages de mise à jour des chemins, ce qui induit un contrôle excessif surtout dans le cas des réseaux de grande taille [3]. Comme dans les réseaux filaires, deux principales méthodes sont utilisés : le routage par vecteur de distance et le routage par état de lien.

Les premiers protocoles proposés pour les réseaux Ad-Hoc furent des algorithmes proactifs basés sur une technique de type vecteur de distance, utilisant un algorithme distribué de *BELLMAN-FORD* [30, 31] où des modifications furent apportées pour traiter le problème de convergence de l'algorithme. Le routage par vecteur de distance permet à chaque nœud de diffuser à ses voisins la distance qui les sépare en nombre de sauts. Les seuls informations conservées sont la liste des nœuds du réseau et l'identité du prochain mobile par le quel passer pour atteindre la destination. Le mobile émetteur choisit la route la plus courte en nombre de sauts vers le destinataire.

Pour résoudre le problème de convergence des algorithmes de type vecteur de distance, l'alternative est de mettre en œuvre un algorithme à état de lien (*Link State*) [12]. Le routage par état de lien consiste à diffuser périodiquement aux mobiles voisins l'état de la liaison qui les sépare. Par conséquence, chaque mobile est capable de dresser une carte de l'état du réseau et de cette façon il peut choisir la route la plus appropriée pour un flux de données.

L'avantage de ce type de protocole est sa capacité de trouver facilement des routes alternatives en cas de rupture de lien. Malheureusement ces protocoles atteignent rapidement leurs limites avec l'accroissement du nombre de nœuds et de leur mobilité. Les changements de topologies sont fréquents. Le réseau sera ainsi constamment inondé par les paquets de contrôle qui ne se propagent pas assez vite pour que chaque nœud soit informé à temps des changements. Il en résulte des incohérences dans les tables, un grand espace de stockage et une bande passante réduite par la surcharge des paquets de mise à jour.

Cette famille de protocole est ainsi limitée à des réseaux de petites taille, avec une faible mobilité et où chaque nœud à besoin d'être en permanence connecté avec les autres membres du réseau. Comme exemple de protocoles de routage proactifs, nous avons *DSDV*[13], *WRP* [14], *CSGR* [39] , *GSR* [15], *FSR* [16], *HSR* [17], *OLSR* [18] … etc.

2.3.2 Routage réactif

À l'opposé de l'approche proactive, l'approche réactive (ou *source-initiated on-demand*) découvre le chemin quand nécessaire. Ainsi, cette approche génèrent a priori moins de messages de contrôle que les protocoles proactifs, puisqu'ils ne maintiennent pas à jour leurs tables de routage. Les informations contenues dans les tables de routage, n'étant pas rafraîchies périodiquement, ne sont utilisables que pendant un temps limité déterminé par la durée de vie d'une route, ou dans le pire des cas, une seule fois.

Lorsqu'un nœud cherche à joindre un correspondant dans le réseau, il utilise un protocole de diffusion (message de recherche). Une fois qu'il trouve le correspondant, ce dernier peut répondre par un message de réponse (point à point) qui informe la source de sa présence et du chemin à suivre pour le joindre (figure 2.3). Cette opération peut être renouvelée à un niveau local pour la maintenance des routes.

Ce type de protocoles présente l'avantage de ne pas surcharger le réseau en trafic de contrôle et de ne requérir des routeurs qu'une capacité de stockage minimale et d'économiser leur consommation d'énergie. Toutefois, le délai d'établissement des communications peut augmenter lorsque le réseau est chargé ou lorsque le récepteur est à une distance importante de l'émetteur.

Cette famille de protocoles convient donc mieux dans le cas de réseaux Ad-Hoc avec une taille importante et/ou à forte mobilité, où les communications entre nœuds du réseau sont plus ponctuelles et ne nécessitant pas une connexion permanente avec tous les nœuds du réseau. Ce type de protocole correspond parfaitement avec les caractéristiques requises pour l'obtention d'une qualité de service acceptable dans un réseau Ad-Hoc. Nous pouvons citer ; comme exemple, les protocoles *DSR* [19], *TORA* [20], *ABR* [21], *SSA* [22] et *AODV* [23].

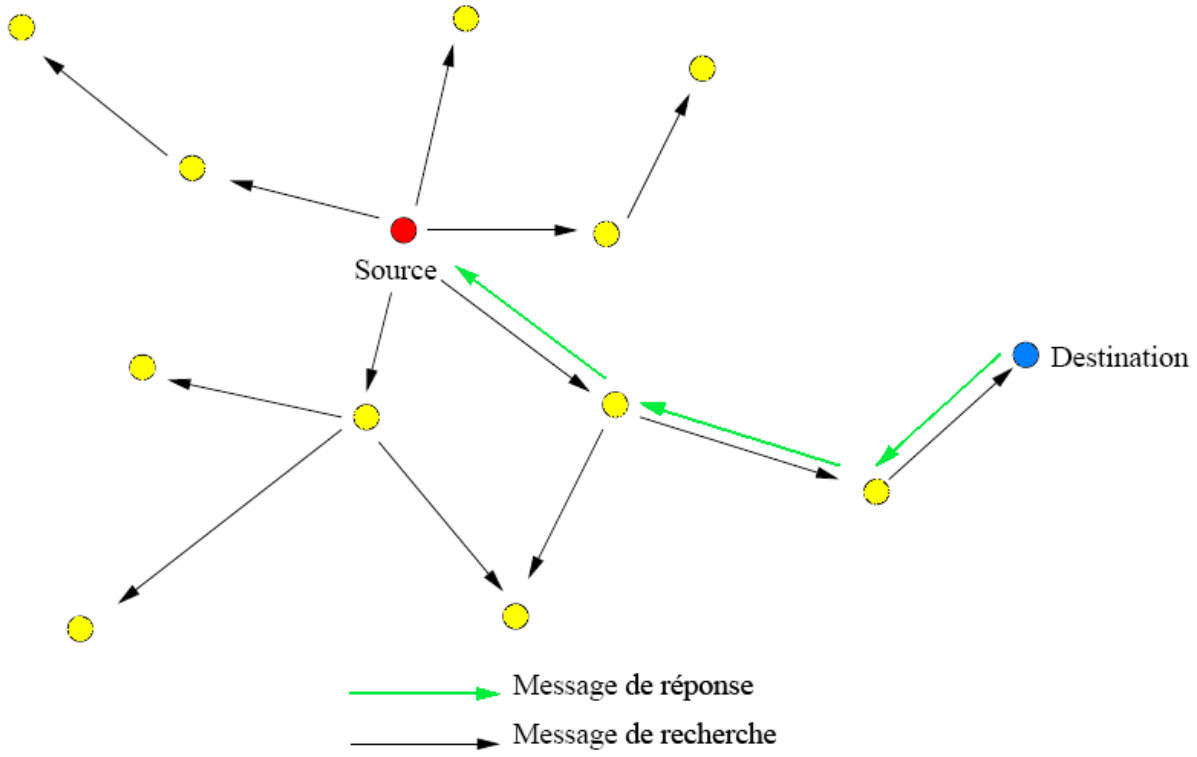


Figure 2.3 – Recherche de route par un protocole réactif

2.3.3 Routage hybride

Le routage hybride englobe les avantages des méthodes actives et proactives. Il utilise un protocole proactif, pour apprendre le proche voisinage par exemple voisinage à deux sauts ou à trois sauts ; ainsi il dispose des routes immédiatement dans le voisinage.

Le nombre de nœuds d'une zone est limité pour réduire *l'overhead* (charge du flux de contrôle) engendré par la maintenance des tables de routage et pour accélérer le processus de routage intra-zone. Au-delà de cette zone prédéfinie, le protocole hybride fait appel aux techniques des protocoles réactifs pour chercher des routes.

À la réception d'une requête de recherche réactive, un nœud peut indiquer immédiatement si la destination est dans le voisinage ou non, et par conséquent savoir s'il faut aiguiller la requête vers les autres zones sans déranger le reste de sa zone.

Ce type de protocole s'adapte bien aux grands réseaux, cependant, il cumule aussi les inconvénients des protocoles réactifs et proactifs : messages de contrôle périodiques plus le coût d'ouverture d'une nouvelle route. *Zone Routing Protocol ZRP* [24], *Core Extraction Distributed Ad-Hoc Routing CEDAR* [25] et *Cluster Based Routing Protocol CBRP* [26] sont des exemples de ce type de protocoles.

2.3.4 Routage géographique

Les protocoles de routage géographiques se basent sur des coordonnées géographiques afin de d'envoyer les messages vers la destination, en routant les messages de façon efficace dans la direction de la destination. Le *GPS (Global Positioning System)* est actuellement le système de localisation le plus utilisé. Pour atteindre cet objectif, les coordonnées géographiques des nœuds sont incluses dans les tables de routage. Concrètement, un nœud inclut l'adresse IP et la position de la destination (fournie par le protocole de routage) dans le message de données à expédier, et envoie ce message dans la direction de la destination. Les nœuds intermédiaires répètent le même mécanisme jusqu'à ce que le message atteigne la destination. Nous avons comme exemples de protocoles de routage géographiques le protocole « *Location-Aided Routing* » *LAR* [28], Le protocole « *Distance Routing Effect Algorithm for Mobility* » *DREAM* [29] et Le protocole « *Zone-Based Hierchical Link State Routing* » *ZHLS* [27].

2.4 Description de quelques protocoles de routage

Le routage dans les réseaux Ad-Hoc constitue, avant tout, 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. On va présenter dans cette section des exemples de protocoles existants pour chaque classification cités précédemment.

2.4.1 Les protocoles de routage proactif

2.4.1.1 Le protocole de routage DSDV

DSDV [13] (*Destination Sequenced Distance Vector*) est l'un des premiers protocoles mis au point par le groupe *MANET*. Il s'agit d'un protocole de routage proactif orienté destination (plus connu sous le nom de *distance vector protocol*) basé sur l'algorithme distribué de *Bellman-Ford* [30, 31]. Ce type de protocole suppose que tous les nœuds du réseau disséminent une copie de leur vecteur de distance (*distance vector*), c'est-à-dire, de leur table de routage. Le vecteur de distance associé à un nœud du réseau indique les autres nœuds du réseau qui lui sont accessibles et le nombre de sauts nécessaires pour atteindre ces nœuds.

Chaque nœud maintient dans sa table de routage un ensemble d'informations pour chaque destinataire, contenant :

- l'adresse du destinataire,
- le nombre de sauts pour l'atteindre, et
- le numéro de séquence associé au nœud destinataire.

La principale amélioration apportée par rapport à *DBF* (*Distributed Bellman-Ford*) est l'utilisation de numéros de séquence permettant aux nœuds mobiles de faire la distinction entre une nouvelle route et une ancienne. La suppression des paquets de contrôle dont l'information de routage est déjà connue permet d'éviter qu'un paquet ne tourne en boucle dans le réseau. Les mises à jour des tables de routage sont envoyées périodiquement dans le réseau afin de maintenir la consistance des tables.

Ce procédé peut cependant générer un nombre important de messages de contrôle sur le réseau entrainant ainsi une utilisation inefficace des ressources. Afin de palier à ce problème *DSDV* définit deux types de paquet de mise à jour. Le premier est une mise à jour complète (*full dump*) des informations de routage et consiste en l'émission entière de sa table de routage. Durant les périodes de mouvement occasionnel, les mises à jour complètes sont rares et seul des paquets de mise à jour incrémentale (*incremental*) sont utilisés pour refléter le dernier changement. Ce dernier procédé est illustre à la figure 2.4 pour la mise a jour des informations de routage concernant le nœud *N8*. Dans le tableau 2.1, chaque ligne de la table de routage comporte l'adresse de destination, son numéro de séquence et le nombre de sauts pour y parvenir.

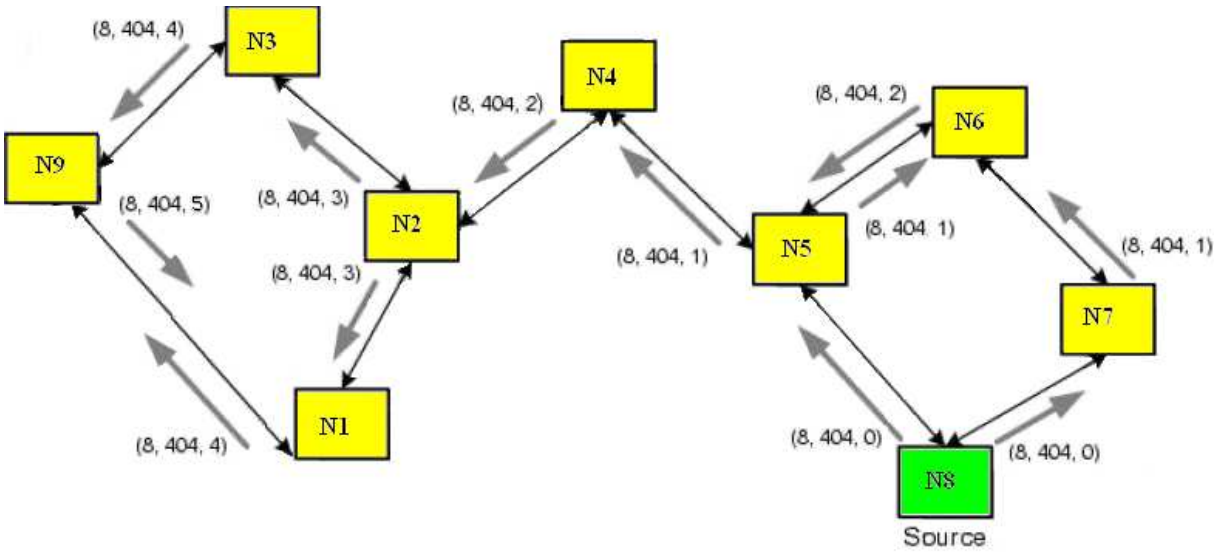


Figure 2.4 - Diffusion des informations de routage du nœud *N8* dans *DSDV*, mise à jour incrémentale

Destination	Prochain saut	Distance	Numéro de séquence
1	1	1	12
2	1	2	22
3	3	1	46
4	3	3	82
5	3	4	134
6	3	5	244
7	3	6	362
9	3	5	404

Tableau 2.1 - Table de routage du nœud *N9* dans *DSDV*

L'actualisation des entrées de la table de routage se base sur le numéro de séquence. À la réception d'un paquet de mise à jour, une comparaison s'opère entre le numéro de séquence sauvé et celui contenu dans le paquet. Si ce dernier est plus grand, il reflète alors une information plus fraiche et l'entrée est directement remplacée par celle du paquet. En cas d'égalité des numéros de séquence, la métrique du protocole étant le nombre de saut, seul la route formée par le plus petit nombre de saut sera retenue.

Le principal défaut de *DSDV* réside dans la lente convergence des tables de routage. Ce problème survient pour deux raisons, d'une part les échanges de vecteurs de distance ne sont pas synchronisés, et d'autre part une route moins coûteuse est remplacée dans la table de routage par une route plus coûteuse, si et seulement si, le nœud ayant publié les deux routes est le même. Ce défaut de *DSDV* est hérité du protocole *RIP* [32] (*Routing Information Protocol*) dont il s'inspire, et qui est utilisé pour le routage dans les réseaux filaires.

2.4.1.2 Le protocole de routage *WRP*

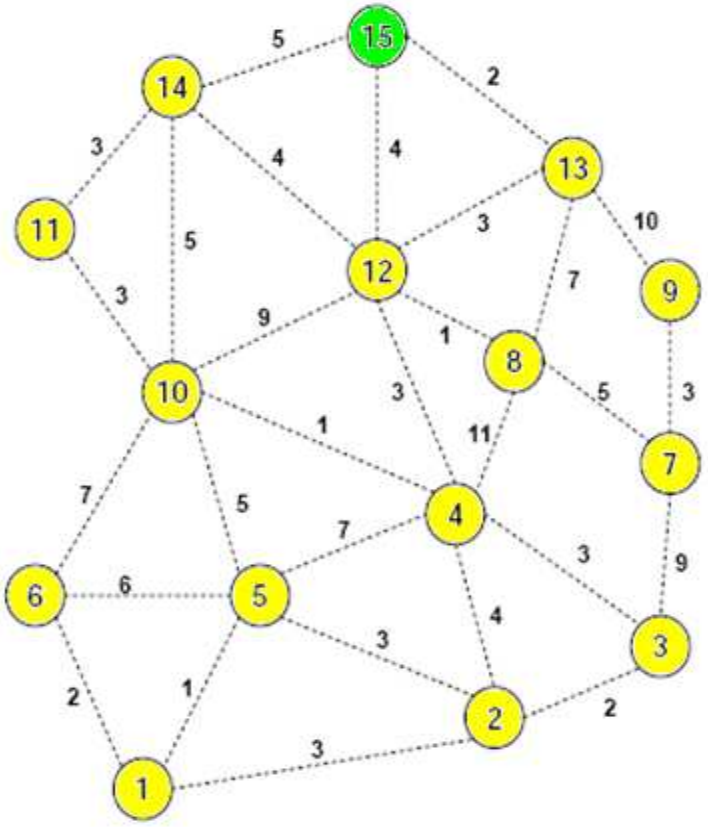
Le principe du protocole de routage sans fil *WRP* [33, 34] (*Wireless Routing Protocol*) est sensiblement identique à *DSDV* [13] mais il utilise, en plus des informations de distance et d'âge des informations, la notion de cout de route (la latence entre le nœud et les différents destinataires) et la table de retransmission des messages (avec les informations de mise à jour). *WRP* utilise le nœud prédécesseur, correspondant au plus court chemin choisi, pour atteindre la destination et afin d'éliminer les boucles. Il se distingue du protocole *DSDV* par une diminution de la mise à jour et de la maintenance des routes.

Dans ce protocole, chaque nœud maintient : une table de distance, une table de routage, une table de coûts des liens et une liste de retransmission de messages *MRL* (*Message Retransmission List*).

- **La table de distance**, est une matrice qui contient la distance et le prédécesseur pour chaque destination de chaque voisin de la source.
- **La table de routage d'un nœud**, garde la trace de toutes les destinations. Dans cette table, chaque entrée spécifie : l'adresse de la destination, la distance vers la destination, le nœud prédécesseur de la destination (correspondant au plus court chemin choisi pour atteindre la destination), le nœud successeur de la source et enfin une marque ou étiquette utilisée dans la mise à jour de la table de routage pour spécifier si l'entrée correspond à un chemin simple (*correct*), une boucle (*error*), ou à une destination qui n'a pas été marquée (*null*). Le tableau de la figure 2.5 de la page suivante représente les valeurs des entrées pour chaque table de routage correspondant à la destination *15*.

- **La table des coûts des liens d'un nœud**, contient le nombre de sauts pour atteindre une destination quand le chemin est valable. Un mécanisme de détection des liens défaillants utilise les durées périodiques de mise à jour (les timeouts ou les délais de garde). Le coût d'un lien défaillant est considéré comme étant infini [14].

- **La liste de retransmission de messages (*MRL*)**, permet à un nœud donné, de connaître l'ensemble de voisins qui n'ont pas acquitté son message de mise à jour, et de retransmettre ce message à cet ensemble de voisins.



Node	NextNode	Pred	Cost
15	15	15	0
14	15	14	5
13	15	13	2
12	15	12	4
11	14	14	8
10	4	12	8
9	13	13	12
8	12	12	5
7	8	12	10
6	10	12	15
5	10	12	13
4	12	12	10
3	4	12	7
2	4	12	11
1	2	12	14

Figure 2.5 – Entrées des tables de routage vers la destination *15*

Un nœud envoi un message de mise à jour, s'il détecte un changement d'état d'un lien voisin, ou après la réception des données de mise à jour d'un autre voisin. Les nœuds présents dans la liste de réponse (*Response List*) du message de mise à jour formé par *MRL*, doivent acquitter la réception du message. S'il n'y a pas de changement, dans la table de routage, par rapport à la dernière mise à jour; le nœud doit envoyer un message "*Hello*" pour assurer la connexion. Lors de la réception du message de mise à jour, le nœud modifie sa distance et cherche les meilleurs chemins en se basant sur les informations reçues. La liste *MRL*, doit être mise à jour après chaque réception d'un acquittement "*ACK*".

Le protocole de routage *WRP* [33, 34] est caractérisé par sa vérification de la consistance des voisins, à chaque fois où un changement d'un lien voisin est détecté. La manière avec laquelle il applique la vérification de la consistance des liens, aide à éliminer les situations des boucles de routage et à minimiser le temps de convergence du protocole.

2.4.1.3 Le protocole de routage *CGSR*

CGSR [39] (*Cluster-head Gateway Switch Routing*) est un protocole de routage proactif ou les nœuds mobiles sont décomposés en groupes appelés « *clusters* ». Un algorithme distribué [40] est utilisé pour élire un membre de chaque groupe le « *cluster head* », l'ensemble des nœuds à sa portée appartienne alors au même *cluster*.

Ce groupage en *cluster* introduit une forme de routage hiérarchique et permet une différentiation au sein de chaque *cluster* du routage, de l'accès au canal, et de l'allocation de bande passante. Un nœud à portée radio de plusieurs *cluster head* est appelé un « nœud de liaison » ou « *gateway* ». Le *cluster head* est chargé du contrôle d'un groupe de nœuds mobiles, ce qui signifie qu'il est chargé de la diffusion (*broadcast*) au sein du *cluster*, de la retransmission des paquets et du *scheduling* de l'accès au canal, tel un *Access Point*. L'allocation du canal se fait par l'utilisation d'un jeton que le *cluster head* se charge de remettre aux nœuds désireux de communiquer. Chaque nœud maintient deux tables :

- une table des membres des *clusters*, qui contient l'adresse du « *cluster head* » de chaque nœud du réseau. Cette table est broadcastée périodiquement par chaque nœud par le protocole *DSDV* [13].

- une table de routage contenant le nœud à emprunter pour joindre une destination.

Lorsqu'un nœud désire envoyer un paquet, il consulte ses deux tables pour découvrir l'adresse du « *cluster head* » de ce nœud ainsi que le chemin à emprunter pour l'atteindre. Il transmet ensuite son paquet à son *cluster head* qui le relayera vers le bon *gateway*. Le paquet voyagera de *gateway* en « *cluster head* » jusqu'à ce qu'il ait atteint le « *cluster head* » du destinataire qui se chargera de lui remettre le paquet.

La figure 2.6 de la page suivante montre l'organisation du réseau en groupes de « *clusters* » dont lesquels on retrouve les nœuds internes, les nœuds de liaisons et les représentants de groupes.

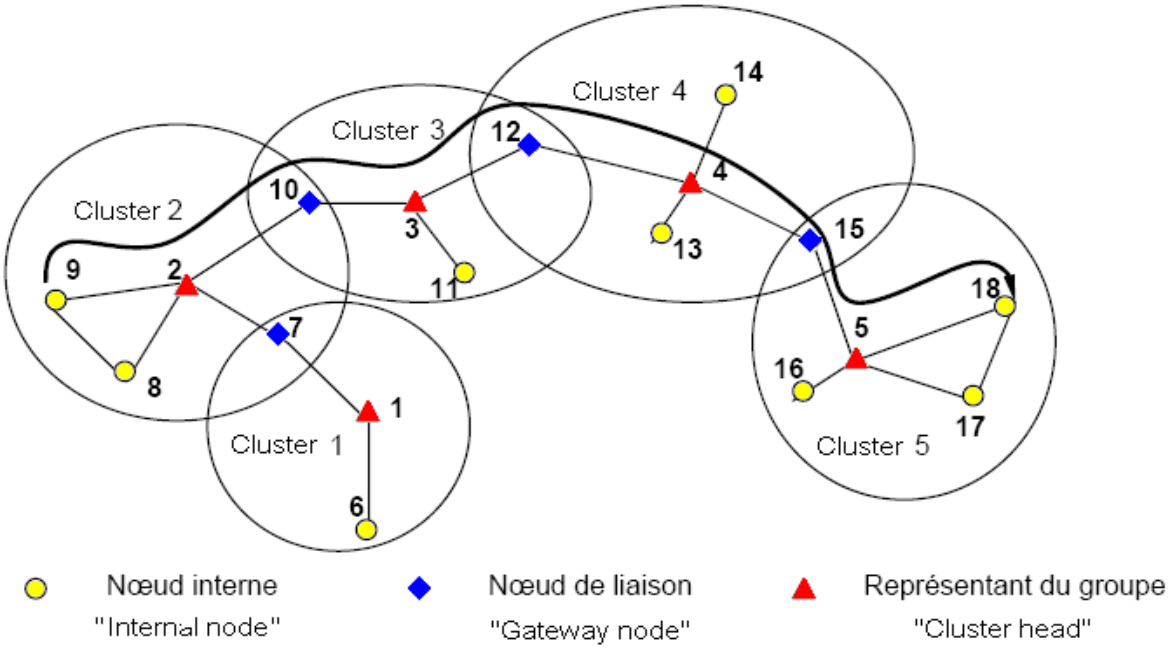


Figure 2.6 – Routage dans *CGSR* [39]

Le protocole *CGSR* obtient de bien meilleures performances que *DSDV* dans des conditions de faible mobilité. Ceci est du essentiellement à la réduction de la taille des table de routage par la conservation d'une seule entrée pour chacun des destinataires d'un même cluster. Ceci à pour effet de limiter le nombre de paquets diffusé entre les différents *clusters*.

L'accès au média par l'acquisition d'un jeton réduit aussi considérablement le taux de collision au sein d'un *cluster* et permet de donner la priorité aux flux dont les contraintes temporelles sont fortes. En contre partie, les « *cluster heads* » et *gateways* sont sollicités d'avantage et constituent un goulot d'étranglement. Une mobilité accrue des nœuds entraine de la complexité dans l'élection du « *cluster head* » et un nombre important d'échanges de messages. Cela rend *CGSR* instable dans un environnement à forte mobilité et le réseau ne peut pas être étendu à une échelle plus grande (*scalable*).

2.4.1.4 Le protocole de routage *GSR*

Le protocole *GSR* [15] (*Global State Routing*) utilise une approche basé sur le routage traditionnel à état de liens (*Link Stat*) et une extension de l'algorithme distribué de *BELLMAN-FORD* (*DBF*) similaire au protocole *DSDV* [13]. Comme dans le cas des protocoles à état de liens, le *GSR* utilise une vue globale de la topologie du réseau basé sur les échanges périodique d'informations d'état entre les différents nœuds du réseau. Ceci est similaire aux méthodes utilisées par *DSDV* [13] pour la dissémination d'informations de routage et en évitant ainsi le mécanisme d'inondation des messages de routage.

Chaque nœud du réseau maintient les informations de routage dans un ensemble de quatre listes et tables : une liste de voisins (*A*), une table de topologie (*TT*), une table des nœuds suivants (*NEXT*) et une table de distance *D*. La liste (*A*) de voisins identifie tous les nœuds adjacents au nœud actuel. La table de la topologie *TT*, contient pour chaque destination, l'information de l'état de lien *TT.LS* telle qu'elle a été envoyée par la destination, et une estampille de l'information *TT.SEQ*. Cette estampille permet d'évaluer la consistance des informations de routage contenues dans la table. Les tables *NEXT* et *D* contiennent une entrée pour chaque nœud destination, vers lequel les paquets seront envoyés. La table de distance, contient la plus courte distance pour chaque nœud destination.

Les messages de routage sont générés suivant les changements d'états des liens et sont envoyés aux nœuds voisins seulement. Lors de la réception d'un message de routage, le nœud met à jour sa table de topologie et cela dans le cas où le numéro de séquence du message reçu serait supérieur à la valeur du numéro de séquence sauvegardée dans la table sinon le message est rejeté (le nouveau message est plus ancien que celui de la table).

Par la suite, le nœud reconstruit sa table de routage et diffuse les mises à jour à ses voisins. Le calcul des chemins, peut se faire avec n'importe quel algorithme de recherche des plus courts chemins. Par exemple, dans les travaux de *Tsu-Wei Chen* et *Mario Gerla*, l'algorithme du *GSR* [15] utilise l'algorithme de *Dijkstra* [37] modifié de telle façon qu'il puisse construire la table des nœuds suivants (*NEXT*) et la table de distance (*D*), en parallèle avec la construction de l'arbre des plus courts chemins (l'arbre dont la racine est le nœud source).

La différence capitale entre le *GSR* et le *Link State* traditionnel, est la façon dans laquelle les informations de routage circulent dans le réseau. Dans le *Link State*, les paquets d'états de liens sont générés et diffusés par inondation dans tout le réseau lorsqu'on détecte des changements de la topologie. Par contre, le GSR maintient la table la plus récente d'état des liens reçus du voisinage direct, et l'échange d'une façon périodique avec ses voisins locaux. En plus de cela, le GSR assure plus de précision, concernant les données de routage qui s'échangent dans le réseau.

2.4.1.5 Le protocole de routage *FSR*

Le protocole *FSR* [16] (*Fisheye link State Routing*) est une extension du protocole *GSR* [15] décrit précédemment dans la section 2.4.1.4. Les deux protocoles utilisent les mêmes méthodes et mécanismes de maintenance des routes. Cependant, *FSR* [16] modifie la fréquence avec laquelle les informations de mise à jour d'état des liens *(link state)* sont envoyées, basé sur la distance des nœuds destinataires, comme le montre la figure 2.7 de la page suivante.

La réduction du volume des données de mise à jour, est obtenue en utilisant des périodes d'échanges différentes pour les différentes entrées de la table. Les entrées qui correspondent aux nœuds les plus proches sont envoyées aux voisins avec une fréquence élevée, représenté par la zone en gris. Le reste des entrées, est échangé avec une fréquence moins élevée. De cette manière, un grand nombre de message de routage est évité, ce qui réduit le volume des paquets qui circule dans le réseau.

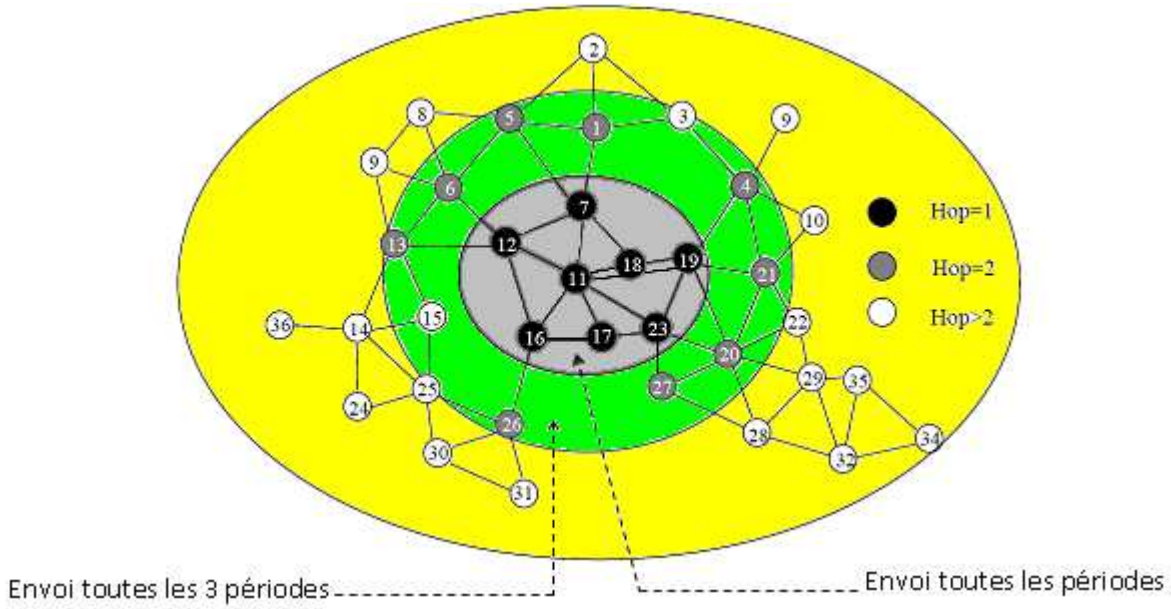


Figure 2.7 - Régions de mise à jour des informations dans *FSR* [16]

Le protocole *FSR* est basé sur l'utilisation de la technique "œil de poisson" (*fisheye*), proposée par *Kleinrock* et *Stevens* [38], qui l'ont utilisé dans le but de réduire le volume d'information nécessaire pour représenter les données graphiques. L'œil d'un poisson capture avec précision, les points proches du point focal. La précision diminue quand la distance, séparant le point vu et le point focal, augmente. Dans le contexte du routage, l'approche du "*fisheye*", pour un nœud, matérialise le maintien des données concernant la précision de la distance et la qualité du chemin d'un voisin direct, avec une diminution progressive, du détail et de précision, quand la distance augmente.

Le protocole *FSR* [16] peut être utilisé dans les réseaux Ad-Hoc dont le nombre d'unités mobiles est grand. Le protocole utilise un volume raisonnable de messages de contrôle, en outre, il évite le travail énorme de recherche de chemins, effectué dans les protocoles réactifs; ce qui accélère la transmission. En plus de cela, le *FSR* maintient des calculs précis concernant les destinations proches.

2.4.1.6 Le protocole de routage *HSR*

Le protocole de routage à état hiérarchique *HSR* [17] « *Hierarchical State Routing* » est basé sur le cheminement multi-niveaux de groupes. Il maintient une topologie hiérarchique logique en employant le groupement périodique des nœuds. Les nœuds au même niveau logique sont groupés dans des *clusters*. Les chefs de groupes « *Clusters heads* » élus par le niveau plus bas vont être des membres du niveau plus élevé. Ces nouveaux membres s'organisent à leur tour dans des *clusters*, et ainsi de suite. Le but principal de ce groupement est d'utiliser efficacement le support de communication et de réduire les messages de contrôle de routage à chaque niveau (c.-à-d. le stockage, le traitement et la transmission des tables de routage).

Un exemple illustratif d'une structure hiérarchique à trois niveaux est représenté dans la figure 2.8 de la page suivante. Dans la décomposition en *clusters*, il y a trois types de nœuds : un nœud représentant du groupe « *cluster head* » ou tête du groupe (par exemple, les nœuds *1, 2, 3* et *4* de la figure 2.8) ; un nœud de liaison « *gateway* », qui relie deux *clusters* (exemple, les nœuds *6, 7, 8* et *11*); et un nœud interne qui n'a aucun rôle spécial (exemple, les nœuds *5, 9, 10* et *12*). Le nœud « *cluster head* » représentant d'un groupe donné, peut être vu comme un coordinateur de transmission de données.

Dans le protocole *HSR* [17], le premier niveau de groupement est appelé niveau physique. Chaque nœud surveille l'état des liens de chaque voisin et les diffuse dans le *cluster*. Le « *cluster head* » propage l'information d'état des liens de son *cluster* aux « *cluster heads* » des clusters voisins par l'intermédiaire des nœuds de liaisons « *Gateways* ». La connaissance de la connectivité entre les « *cluster heads* » conduit à la formation des *clusters* de niveau supérieur.

Par exemple dans la figure 2.8, les « *cluster heads* » des différents *clusters C1-1, C1-2, C1-3* et *C1-4* deviennent des membres dans les *clusters* du niveau *2*. Les états de liens des nœuds du niveau *2* sont virtuels. Un lien virtuel entre les nœuds voisins *1* et *2* comprend le chemin du niveau *1* du « *cluster head* » *1* au « *cluster head* » *2* via le nœud passerelle *6*. En appliquant la même manière de groupement, de nouveaux « *cluster heads* » sont élus à chaque niveau et deviennent membres des *clusters* du niveau plus haut [39, 40]. Après obtention des informations d'états des liens à un niveau, chaque nœud virtuel l'inonde aux nœuds des *clusters* plus bas. De cette façon, chaque nœud physique possède l'information hiérarchique de la topologie du réseau.

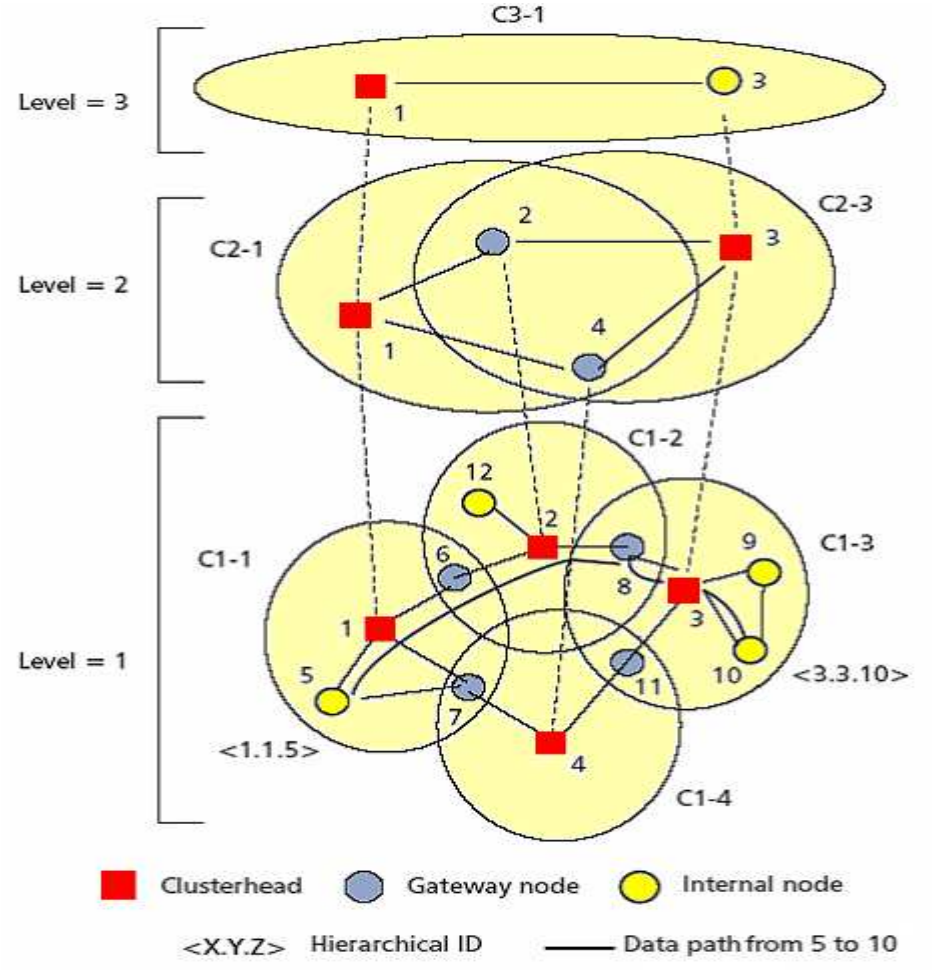


Figure 2.8 – Partitionnement du réseau en groupes dans *HSR* [17]

La hiérarchie ainsi développée nécessite une nouvelle adresse pour chaque nœud mobile, c'est l'adresse hiérarchique. Les identificateurs (*IDs*) des nœuds représentés dans la figure 2.8 (niveau *1*), sont des adresses physiques (unique pour chaque nœud, par exemple : des adresses *MAC*).

Une des méthodes qui peuvent être appliquées afin d'associer des adresses hiérarchiques, ou *HIDs* (*Hierarchical IDs*), aux différents nœuds; est de prendre les numéros des groupes, dans le chemin reliant la racine et le nœud en question. Par exemple le nœud *5* de la figure précédente possède l'adresse hiérarchique *HID*(*5*) égale à *<1.1.5>*, le chemin vertical reliant la racine et le nœud *5*, ainsi il est composé de *3* nœuds : le représentant du groupe *C2-1* (d'où la première composante est *1*, c.à.d. le numéro du groupe), le représentant du groupe *C1-1* (d'où la deuxième composante est *1*), et enfin le nœud *5* d'*ID* égal à *5* d'où la dernière composante de l'adresse est *5*.

L'avantage de l'adressage hiérarchique, est le fait que chaque nœud puisse dynamiquement et localement mettre à jour son *HID*, lors de la réception des données de mise à jour du routage, provenant des nœuds de niveau supérieurs. Un nœud de liaison « *Gateway* », peut être atteint à partir de la racine en suivant plusieurs chemins, par conséquent, ce genre de nœud peut avoir plus d'une adresse hiérarchique. Cela ne pause aucun problème, car le nœud peut être atteint à travers ces adresses, et ces dernières sont associées à un nœud unique. On peut toujours trouver une manière d'associer une seule adresse à ce genre de nœuds, par exemple en prenant la plus petite valeur des numéros de groupes dans les quels appartient le nœud. Exemple : *<1.1.7>* est une adresse du nœud de liaison d'*ID 7*.

Dans la figure 2.8, le nœud mobile *3* est membre du groupe hiérarchique le plus élevé (niveau *3*), il est aussi, le représentant « *cluster head* » du groupe *C2-3*. Le nœud *2* est un membre du groupe *C2-1*, et en même temps il est le représentant du groupe *C1-2*. L'adresse hiérarchique, suffit pour délivrer les paquets de données à une destination, indépendamment de la localisation de la source, et cela en utilisant la table *HSR*. Prenant comme exemple le nœud *5* (figure 2.8) comme source, et le nœud 10 comme destination. Les adresses de ces nœuds sont respectivement : *HID*(*5*) = *<1.1.5>* et *HID*(*10*) = *<3.3.10>*. Pour acheminer une information du nœud 5 vers le nœud 10, le nœud 5 envoie l'information au nœud supérieur, qui le suit hiérarchiquement, i.e. le nœud *d'ID 1*. Le nœud *1*, délivre l'information au nœud *3* qui suit le nœud destination dans l'ordre hiérarchique. Un "lien virtuel", existe entre les deux nœuds *1* et *3*, qui est matérialisé par le chemin (*1.6.2.8.3*); par conséquent, l'information suivra ce chemin pour atteindre la destination. Dans la dernière étape, le nœud *3*, délivre l'information au nœud *10*, en suivant le chemin hiérarchique qui lui relie avec la destination, dans notre cas, ce chemin se réduit en un seul saut.

2.4.1.7 Le protocole de routage *OLSR*

Le protocole *OLSR* [18] (*Optimized Link State Routing Protocol*) est un protocole de routage proactif parce que chaque nœud maintient une table de routage qui contient pour chaque destination dans le réseau un chemin. Cette table est périodiquement maintenue à jour, par l'échange de messages de contrôle entre les différents nœuds dans le réseau. Ces messages de contrôle sont principalement regroupés dans deux groupes.

Les messages de contrôle du premier groupe, portants le nom « messages *Hello* » sont périodiquement envoyés sur une distance d'un saut, pour garantir la détection des voisins mais aussi pour permettre la sélection des nœuds *MPR* (*Multi Point Relay*), qui ont pour rôle de transmettre les messages de contrôle au deuxième groupe. La figure 2.9 de la page suivante montre comment des nœuds voisins sont sélectionnés pour devenir *MPR*.

Les messages de contrôle de topologie « *TC - Topology Control* » du deuxième groupe sont périodiquement diffusés dans le réseau à travers les nœuds *MPR* afin de garantir pour chaque nœud une vue globale sur la topologie du réseau. Les nœuds *MPR* ont pour but de réduire le nombre de messages de contrôle dans le réseau.

Dans *OLSR* [18], quand un nœud source « *S* » cherche une route pour acheminer des données vers une destination « *D* », il n'a pas à chercher le chemin vers cette dernière, puisqu'il doit en avoir déjà une dans sa table de routage. Ce type de routage, contrairement au routage réactif, n'a pas le problème des longs délais de communication puisque les chemins sont préconstruits et prêts à êtres utilisés à tout moment. Par contre, il existe un coût constant dans le réseau pour diffuser les informations de routage. Même si certains algorithmes réduisent la quantité d'information, cette occupation d'une partie de la ressource radio réduit la quantité disponible pour les communications entre mobiles.

En plus, ce type de protocole n'est pas adapté pour de grands réseaux. En effet, la quantité d'information à diffuser pour une maintenance cohérente augmente proportionnellement au nombre de nœuds. Finalement, une mobilité importante dans le réseau peut entraîner des risques de perturbations énormes. En effet, la quantité d'information devient trop importante pour le réseau car le nombre de messages est fonction des changements topologiques.

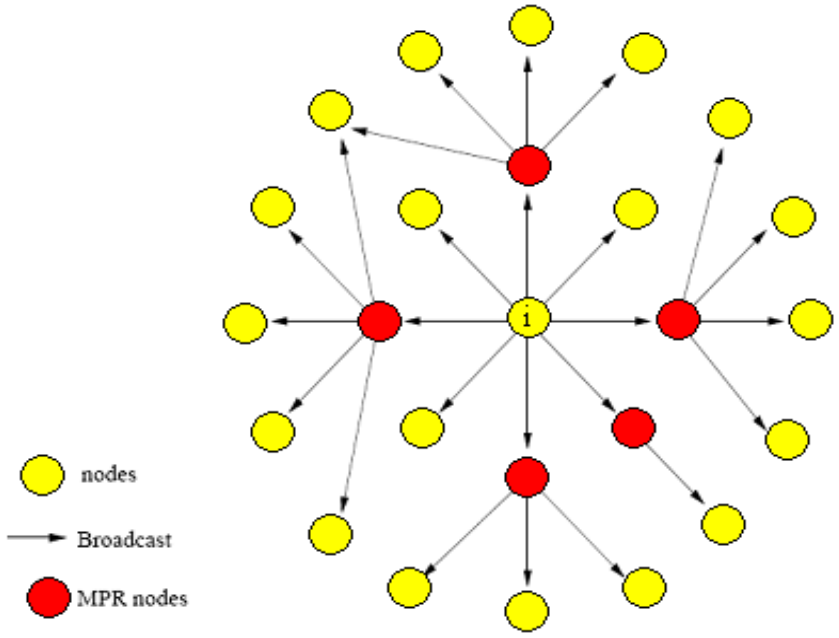


Figure 2.9 - Sélection des nœuds *MPR* dans *OLSR* [18].

2.4.2 Les protocoles de routage réactif

2.4.2.1 Le protocole de routage *DSR*

Le protocole de routage *DSR* [19] (*Dynamic Source Routing*) est basé sur le principe de diffusion à la demande pour calculer une route vers une destination. Il utilise le concept du routage à la source qui signifie que la le nœud source connait l'intégralité du chemin vers le nœud destination. Ainsi, chaque paquet transporte dans son entête la liste complète des nœuds du chemin donc les nœuds intermédiaires n'ont pas besoins de garder les informations de routage.

Les principales procédures dans le protocole *DSR* [19] sont : la découverte de route (*route discovery*) et la maintenance de route (*route maintenance*). Chaque nœud possède un cache pour conserver toutes les routes non expirées. Lorsqu'un nœud mobile cherche une route pour une destination, il consulte d'abord son cache pour vérifier s'il existe une route à cette destination. Si la vérification échoue, le nœud mobile diffuse un paquet de découverte de route *RREQ* (*route request*). Lorsque le nœud reçoit un paquet *RREQ*, il rajoute sa propre adresse au chemin de la route et rediffuse le paquet *RREQ* s'il n'atteint pas la destination. Finalement, le paquet *RREQ* arrive à la destination ou à un nœud intermédiaire qui connait la destination. Le nœud destination ou intermédiaire va retransmettre un paquet *RREP* (*route reply*) à la source. Pour envoyer le paquet *RREP*, le nœud destination ou intermédiaire a besoin du chemin à la source. La figure 2.10 de la page suivante, représente un exemple de description du processus de découverte de route dans le protocole *DSR* [19].

Il existe trois cas de figures pour trouver le chemin inverse :

- Premier cas, si le nœud destination ou intermédiaire peut trouver le chemin à la source dans son cache, il peut donc utiliser ce chemin.

- Deuxième cas, si le chemin à la source ne se trouve pas dans son cache mais les liens symétriques sont supportés, le nœud peut inverser le chemin qui se trouve dans le paquet *RREQ*.

- Troisième cas, si les liens symétriques ne sont pas supportés, le nœud peut effectuer une nouvelle découverte de route pour trouver un chemin à la source.

Pour réaliser la maintenance de route, *DSR* utilise les parquets *RERR* (*route error*). Le nœud génère un paquet *RERR* quand il détecte une rupture de lien. Quand un notre nœud reçoit le paquet *RERR*, il supprime le nœud erroné du cache par conséquent tous les chemins contenant ce nœud seront tronqué à ce niveau (nœud erroné).

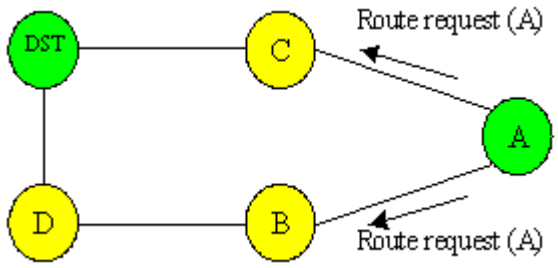
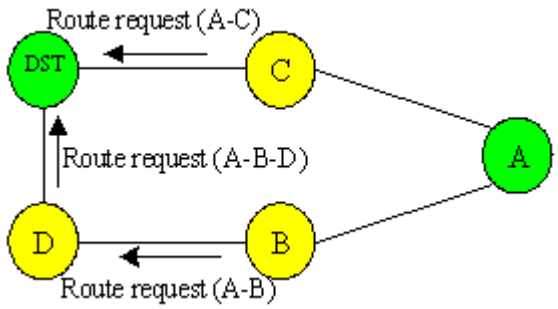
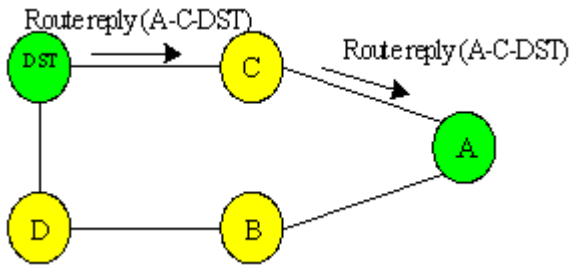
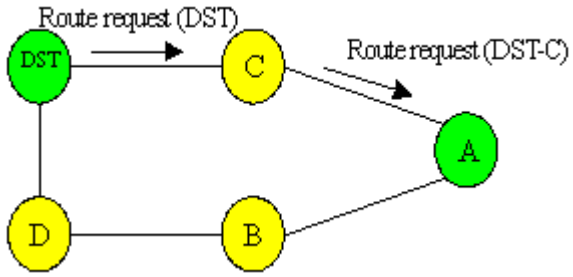
	*A* cherche un une route à *DST*. *A* diffuse le paquet *RREQ*. Le chemin enregistré durant la découverte de route est mis entre parenthèses.
	Les nœuds intermédiaires *B*, *C* et *D* insèrent leurs adresses dans le paquet de découverte de route *RREQ* et les rediffusent. On suppose que *B*, *C* et *D* ne connaissent pas la route vers la *DST*.
	Le premier paquet de découverte contenant le chemin arrive au nœud *DST* puis ce dernier l'envoi à la source *A*. Dans ce cas de figure, on suppose que les liens symétriques sont supportés.
	Dans cette figure, on suppose que les liens entre *DST* et *C* ne sont pas symétriques. Donc, *DST* initie un nouveau processus de découverte de route à la source *A*.

Figure 2.10 - les opérations de découverte de route dans *DSR* [19]

Le principal avantage dans *DSR* est qu'il ne nécessite pas des liens symétriques entre les différents nœuds. Quand il ne trouve pas de liens symétriques, *DSR* peut utiliser des liens asymétriques.

2.4.2.2 Le protocole de routage *AODV*

Le protocole *AODV* [23] (*Ad-hoc On-Demand Distance Vector*) est fondé sur l'algorithme *DSDV* [13] décrit précédemment. Généralement, *AODV* est considéré comme une amélioration par rapport à *DSDV* parce qu'elle réduit au minimum le nombre de diffusion de requêtes requises par la création d'itinéraires sur la base de la demande de route, par opposition au maintien d'une liste complète des routes, comme dans l'algorithme *DSDV*. Les auteurs de *AODV* le classe comme un simple système d'acquisition de route sur demande, comme les nœuds qui ne sont pas sur un chemin d'accès ne maintiennent pas les informations de routage.

Quand un nœud source désire envoyer un message à un nœud destination dont il n'a pas déjà un itinéraire valide, il entame un processus de découverte de chemin pour localiser ce nœud destination. Il diffuse un paquet de découverte de route *RREQ* (*route request*) à ses voisins directs, qui à leurs tour rediffusent le paquet de découverte de route *RREQ* à leurs voisins, et ainsi de suite, jusqu'à ce que le nœud destination soit localisé ou bien jusqu'à ce qu'un nœud intermédiaire avec une route à la destination "assez fraîche" soit atteint. La figure 2.11.a de la page 46 illustre la propagation des requêtes de découverte *RREQ*s de routes à travers le réseau.

Cependant, l'*AODV* [23] maintient les chemins d'une façon distribuée en gardant une table de routage, au niveau de chaque nœud de transit appartenant au chemin cherché. Une entrée de la table de routage contient essentiellement :

- L'adresse de la destination.
- Le nœud suivant.
- La distance en nombre de nœud (c'est à-dire, le nombre de nœud nécessaire pour atteindre la destination).
- Le numéro de séquence destination qui garantit qu'aucune boucle ne peut se former.
- Liste des voisins actifs (origine ou relais d'au moins un paquet pour la destination pendant un temps donné).
- Le temps d'expiration de l'entrée de la table (temps au bout duquel l'entrée est invalidée).
- Un tampon de requête afin qu'une seule réponse soit envoyée par requête.
- A chaque utilisation d'une entrée, son temps d'expiration est remis à jour (temps courant + active route time).

L'*AODV* [23] utilise des numéros de séquence de destination pour s'assurer que tous les itinéraires ne comportent pas de boucles et qu'ils contiennent les informations les plus récentes sur le routage. Chaque nœud a son propre numéro de séquence, ainsi qu'un identificateur de diffusion *ID*. Cet identificateur (*ID*) est incrémenté pour chaque *RREQ* que le nœud initiateur diffuse, et en collaboration avec les autres nœuds, identifie de façon unique un paquet *RREQ*. En plus de son propre numéro de séquence de l'émission et *ID*, le nœud source inclut dans le paquet *RREQ* le numéro de séquence le plus récent qu'il a pour la destination. Les nœuds intermédiaires peuvent répondre à la *RREQ* seulement s'ils ont un itinéraire vers la destination et dont le numéro de séquence est supérieure ou égale à celle contenue dans le paquet *RREQ*.

Au cours du processus de la rediffusion du paquet *RREQ*, les nœuds intermédiaires enregistrent dans leurs tables de routage, l'adresse du voisin à partir de laquelle la première copie de l'émission du paquet est reçue, établissant ainsi un chemin inverse. Si des exemplaires supplémentaires de la même *RREQ* sont reçus plus tard, ces paquets sont rejetés.

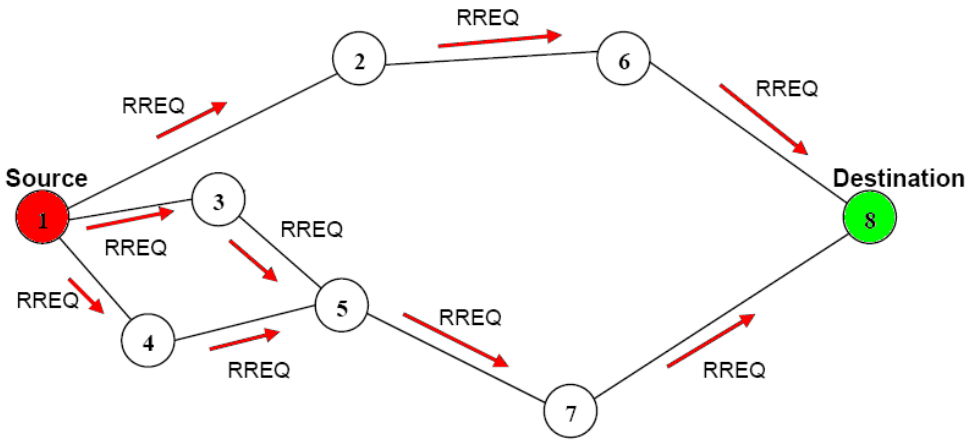
Une fois le *RREQ* atteint la destination ou un nœud intermédiaire avec une route « assez fraiche », la destination ou le nœud intermédiaire répond par l'envoi d'un paquet de réponse *RREP* (*route reply*) en mode unicast au voisin d'où il a reçu en premier la *RREQ* (figure 2.11b). Comme le renvoi du paquet *RREP* est aiguillé le long de la route inverse, les nœuds le long de ce chemin mettent à jour les nouvelles entrées dans la table de routage leur parcours, qui pointent vers le nœud à partir duquel le paquet *RREP* est venu. Ces entrées indiquent le nouvel itinéraire inverse actif. De plus, un temporisateur (*Timer*) est associé à chaque itinéraire dans la table de routage qui entraînera la suppression de l'entrée si elle dépasse la durée de vie spécifiée. Puisque le paquet *RREP* est transmis le long de l'itinéraire établi par le *RREQ*, *AODV* [23] ne supporte que l'usage de liens symétriques.

Format général d'une *RREQ :*

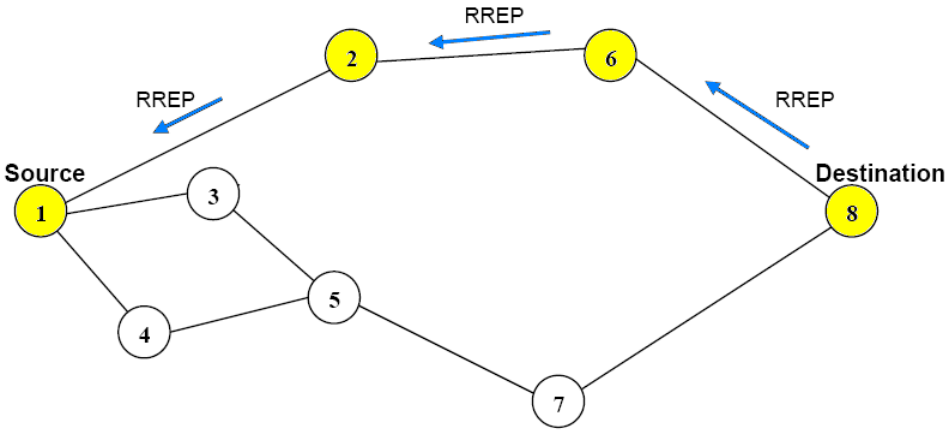
@source	Num. seq. Source	Broadcast id	@destination	Num. seq. Destination	Nombre de sauts

Format général d'une *RREP :*

@source	@destination	Num. seq. destination	Nombre de sauts	Durée de vie (TTL)



(a) La propagation du paquet RREQ (requête de route).



(b) Le chemin pris par le paquet RREP (requête de réponse).

Figure 2.11- La découverte de route dans *AODV* [23]

Les routes sont maintenues comme suit. Si un nœud source se déplace, il est en mesure de reprendre le processus de découverte d'un nouvel itinéraire vers la destination. Si un nœud le long de la route se déplace, son voisin en amont détecte ce déplacement et propage un message de notification de rupture de lien (un paquet *RREP* avec une métrique infinie) à chacune de ses voisins actifs en amont pour les informer de l'effacement de cette partie de l'itinéraire. Ces nœuds, à leur tour, propagent la notification de défaillance du lien à leurs voisins en amont, et ainsi de suite, jusqu'à ce que le nœud source soit atteint. Le nœud source peut alors choisir de reprendre la procédure de découverte de nouvelles routes si la communication est encore désirée. Les nœuds écoutent les transmissions de paquets de données pour s'assurer que le saut prochain est toujours à sa portée. Si une telle retransmission n'est pas entendue, le nœud peut utiliser une autre technique, telle que la réception des messages « *Hello* », afin de déterminer si le saut suivant est à portée de communication. Les messages « *Hello* » peut renfermer tout le voisinage à partir desquels un mobile a entendu, ce qui donne une plus grande connaissance de la connectivité réseau.

2.4.2.3 Le protocole de routage *TORA*

Le protocole *TORA* [44] (*Temporary Ordering Routing Algorithme*) est un algorithme de routage distribué très adaptatif, sans boucles, basé sur le concept d'inversement de liens. *TORA* est proposé pour opérer dans un environnement de réseaux de mobiles très dynamique. Il utilise le concept du routage à la source et fournit de multiples chemins pour toute paire source/destination souhaitée.

La principale caractéristique de *TORA* [44], est que les messages de contrôle sont limités à un ensemble réduit de nœuds. Cet ensemble représente les nœuds proches du lieu de l'occurrence du changement de la topologie dans le réseau mobile. Pour réaliser ceci, les nœuds ont besoin de maintenir les informations de routage sur les nœuds adjacents.

Le protocole accomplit trois fonctions essentielles: (a) la création de routes, (b) la maintenance de routes, et (c) l'élimination de routes. Au cours des phases de la création et de la maintenance de routes, les nœuds utilisent la métrique "taille de nœud" pour créer un graphe acyclique dirigé *DAG* (*Directed Acyclic Graph*) orienté destination. Par la suite, des liens sont assignés à une direction (vers la source ou vers la destination) sur la base des métriques relatives aux nœuds voisins, comme le montre la Figure 2.12.a. En période de mobilité des nœuds, les liens du graphe acyclique dirigé *DAG* peuvent être interrompus, et la maintenance de ces routes est nécessaire pour rétablir un *DAG* dirigé à la même destination. Comme le montre la figure 2.12.b, en cas de rupture du lien avec le nœud suivant, un nœud génère un nouveau niveau de référence qui représente le maximum de tailles de tous les nœuds voisins. Le nœud propage ce niveau de référence, ainsi tous ses liens vont être orientés vers ses voisins puisque sa taille est maintenant la plus grande. Les liens sont inversés pour refléter le changement dans l'adaptation au nouveau niveau de référence.

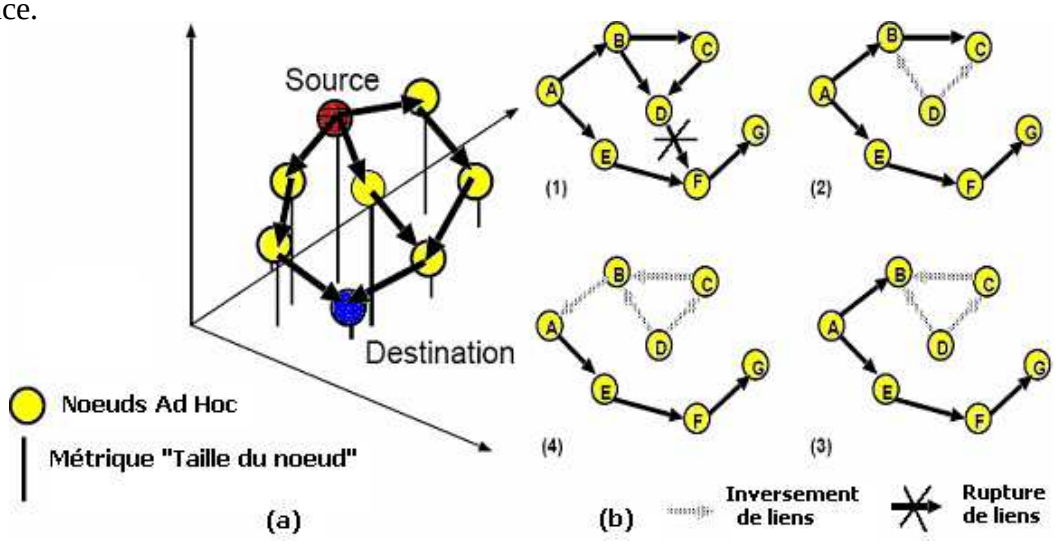


Figure 2.12 (a) - Création de routes (assignation des directions aux liens),
(b) - Maintenance des routes (Processus d'inversement de liens) dans *TORA* [44]

Le Timing est un facteur très important dans le protocole *TORA* [44], parce que la métrique "taille de nœud" dépend du temps logique de la rupture d'un lien ; *TORA* suppose que tous les nœuds ont une horloge synchronisée. La métrique de *TORA* est un quintuplé composé des champs suivants : (a) le temps logique de défaillance, (b) l'unique *ID* du nœud définissant le nouveau niveau de référence, (c) un bit indicateur de réflexion, (d) le paramètre d'ordre de propagation, et (e) l'unique *ID* du nœud. Les trois premiers champs regroupés représentent le niveau de référence.

Un nouveau niveau de référence est définit à chaque détection d'une défaillance dans les chemins entre la source et la destination. La phase de l'élimination de routes est effectuée en propageant un paquet *CLR* (*clear packet*) dans le réseau. Par conséquent tous les chemins défaillants sont supprimés des caches locaux des nœuds.

2.4.2.4 Le protocole de routage *ABR*

Le protocole *ABR* [21] (*Associativity-Based Routing*) est un protocole réactif développé spécifiquement pour tenir compte de la mobilité des nœuds. L'idée principale est d'utiliser le degré d'associativité des nœuds constituant la route comme métrique de routage. L'associativité est liée à la stabilité spatiale et temporelle d'une connexion entre deux nœuds voisins.

Dans un réseau constitué de nœuds mobiles, le chemin le plus court à l'instant τ ne l'est plus forcément a l'instant $\tau + 1$. Pire encore, le chemin le plus court est souvent constitué de lien a la limite de la portée radio des équipements de transmission et sont donc sujet à beaucoup de cassure Partant de ces constatations, l'auteur propose une solution basée sur la stabilité des routes, minimisant ainsi les procédures de maintenance des routes. Plus précisément, le degré d'associativité est mesuré sur un ensemble de paramètres tels que la puissance du signal reçu, le délai du lien, la durée de vie de la batterie et le paramètre « *associativity tick* ». Ce dernier paramètre reflète la stabilité temporelle et spatiale d'un lien. Il s'agit d'un compteur incrémenté périodiquement lorsque des nœuds voisins restent a portée radio mutuelle au fil du temps.

Afin de pouvoir mesurer ces paramètres, chaque nœud envoie périodiquement un « *beacon* », c'est-à-dire des messages *"Hello"* et compte le nombre de *beacons* reçu de ses voisins pour augmenter leur « *associativity tick* » dans les tables. Leur valeur est remise à 0 si aucun message *beacon* n'est reçu durant un certain intervalle de temps. Sur base de ces mesures, le protocole classe alors les liens entre nœuds selon leurs degrés de stabilité et utilise cette nouvelle métrique pour le routage. Cette notion de stabilité des nœuds est très intéressante car elle permet d'avoir un comportement différencié suivant ses voisins : par exemple, pour un réseau où la mobilité est quasi nulle, la stabilité des liens est très grande. Le routage peut donc diminuer de manière importante les fréquences d'émission des messages de contrôle et par conséquent diminuer la consommation énergétique des nœuds.

Le protocole couvre essentiellement trois phases : découverte des routes, reconstruction de route et suppression des routes. La découverte se fait par diffusion d'un paquet *BQ* (*Broadcast Query*) auquel chacun des nœuds traversé ajoute son *ID* et ses mesures quant à la qualité de la route. A la réception du message par le destinataire, celui-ci attend un laps de temps avant de répondre afin de laisser le temps à d'autres *BQ* de parvenir par d'autres routes. Cela permet au nœud destinataire de sélectionner le meilleur chemin et d'envoyer alors un *BQ-REPLY* à la source suivant ce chemin. La figure 2.13 illustre la route retenue entre *1* et *13*, c'est la route *1-2-5-11-13*.

La phase de réparation survient lorsqu'un nœud de la route n'est plus joignable. Contrairement à *DSR* [19], une réparation au point de cassure est mise en place afin rétablir le chemin vers la destination. Si la réparation locale échoue, la source est avertie afin qu'elle puisse initier une nouvelle recherche de route. La procédure de réparation est complexe et dépend du nœud ayant provoqué la cassure. En effet, la routine de réparation invoquée dépend du type de nœud en mouvement, il peut s'agir de la source, la destination ou un nœud intermédiaire.

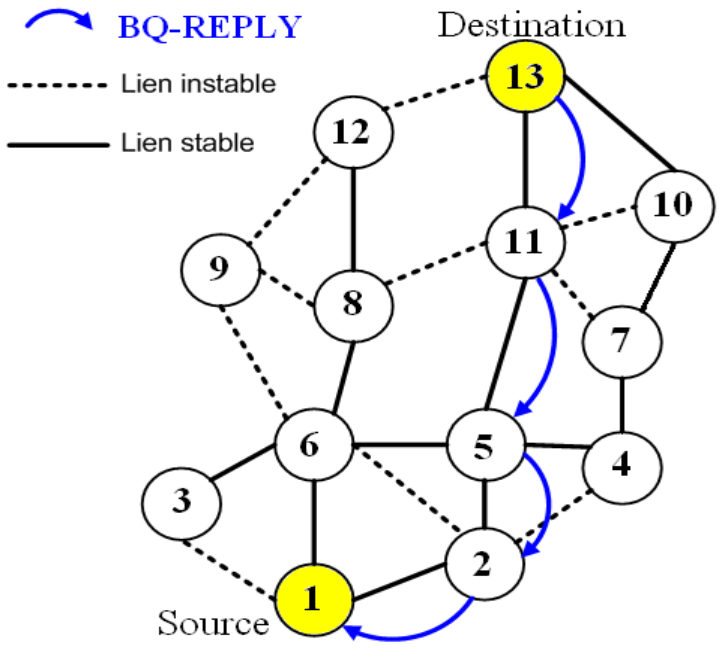


Figure 2.13 - Découverte des routes dans *ABR* [21]

La suppression d'une route a lieu lorsque celle ci n'est plus désirée, la source à la base de sa construction diffuse alors un message *RD* (*Route Delete*) afin de mettre à jour les tables de routage.

2.4.2.5 Le protocole de routage *SSA*

Le protocole *SSA* [22] (*Signal Stability-based adaptive routing algorithm*) est un protocole réactif similaire au protocole *ABR* [21] mais qui étudie, cette fois ci, la puissance du signal en réception. À partir de ces informations, ils peuvent associer une mesure qualitative à chaque connexion et sélectionner des routes sur la base de la stabilité du signal (lien à forte interconnexion) entre les différents nœuds et sur la stabilité de délocalisation.

Il est divisé en deux protocoles coopératifs entre eux: le protocole de routage dynamique *DRP* (*Dynamic Routing Protocol*) et le protocole de routage statique SRP (*Static Routing Protocol*). Le *DRP* est responsable du maintient de la table de stabilité du signal *SST* (*Signal Stability Table*) et de la table de routage *RT* (Routing Table). La table *SST* sauvegarde les puissances des signaux des nœuds voisins, obtenues par l'échange périodique des messages *beacons* avec ces derniers. La puissance d'un signal est sauvegardée dans la table sous l'une des deux formes suivantes: canal de forte puissance (*strong*) ou canal de faible puissance (*weak*).

Le protocole *DRP* fait passer le paquet reçu au protocole *SRP* qui va le router en utilisant la table *RT* vers la destination spécifiée. Si aucune entrée dans la table *RT* relative au nœud destination n'est disponible, le *SSA* [22] lance un processus de recherche de routes en diffusant un paquet requête de route (*route-request)* à travers les canaux à forte puissance. Le nœud destination choisit le chemin du premier paquet requête de route (*route-request)* qui arrive et transmet un paquet de réponse (*route-reply)* vers la source en prenant le chemin inverse parce qu'il y a une grande chance pour que ce dernier ait traversé le plus court chemin et/ou le moins chargé. Si le timeout du message de réponse expire, le nœud source relance de nouveau un processus de recherche de route en autorisant la propagation des paquets à travers les canaux de faible puissance.

Lorsqu'une une rupture de lien est détecté dans le réseau, le nœud détectant la défaillance envoie un message d'erreur au nœud source. La source envoie un message de suppression du chemin défaillant et relance, par la suite, un nouveau processus de recherche de routes pour trouver une nouvelle alternative vers la destination.

2.4.3 Les protocoles de routage hybride

2.4.3.1 Le protocole de routage *ZRP*

Le protocole *ZRP* [24, 48] (*Zone Routing Protocol*) utilise un protocole proactif au niveau local pour joindre les stations situées à une distance inférieure à k sauts et un protocole réactif pour le routage entre les groupes (appelés *routing zone*). L'architecture du protocole *ZRP* est composée de quatre sous-protocoles :

1. *IntrAzone Routing Protocol* (*IARP*),

2. *IntErzone Routing Protocol* (*IERP*) [48],

3. *Bordercast Resolution Protocol* (*BRP*), et

4. *Neighbor Discovery/Maintenance Protocole* (*NDP*) qui se situe dans la 2ème couche.

IARP fournit de manière proactive les routes aux nœuds situés dans la même zone que la source. *IARP* repose sur *NDP* pour découvrir ses voisins, son rôle principal est d'assurer que chaque nœud au sein d'une zone possède une table de routage à jour reflétant la route à emprunter pour chaque nœud de la même zone.

IERP [48] repose sur les nœuds frontières afin de découvrir de manière réactive les routes interzones lors d'une communication entre deux nœuds de zones différentes. De manière plus précise, *IERP* émet des paquets de requête aux nœuds frontières via *BRP*, une sorte d'algorithme multicast, afin qu'ils vérifient si le nœud destinataire fait parti de leur zone respective. Si c'est le cas, ils généreront un paquet de réponse vers la source, dans le cas contraire ils propageront la requête vers leurs nœuds frontières. Ce procédé est illustré à la figure 2.14 de la page suivante.

Le problème de ce protocole est qu'il n'y a pas de coordination entre les nœuds, il en résulte que les zones se chevauchent et un nœud peut être à la fois membre d'une zone et nœud frontière de plusieurs zones. Dans ces conditions, l'algorithme de recherche peut conduire à des résultats moins bons qu'une diffusion standard. Des solutions ont été proposées dans la littérature pour contrôler et stopper la diffusion redondante des paquets de requête. Afin de palier aux problèmes du protocole *ZRP* [24], une amélioration fut proposée le : *Distributed Dynamic Routing algorithm* (*DDR*). Ce protocole est basé sur la construction d'une forêt de zones dynamiques qui ne se chevauchent pas.

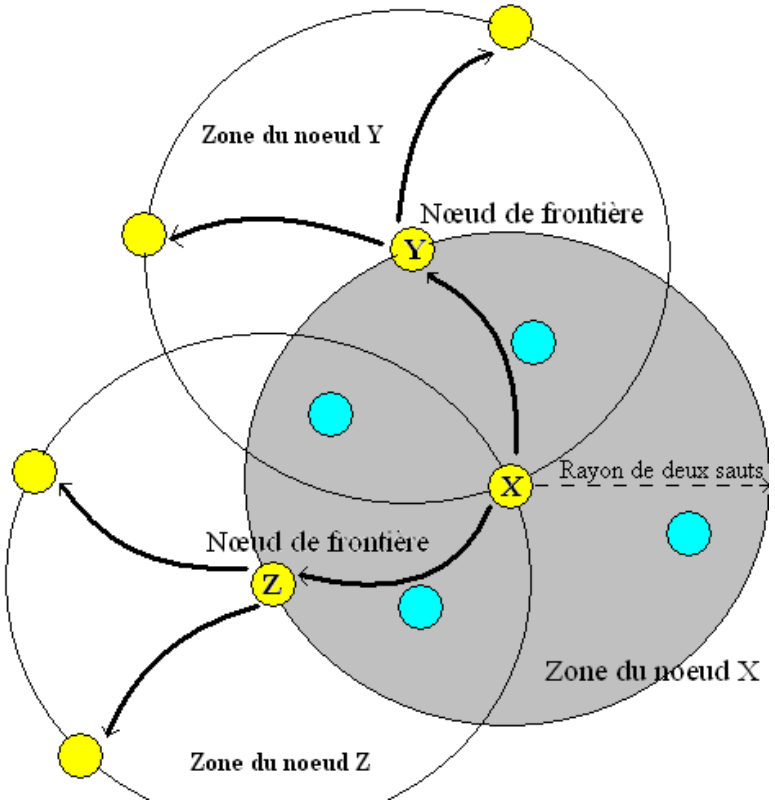


Figure 2.14 — Découverte de routes interzones dans *ZRP* [24]

2.4.3.2 Le protocole de routage *CBRP*

Le protocole *CBRP* [26] (*Cluster Based Routing Protocol*) est un protocole hybride utilisant deux niveaux hiérarchiques plus particulièrement adapté aux réseaux Ad-Hoc caractérisés par une faible mobilité. Il est basé sur une gestion de groupes ou « *Clusters* » difficiles à maintenir si les nœuds sont fortement mobiles. Plus précisément, ce protocole décompose le réseau en groupes ou « *Clusters* » de rayon égal à un saut. Chaque groupe ou « *Clusters* » est composé d'un unique coordinateur (tête de cluster) ayant une complète connaissance du groupe c'est-à-dire des membres du groupe et des liens entre lui-même et les membres du groupe (Figure 2.15, de la page suivante).

Chaque nœud du réseau maintient deux tables : une table de ses voisins adjacents, et une table des groupes adjacents composée de la liste des groupes adjacents et de leur coordinateur respectif. Ces deux tables sont maintenues à jour grâce aux messages « *Hello* » diffusés périodiquement par chaque nœud sur deux sauts. Ce message contient l'état du nœud (du point de vue de son appartenance à un groupe), l'identifiant du coordinateur du (ou des) groupe(s) au(x)quel(s) appartient le nœud, une liste des nœuds voisins adjacents et une liste des groupes adjacents incluant l'identifiant de leur coordinateur respectif.

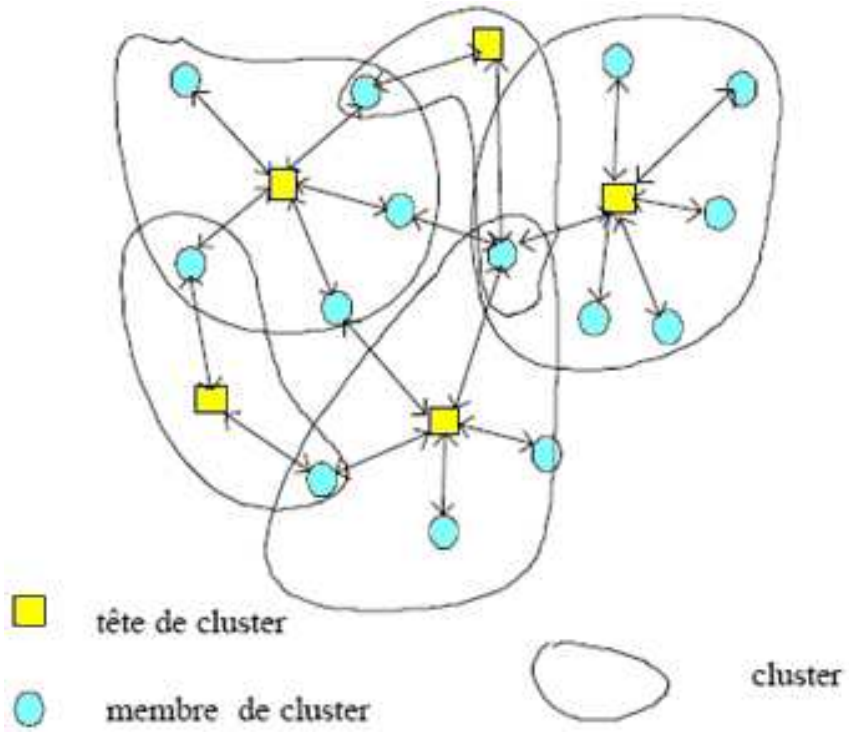


Figure 2.15 — Protocole de Routage Basé sur les Groupes *CBRP* [26]

Lorsqu'un nœud reçoit un message « *Hello* », il rafraîchit ses deux tables. Plus précisément, si l'expéditeur du message « *Hello* » ne fait pas partie de son groupe, il définit l'expéditeur comme étant une passerelle vers un groupe adjacent, et ajoute cette information à sa table de groupes adjacents. Par ailleurs, ces messages « *Hello* » sont utilisés pour élire un coordinateur. En effet, un nœud, n'ayant aucun coordinateur, spécifie dans le message « *Hello* » qu'il cherche un coordinateur, c'est-à-dire que le nœud se trouve dans l'état indécis. Après expiration d'un délai, si ce dernier ne reçoit pas en réponse un message « *Hello* » de la part d'un coordinateur, alors, il se définit comme coordinateur. Dans le cas contraire, il se définit comme membre d'un groupe.

L'utilisation de groupes permet de limiter le trafic généré lors de la localisation d'un nœud. Plus précisément, lorsqu'un nœud souhaite localiser une destination, il diffuse (sur un saut) sa requête contenant : le coordinateur du (ou des) groupe(s) au(x)quel(s) il appartient, la liste des coordinateurs adjacents et des passerelles correspondantes. Les nœuds passerelles (spécifiés dans la requête) font suivre l'information aux coordinateurs adjacents. Ces derniers répondent à la requête si la destination appartient au groupe qu'ils coordonnent. Dans le cas contraire, ils font suivre la requête aux coordinateurs des groupes adjacents.

2.4.3.3 Le protocole de routage *CEDAR*

Le protocole *CEDAR* [25] (*Core Extraction Distributed Ad-Hoc Routing*) fait partie la famille des protocoles de routage hybride qui est adapté au dynamisme rencontré dans les réseaux Ad-Hoc et fournit une qualité de service suivant la métrique bande passante disponible. *CEDAR* repose sur le principe de réseaux de cœur, à la façon des réseaux overlay, certains nœuds sont choisis pour faire partie du cœur du réseau (*Dominating Set*). Des informations sur les liens stables disposant d'une grande bande passante sont propagées entre les nœuds du cœur. Le calcul des routes est effectué par les nœuds du réseau cœur en utilisant des informations locales. L'une des spécificités de *CEDAR* est de limiter au maximum l'inondation du réseau pour les découvertes de routes. Utilisé dans des réseaux de petite à moyenne taille, il est basé sur trois composantes essentielles :

- **Extraction d'un cœur du réseau** (*Core extraction*): Lors de cette phase, un ensemble de nœud (*Dominating Set*) est dynamiquement choisi pour calculer les routes et maintenir l'état des liens du réseau. L'avantage d'une telle approche est qu'avec un ensemble réduit de nœuds les échanges d'informations d'état et de route seront minimisés, évitant ainsi des messages supplémentaires circulant dans le réseau. En outre, lors d'un changement de route, seuls les nœuds du cœur serviront au calcul,

- **Propagation de l'état des liens** (*Link state propagation*): le routage avec qualité de service est réalisé grâce à la propagation des informations sur les liens stables avec une grande bande passante. L'objectif est d'informer les nœuds distants sur les liens de grande capacité, alors que les liens de faible capacité reste connus au niveau local (les nœuds n'ont pas une information sur la topologie globale du réseau),

- **Calcul de route** (*Route computation*): celui-ci est basé sur la découverte et l'établissement d'un plus court chemin vers la destination satisfaisant la bande passante demandée. La première phase consiste à l'établissement du chemin cœur. On cherche le chemin entre le représentant de cœur de *A* et le représentant de cœur de *B* (Figure 2.16, de la page suivante). Le chemin choisi sert de guide pour trouver la route entre *A* et *B.* Dans la deuxième phase on Calcul les routes avec *QoS* en se basant sur sa connaissance locale de la topologie, le représentant de cœur de *A* définit le plus cour chemin et satisfaisant le besoin en bande passante entre *A* et un nœud du réseau. Ainsi de suite, chaque nœud demande à son représentant la meilleure route pour aller à *B*, jusqu'à atteindre *B.*

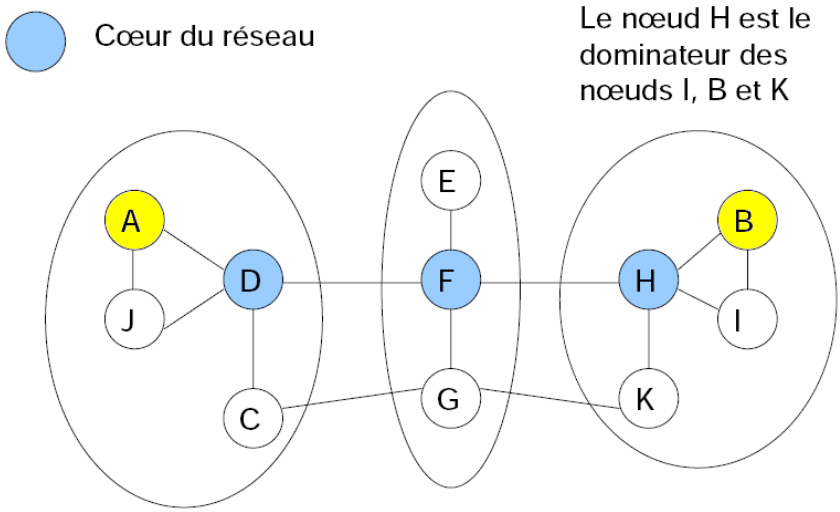


Figure 2.16— Les nœuds cœur du réseau dans *CEDAR* [25]

Des routes de secours sont utilisées lors de la reconstruction de la route principale, quand cette dernière est perdue. La reconstruction peut être locale (à l'endroit de la cassure), ou à l'initiative de la source. Au lieu de calculer une route avec un minimum de saut, l'objectif principal de *CEDAR* [25] est de trouver un chemin stable pour garantir plus de bande passante. Dans ce protocole de routage, les nœuds du cœur du réseau auront plus de trafics à gérer, en plus des messages de contrôle (pour la découverte et la maintenance des routes). En outre, en cas de forte mobilité, la convergence de l'algorithme est difficile à atteindre.

Cependant malgré ses défauts, la connaissance partielle de la topologie du réseau de cœur permet de réduire énormément les *overhead* dus aux inondations du réseau, principal obstacle à la mise en place du routage avec *QoS* car très coûteux.

2.4.4 Les protocoles de routage géographique

2.4.4.1 Le protocole de routage *LAR*

Le protocole *LAR* [28] (*Location-Aided Routing protocol*) est un protocole réactif. La nouveauté introduite par les protocoles assistés par un système de localisation tel que *GPS (Global Positioning System)*, est l'utilisation d'une estimation de la position afin d'accroitre l'efficacité de la procédure de découverte. Il limite la surcharge du réseau induite par la localisation d'un nœud, en définissant soit une zone de recherche, soit les coordonnées du nœud. Pour localiser un nœud, c'est-à-dire, identifier ses coordonnées géographiques ou la zone dans laquelle il se trouve, le protocole *LAR* [28] utilise les dernières coordonnées, vitesses et directions connues et suivies pour le nœud à localiser. Lors de la recherche du nœud, deux techniques sont proposées *LAR1* et *LAR2*.

Dans la première technique (Figure 2.17), le nœud source définie une région circulaire dans laquelle la destination peut être localisée.

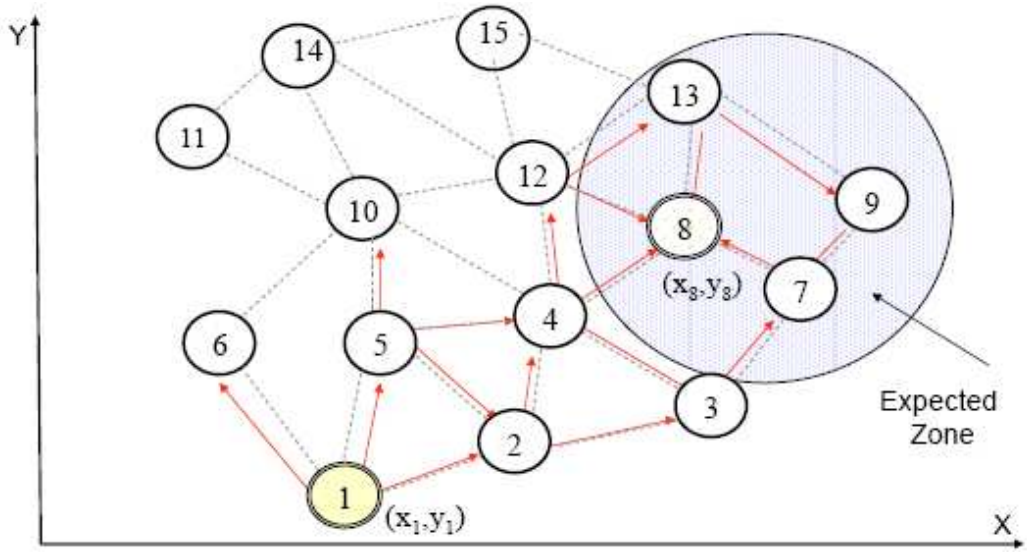


Figure 2.17— Le principe de recherche de route dans *LAR1* [28]

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.
- La vitesse moyenne du mouvement de la destination.

Le plus petit rectangle couvrant la région circulaire et le nœud source, est appelé la zone de requête. L'information calculée, est rattachée au paquet de requête de route. Cela est fait uniquement par le nœud source, et les nœuds qui appartiennent à la zone de requête.

Dans la deuxième technique, elle ne repose pas sur la détermination de zone de recherche, mais sur le choix du plus court chemin en termes de distance entre la source et la destination. Chaque nœud envoie l'information de recherche de route vers son voisin le plus proche, en distance, de la destination finale (Figure 2.18, de la page suivante).

Ainsi, l'inondation du réseau est contrôlée par les nœuds durant l'étape de localisation car ils ne transmettent une requête que si la requête se dirige dans la bonne direction, c'est-à-dire, vers la zone de recherche.

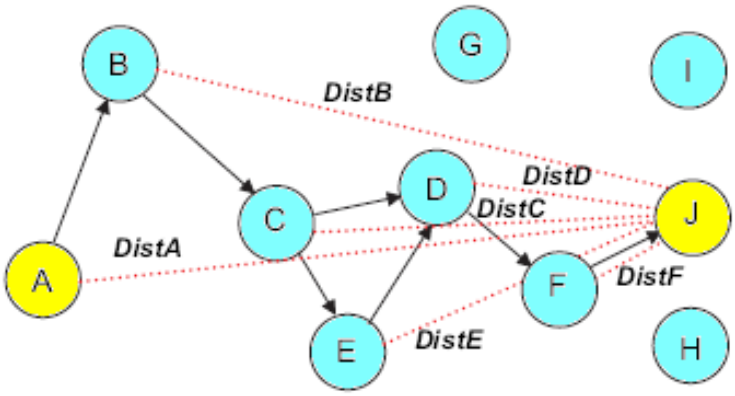


Figure 2.18 — Le principe de recherche de route dans *LAR2* [28]

2.4.4.2 Le protocole de routage *DREAM*

Le protocole *DREAM* [29] (*Distance Routing Effect Algorithm for Mobility*) est conceptuellement proche des protocoles de routage proactifs. Régulièrement, il maintient à jour les informations de localisations des unités mobiles, dans des tables de positions qui peuvent être assimilées aux tables de routage des nœuds utilisées par un protocole proactif. La fréquence de mise à jour des informations pour un mobile dépend de deux paramètres :

- d'une part de son éloignement par rapport au nœud mettant à jour sa table,
- et d'autre part de sa mobilité, c'est-à-dire s'il se déplace plus ou moins rapidement.

En effet, plus les nœuds sont distants, moins leur déplacement relatif est important, mais la route est moins stable car il y a beaucoup de sauts entre eux. De même, plus un nœud est mobile, plus les informations le concernant doivent être actualisées.

La prise en compte de ces deux paramètres permet de réduire le trafic de contrôle par rapport à un protocole réactif pur. Les paquets de mise à jour ont donc des fréquences d'émission variables suivant la localisation relative d'un nœud et sa mobilité. Pour déterminer la route à suivre à partir d'un nœud mobile *A* pour atteindre un nœud mobile *J*, on utilise la table de position.

Dans la table du nœud mobile *A* de la figure 2.19, il possède la localisation de *J* à l'instant t_1 ainsi que sa vitesse v_J, *A* peut donc en déduire une position probable de *J* à l'instant présent t_2. En effet, *J* se trouve probablement dans un cercle de rayon v_J (t_2 — t_1). Le mobile *A* détermine ensuite un cône dont il est le sommet et donc la zone probable de présence de *J* est la base. Le mobile *A* envoie ensuite les informations à transmettre à tous les nœuds inclus dans le cône qui retransmettent jusqu'à atteindre *J*. Lors de chaque retransmission, la route est mise à jour dans le paquet afin que *J* connaisse le chemin de retour pour les messages de retour comme les acquittements de trame. Si plusieurs chemins sont possibles, *J* choisit par exemple celui comportant le moins de sauts.

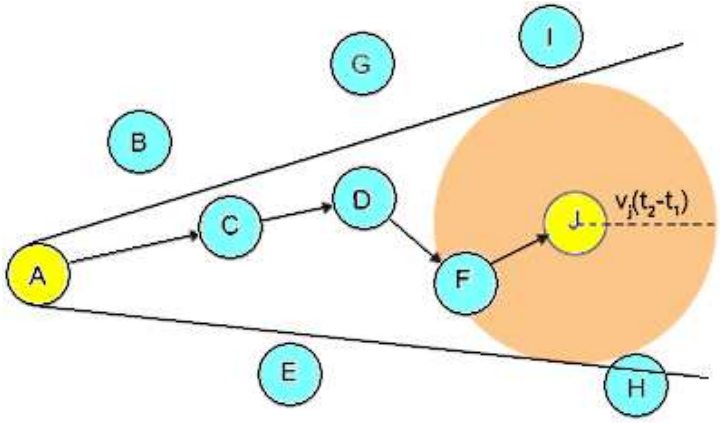


Figure 2.19 — Le principe de recherche de route dans *DREAM* [29]

Le protocole *DREAM* [29] doit malgré tout avoir un second protocole de secours dans le cas où il n'y a pas de nœuds présents dans le cône, mais où il existe une route moins directe à l'extérieur du cône, dans ce cas on utilise la diffusion (*flooding*) sur l'ensemble du réseau. Dans le cas de *DREAM*, les paquets de recherche de route, contiennent aussi des données.

2.4.4.3 Le protocole de routage *ZHLS*

Le protocole *ZHLS* [27] (*Zone-Based Hierarchical Link State Routing*) est un protocole de routage à état de liens hiérarchique basé sur la décomposition du réseau, de façon statique, en un ensemble de zones disjointes à partir d'une carte.

Contrairement aux autres protocoles hiérarchiques, le groupe ou "*cluster*" dans ce protocole ne possède pas de représentant ou de chef de zone. Cette décomposition fait ressortir deux niveaux de hiérarchies: le niveau nœud, et le niveau zone.

- Le premier niveau, donne la façon dans laquelle les nœuds, d'une zone donnée, sont connectés physiquement entre eux.
- Le deuxième niveau est basé sur le schéma de l'interconnexion des différentes zones.

De cette façon, chaque nœud connaît la topologie de sa zone (approche proactive) et les nœuds jouant le rôle de passerelle (*gateway*) entre les zones. Ces nœuds passerelles sont connus par les autres nœuds puisqu'ils diffusent sur l'ensemble du réseau les informations concernant les zones qu'ils connectent. Ainsi, chaque nœud du réseau connaît le découpage du réseau en zones et les nœuds donnant accès à ces zones (Figure 2.20).

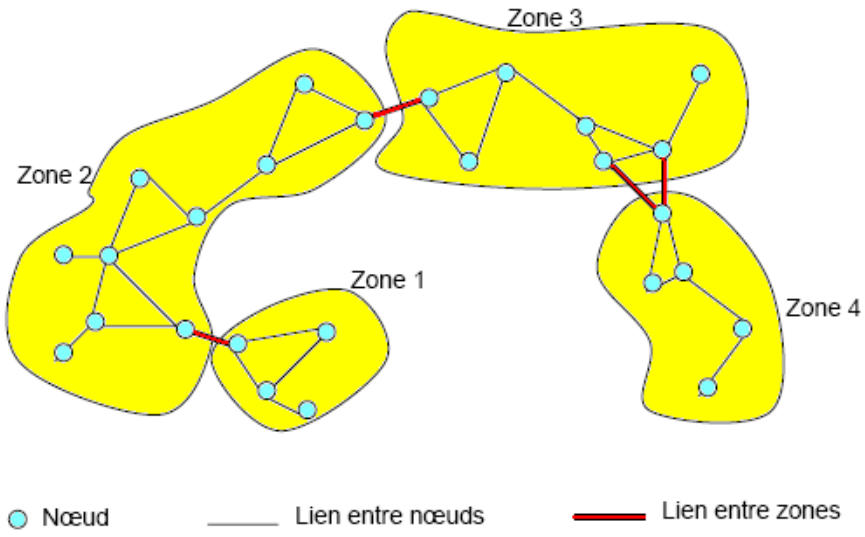


Figure 2.20 — La décomposition du réseau en zones dans *ZHLS* [27]

Dans ce protocole, les paquets qui contiennent les états des liens ou les *LSPs* (*Link State Packet*), peuvent être divisées en deux classes : les *LSPs* orientés nœuds, et les *LSPs* orienté zones. Pour un nœud donné, un paquet *LSP* orienté nœud, contient l'information d'un nœud voisin, tandis qu'un paquet *LSP* orienté zone, contient l'information de la zone et il est envoyé vers n'importe quel nœud du réseau global. Par conséquent, l'acheminement des données, se fait de deux façons : le routage inter zone, et le routage intra zone. Pour une destination donnée, les données sont envoyées entres les zones en utilisant les identificateurs des zones, jusqu'à ce que les données atteignent la zone de destination.

Lorsqu'un nœud désire connaître la localisation d'un autre nœud, il vérifie que le nœud ne se trouve pas dans sa zone. Si ce n'est pas le cas, il envoie une requête de localisation vers toutes les zones présentes dans le réseau en utilisant les informations correspondantes aux nœuds passerelles pour atteindre ces zones. Par la suite, les paquets de données circulent à l'intérieur de la zone destinée, en utilisant l'identificateur du nœud destination. L'adresse *<ID zone, ID nœud>*, est suffisante pour atteindre n'importe quel nœud destination même si le réseau change de topologie.

2.5 Conclusion

Dans ce chapitre nous avons étudié les stratégies de routage et les différents modes de communication dans les réseaux à média sans fil. Ensuite, nous avons exposé les quatre classes de protocoles des réseaux Ad-Hoc avec leurs caractéristiques spécifiques. Enfin, nous avons présenté, par classes de protocoles, des exemples connus de protocoles de routage des réseaux Ad-Hoc tout en explicitant leur principe de fonctionnement.

Dans le chapitre suivant, nous décrirons la notion de qualité de service et en particulier dans les réseaux Ad-Hoc et nous présenterons les différentes catégories de solutions de qualité de service avec quelques exemples de ces protocoles.

Chapitre 3

Qualité de Service dans les Réseaux Ad-Hoc

3.1 Introduction

Le groupe *MANET* de *l'IETF* (*Internet Engineering Task Force*) a proposé plusieurs protocoles de routage pour les réseaux mobiles Ad-hoc. Ceux-ci fonctionnent en mode *best effort* c'est-à-dire : au mieux. Cependant, ils ne permettent pas de garantir une qualité de service.

Dans les réseaux de télécommunication, l'objectif de la qualité de service est d'atteindre un meilleur comportement de la communication, pour que le contenu de cette dernière soit correctement acheminé, et les ressources du réseau utilisées d'une façon optimale [10]. La recherche sur la qualité de service dans les réseaux mobiles Ad-Hoc touche à plusieurs domaines ; les protocoles de signalisation, le routage avec *QoS*, la différentiation au niveau de la couche *MAC* (*Medium Access Control*) et les modèles de *QoS*.

Dans ce chapitre on s'intéresse aux solutions de *QoS* dans *MANET*. On a commencé par présenter les principales notions et les différentes métriques de qualité de service dans les réseaux Ad-Hoc. Par la suite, on a explicité les principales catégories de solutions de routage avec *QoS* dans les *MANETs.*

3.2 Définition de la Qualité de service

La recommandation *E800* du *CCITT (Consultative Committee for International Telegraph and Telephone)* définit la qualité de service *QoS* (*Quality of Service*) ou QdS (Qualité de Service) comme : « l'effet général de la performance d'un service qui détermine le degré de satisfaction d'un utilisateur du service ». Cette définition reflète la perception de la qualité de service du point de vue d'un utilisateur [35].

Dans le *RFC 2386* [53], la *QoS* est définit comme un ensemble de besoins à assurer par le réseau pour le transport d'un trafic d'une source à une destination. Ces besoins peuvent être traduits en un ensemble d'attributs pré-spécifiés et mesurables en terme de :

- Délai de bout en bout,
- Variation du délai (gigue),
- Bande passante,
- Pertes de paquets.

3.3 Les métriques de la qualité de service

Les principaux aspects connus de la qualité de service sont : la bande passante, le délai de bout en bout, la gigue (variation du délai), et les pertes de paquets ou taux d'erreurs [10].

Les métriques de *QoS* peuvent être additives, concaves ou multiplicatives :

- la bande passante est une métrique concave,
- le délai et la gigue sont des métriques additives,
- et la disponibilité d'un lien, basée sur des critères comme la probabilité de perte du lien quant à elle est une métrique multiplicative [10].

Une métrique additive [10] A_m est définie comme $\sum L_i(m)$ où $L_i(m)$ est la valeur de la métrique m sur le lien L_i et $L_i \in P$. P est le chemin de la source à la destination.

Une métrique concave [10] définit la valeur minimale sur un chemin P et représenté comme suit : $C_m = min\ (L_i\ (m)),\ L_i\ (m) \in P$. Pour trouver une route qui satisfait une métrique concave, les ressources disponibles dans chaque lien doivent être au moins égales à la valeur désirée de la métrique.

Une métrique multiplicative [10] représente le produit des valeurs des métriques de *QoS*, elle est définie comme le produit des $L_i(m)$ avec i allant de 1 à h, $Li\ (m) \in P$. h représente la longueur du chemin P..

3.3.1 La bande passante

La bande passante représente la source de transmission qu'occupe ou reçoit un flot. La gestion de la bande passante est un élément important pour la garantie de la qualité de service [10].

3.3.2 Délai de bout en bout

Le terme « délai » englobe en réalité trois aspects temporels différents [10]:

a) **le délai de propagation**, est fonction de la distance physique qui sépare la source de la destination (plus la distance est grande, plus le délai est important).

b) **le délai d'attente et de traitement des paquets**, à l'intérieur des files d'attente, déterminé par la charge du réseau, ainsi que les politiques de traitement de l'information dans les nœuds pour obtenir une fluidité maximale de l'écoulement de l'information.

c) **le délai de transmission** dépendant de la taille des flots. Ce paramètre est aussi étroitement lié à l'utilisation du réseau et au partage de la bande passante disponible.

Garantir le délai, implique la nécessité de mettre en œuvre des mécanismes permettant de gérer au mieux l'acheminement de l'information vers la destination en un temps minimal [10], tenant compte des trois natures de délais précédemment cités.

3.3.3 La gigue

La gigue correspond à la variation du délai de transmission de bout en bout entre les différents paquets d'un flot à travers un réseau. La gigue est due principalement aux délais de traitement variables dans les nœuds du réseau. Ce paramètre nuit automatiquement à la qualité de service demandée [10].

3.3.4 La perte de paquets

Elle se produit lorsqu'il y a des erreurs d'intégrité sur les données. La perte de paquets se produit principalement lorsque l'intensité du trafic sur les liens de sorties devient supérieure à leur capacité d'écoulement [10]. Lorsque les nœuds évoluent dans un milieu à fading fort (affaiblissement important) ou à grande mobilité, la probabilité que le signal émis n'arrive pas est fonction de la distance qu'il parcourt. Les données transportées sont donc sujette à de nombreuses perturbations et le taux d'erreur des paquets y est important.

La solution proposée dans ce mémoire se base essentiellement sur cette nouvelle métrique, le degré de stabilité d'un lien, pour la construction de ses routes. Nous verrons dans le chapitre suivant que la probabilité de rupture des liens est fonction de la distance qui sépare les nœuds communiquant et qu'il convient donc de la minimiser (la distance).

3.4 Solutions de QoS pour les réseaux Ad-Hoc

Les solutions de *QoS* pour les réseaux mobiles Ad-Hoc peuvent être classifiées en quatre catégories (figure 3.1) :

- Les mécanismes de réservation (protocoles de signalisation) définissent un ensemble de messages de contrôle, destinés par exemple à provoquer la réservation de ressources dans les routeurs (par exemple, *RSVP* [56]).
- Les protocoles de routage avec qualité de service sont chargés de la recherche de routes répondant à certains critères.
- Différenciation des services au niveau de la couche *MAC* [59] (*Medium Access Control*), fournissent un ensemble d'outils permettant de mettre en œuvre certaines règles de qualité de service.
- Les modèles de qualité de service regroupent les définitions d'architectures destinées à assurer une certaine qualité de service (par exemple *intserv* et *diffserv*)

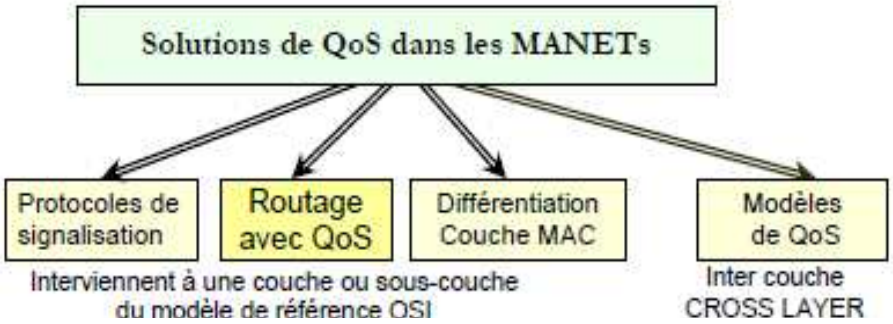


Figure 3.1 - Solutions de *QoS* pour les réseaux Ad-Hoc

3.4.1 Protocoles de signalisation

Dans le premier modèle, les protocoles de signalisation reposent sur un moyen de propager des informations de contrôle à travers un réseau. Les informations transmises peuvent être de différentes natures. Il peut s'agir d'informations topologiques, de demandes de recherche de routes satisfaisant certaines contraintes ou encore de rapports sur l'état du réseau et la disponibilité des ressources.

Concevoir un protocole de signalisation consiste à définir les données à échanger afin de réaliser une tache particulière ainsi que la manière de les échanger. La signalisation permet de réserver et de mettre à jour les ressources, d'initialiser et d'arrêter le trafic ainsi que de renégocier le profil du trafic. Elle peut s'effectuer à l'intérieur des paquets de données (signalisation *in-band*) ou grâce à des paquets explicites de contrôle [10] (signalisation *out-band*).

Par exemple, le protocole *INSIGNIA* [36] inclus la signalisation dans les entêtes des paquets de données (signalisation in-band) pour la réservation de la bande passante dans les réseaux Ad-Hoc. *INSIGNIA* offre des garanties sur la base d'une granularité par flot aux applications adaptatives capables de modifier leur comportement en fonction de la quantité de bande passante qui leur est allouée.

Chaque application spécifie deux niveaux de qualité de service. Le niveau de base permet de spécifier la bande passante minimale nécessaire au trafic et le niveau amélioré, le débit optimal à atteindre lorsque les ressources sont disponibles. Ce protocole a été conçu pour réagir rapidement aux changements de topologie. *INSIGNIA* [36] n'est pas lié à un protocole de routage particulier.

3.4.2 Protocoles de routage avec *QoS*

La deuxième architecture, le routage avec qualité de service (*QoS*), a été définie dans le *RFC 2386* [53] comme étant : « Un mécanisme de routage dans lequel les chemins sont déterminés en fonction des connaissances sur la disponibilité des ressources du réseau ainsi que l'exigence de qualité de service de flux ». En d'autres termes l'auteur de [42] le définit comme étant : « un processus d'établissement et de maintenance de routes optimales satisfaisant un certain critère sur la qualité de la transmission de données ». Les algorithmes de routage traditionnels ont été proposés pour router les données sans tenir compte des contraintes spécifiques ou à des demandes des utilisateurs. Ainsi, ils sont inadaptés aux applications qui nécessitent le support de la qualité de service.

Le routage avec *QoS* est un élément clé pour réaliser une architecture de *QoS* pour les réseaux mobiles Ad-Hoc. Le protocole de routage *QoS* peut informer une source sur la bande passante, le délai, la gigue ou la et la disponibilité (en termes de *QoS*) de la destination. Cette connaissance va permettre l'établissement de connections avec qualité de service.

Quelques exemples de protocoles de routage avec qualité de service sont présentés dans le tableau 3.1 de la page suivante, décrivant les différentes caractéristiques et techniques pour fournir la qualité de service au niveau de la couche réseau.

Chacun de ces protocoles aborde les problèmes de l'estimation de la bande passante et du délai, la découverte de routes avec *QoS*, la réservation des ressources, et l'approche utilisée dans la maintenance des routes, les protocoles présentés dans ce tableau sont exposés dans [43].

Protocole de routage	Architecture	Métrique de *QoS*	Estimation de délai bande passante	découverte de routes	Réservation de ressources	Prise en compte des cassures de liens	Routes redondantes
CEDAR	Hiérarchique	Bande passante	Non	Proactive /Réactive	Oui	Non	Non
Ticket-based	Plat	Bande passante /délai	Non	Réactive	Oui	Non	Oui
OLSR-based	Hiérarchique	Bande passante	Oui	Proactive	Non	Non	Non
AQOR	Plat	Bande passante /délai	Oui	Réactive	Oui	Non	Non
ADQR	Plat	Bande passante	Non	Réactive	Oui	Oui	Oui
TDR	Plat	Bande passante	Non	Réactive	Oui	Oui	Non
BEQR	Plat	Bande passante	Oui	Réactive	Non	Non	Non

Tableau 3.1 - Exemples de protocoles de routage avec *QoS* [43].

3.4.3 Protocoles de différentiation de services (couche *MAC*)

Dans le troisième modèle, les protocoles de différenciation des services au niveau de la couche *MAC* (*Medium Access Control*), s'intéresse aux problèmes d'accès au médium ainsi qu'à la fiabilité des communications.

Récemment, des schémas de différenciation de service au niveau *MAC* ont été proposés. Le principe est de doter le protocole *IEEE 802.11* d'un mécanisme de priorités entre les trames afin de concevoir des mécanismes de différenciation de services efficaces. Pour ce faire, les auteurs de [59] proposent d'adapter certains paramètres de la fonction de coordination distribuée (*DCF*) du protocole *802.11* selon les priorités des paquets.

3.4.4 Modèles de *QoS* (*IntServ* et *DiffServ*)

Dans le quatrième modèle de qualité de service, on définit quels types de services peuvent être fournis dans un réseau pour cela certains mécanismes sont utilisés afin d'offrir ces services. Les modèles les plus connus dans le monde filaire sont: *IntServ* (*Integrated services*) et *DiffServ* (*Differentiated Services*).

Le modèle *IntServ* [57] propose une architecture à base de réservation de ressources pour chaque flot. Cette dernière s'effectue au moyen du protocole *RSVP* [56] (*Resource Reservation Setup Protocol*), Cependant, *IntServ* n'est pas propice à des réseaux de grande envergure.

Le modèle *DiffServ* [58] est fondé sur le concept de la gestion du trafic par classe, sur des méthodes de conditionnement du trafic à l'entrée du réseau et sur le marquage de celui-ci en fonction de son appartenance à une classe. Trois principaux services de *DiffServ* sont mis en œuvre: le service au-mieux « *Best-Effort* » comme le fait Internet actuellement, le service garanti «*Assured Forwarding*» pour des classes à priorité plus grande et enfin le service assuré «*Expedited Forwarding* » pour les classes les plus prioritaires.

Le modèle de qualité de service proposé par *Xiao* et *al* [41], *FQMM* « *A flexible quality of service model for mobile ad-hoc networks* », définit une architecture hybride adaptée à des réseaux Ad-Hoc de taille moyenne. Les concepteurs du modèle *FQMM* prennent en compte le fait que les réseaux Ad-Hoc pourraient, à terme être connectés à des réseaux filaires de type Internet. Il apparaît dès lors nécessaire d'offrir un mécanisme de qualité de service suffisamment proche des protocoles filaires afin de s'interfacer avec ces derniers. Le modèle *FQMM* se situe entre les deux modèles *IntServ* [57] et *DiffServ* [58], il définit plusieurs classes de service dont la plus haute permet à chaque flux de spécifier les contraintes qui lui sont propres. *FQMM* définit trois types de nœuds:

- les nœuds d'entrée (émetteurs),
- les nœuds intermédiaires, et
- les nœuds de sortie (récepteurs).

Compte tenu du fait que dans un réseau Ad-Hoc, chaque nœud assure la fonction de routeur, chaque mobile joue différents rôles pour différents flux. Le conditionnement du trafic (lissage, marquage, etc.) est à la charge des émetteurs. *FQMM* [41] requiert l'utilisation d'un protocole de routage capable d'offrir une certaine *QoS*, c'est à dire capable de rechercher des routes satisfaisant certaines contraintes [42].

3.5 Conclusion

L'introduction de la *QoS* dans les *MANETs* est devenue une nécessité pour certaines applications. Pour atteindre cet objectif, plusieurs solutions ont été proposées touchant une ou plusieurs couches du réseau. Dans notre étude on s'intéressera plus particulièrement à la couche réseau, en essayant d'établir un routage efficace en termes de certains critères de la *QoS*.

Dans le prochain chapitre, nous proposerons une solution de qualité de service, au niveau de la couche réseau, basée particulièrement sur le routage suivant les liens stables.

Chapitre 4
Routage Suivant les Liens Stables

4.1 Introduction

Deux approches sont possibles pour la conception d'un protocole de routage avec qualité de service (*QoS*) :

- Approche révolutionnaire qui consiste à concevoir un nouveau protocole avec de nouvelles fonctionnalités.
- Approche évolutionnaire qui consiste à faire des extensions des protocoles best effort existants ou d'apporter des améliorations au protocole de routage avec *QoS* en ajoutant, par exemple, d'autres métriques.

Dans ce mémoire nous avons choisi cette dernière car il est plus efficace et facile et encore moins couteux d'améliorer un travail existant que de refaire un nouveau travail. En plus, l'extension d'un protocole existant est plus compatible avec la version originale, ce qui permet l'utilisation des deux à la fois et facilite l'interconnexion des deux plates formes fonctionnant avec l'ancienne et la nouvelle version du protocole.

Des études comparatives montrent que certains protocoles sont plus performants que d'autres selon les caractéristiques du réseau. Ces études ont montré que le protocole *AODV* [23] (*Ad-hoc On-Demand Distance Vector*) semble convenir à des réseaux à forte mobilité et semble performant dans les réseaux de faible densité [16]. Nous avons donc choisi de faire une extension du protocole *AODV Standard*, étudié en détail dans la section 2.4.2.2.

Dans les réseaux mobiles Ad-Hoc, les stratégies de routage multi-sauts selon le nombre de sauts optimal ne sont pas satisfaisantes pour améliorer la qualité de service du réseau et pour prémunir contre les ruptures de liens. Ce qui nous a conduit à l'utilisation d'une nouvelle métrique : il s'agit de la puissance du signal reçu permettant ainsi la sélection d'itinéraire à forte stabilité entre les paires de nœuds et une plus grande durée de vie du chemin de bout en bout.

Comme nous l'avons vu, la stratégie de routage du protocole *AODV Standard* [23], ne permet pas de garantir des critères de qualité de service et de prémunir contre les ruptures de liens. C'est pourquoi il semble important de faire une extension de ce protocole afin d'assurer une certaine qualité de service.

Ce chapitre traite une solution qui permet d'étendre le protocole *AODV* [23] pour garantir la qualité de service en termes stabilité des itinéraires. On a commencé par expliciter la solution à implémenter pour dégager le comportement du protocole *AODV* suite à l'introduction d'un contrôle d'admission des nouvelles connexions basé sur la puissance du signal reçu par le nœud. Ainsi l'estimation de la stabilité des itinéraires et les extensions nécessaires au protocole *AODV* pour garantir la qualité de service sont précisés et sont suivis par une description du nouveau fonctionnement du protocole.

4.2 Le routage AODV avec qualité de service

L'introduction de la qualité de service dans *AODV Standard* [23] repose sur l'ajout d'un champ dans les paquets de contrôle *RREQ*, *RREP*. Ce champ peut être associé au paramètre de non stabilité ou de la probabilité de rupture du lien. À la réception d'un message *RREQ*, chaque mobile vérifie qu'il est au dessus du seuil de rupture fixé, avant de retransmettre le message. Le protocole de routage *AODV* avec *QoS* a pour objectif de :

- Améliorer la *QoS* dans les réseaux Ad-Hoc.
- Introduire une métrique plus appropriée que le nombre de sauts.
- Faire face aux changements fréquents de la topologie due à la mobilité des nœuds.

Dans ce qui suit, on présente une proposition qui intègre de la qualité de service en termes de stabilité des itinéraires dans le protocole *AODV*.

4.2.1 Facteurs d'atténuation du signal

La puissance du signal reçu par un récepteur diffère de celle transmise à l'origine. Cet affaiblissement est du essentiellement à trois phénomènes [47] :

- **Le *Path Loss***: caractérise l'affaiblissement que subit un signal électromagnétique lorsqu'il parcourt une distance. Plus le récepteur s'éloigne de l'émetteur et plus l'affaiblissement sera important.
- **Le *Shadowing***: prend en compte les obstacles rencontrés par le signal sur son trajet. Cette métrique enrichit le *Path Loss* d'un affaiblissement probabiliste fonction du milieu de propagation et du type d'obstacle pouvant être rencontré.

- **Le *Fast fading* ou *multipath fading***: il est lié au fait que l'onde reçue est une superposition de plusieurs copies du signal aux propriétés différentes (amplitude, phase…) ayant emprunté des chemins différents.

Les effets de ces perturbations s'additionnent ou se multiplient selon qu'on les considère à une échelle logarithmique ou linéaire (figure 4.1), auquel vient s'ajouter une autre perturbation supplémentaire engendrée par le bruit (signaux parasites originaires du système électronique interne au récepteur ou de facteurs externes : intermodulation, diaphonie, …).

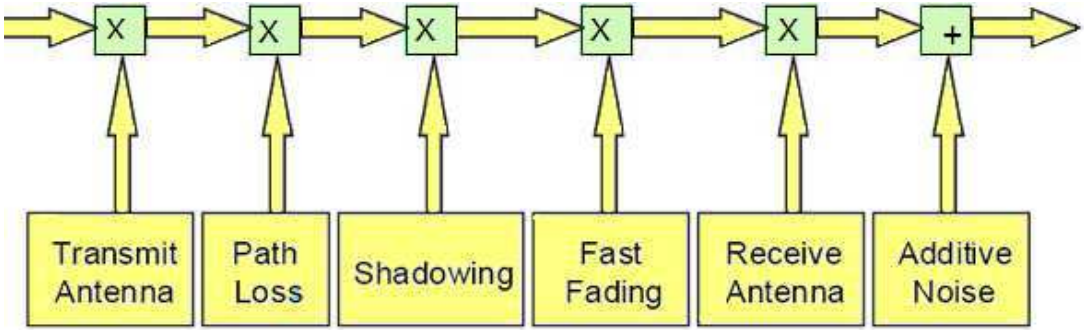


Figure 4.1 - Modélisation du canal de communication

4.2.2 Estimation du *Path Loss* et de la Puissance reçue

Pour trouver la valeur de la puissance reçue dans une transmission sans fil, on a besoin de calculer la *Path Loss*. Ce dernier dépend, comme on a déjà vu, de la distance entre l'émetteur et le récepteur [47].

La longueur d'onde Lambda (*L*) de la transmission du paquet est obtenue en divisant la vitesse de la lumière (*C*) par la fréquence utilisée (*2.4 GHZ*).

La vitesse de la lumière (*C*) = 3,0 10^{+8} mètre/seconde.

La longueur d'onde Lambda (*L*) est donnée par l'équation (4.1) suivante:

$$L(\text{mètre}) = \frac{C\ (mètre/\text{sec})}{Fréquence\ (Hz)} \qquad (4.1)$$

Le *Path Loss* (*PL*), dans le modèle en espace libre [50], est donnée par l'équation (4.2) suivante:

$$Path\ Loss\ (PL) = \frac{16\ \pi^2\ d^{\,i}}{L^2} \qquad (4.2)$$

tel que $i = 2$ pour le modèle en espace libre (*Free Space*), et $i = 3$ ou $i = 4$ pour un environnement normal.

Si la puissance de transmission (*Tx_power*) et le *Path Loss* (*PL*) sont connues, alors la puissance de réception (*Rx_power*) est donnée par l'équation (4.3) suivante:

$$Rx\_power = \frac{Tx\_power}{PL} \qquad (4.3)$$

Par exemple, dans un réseau sans fil de *2* nœuds, situés à *300* mètres l'un de l'autre et émettant avec une puissance de *0.001W*, le *Path Loss* et la puissance reçue sont calculés comme suit:

$$PL = \frac{16\Pi^2 (300)^2}{(0{,}124947938)^2}$$

$$= \mathbf{9{,}10340890367493E}+08 \qquad \text{(modèle en espace libre)}$$

$$Rx\_power = \frac{0{,}001}{9{,}10340890367493E+08}.$$

$$= \mathbf{1{,}09848959942502E\text{-}12\ W}$$

La figure 4.2 est un exemple pris du tableau 4.1 de la page suivante, montrant qu'un nœud émetteur a besoin d'une puissance minimale de transmission égale à *0,00025 Watt* pour atteindre un nœud récepteur distant à *150* mètres. Si on dépasse ces distances, le seuil ou la sensibilité de réception ne sera pas atteinte (inférieur à *-90 dB = 1,00000E-12 W*) et la transmission ne pourra pas s'effectuer.

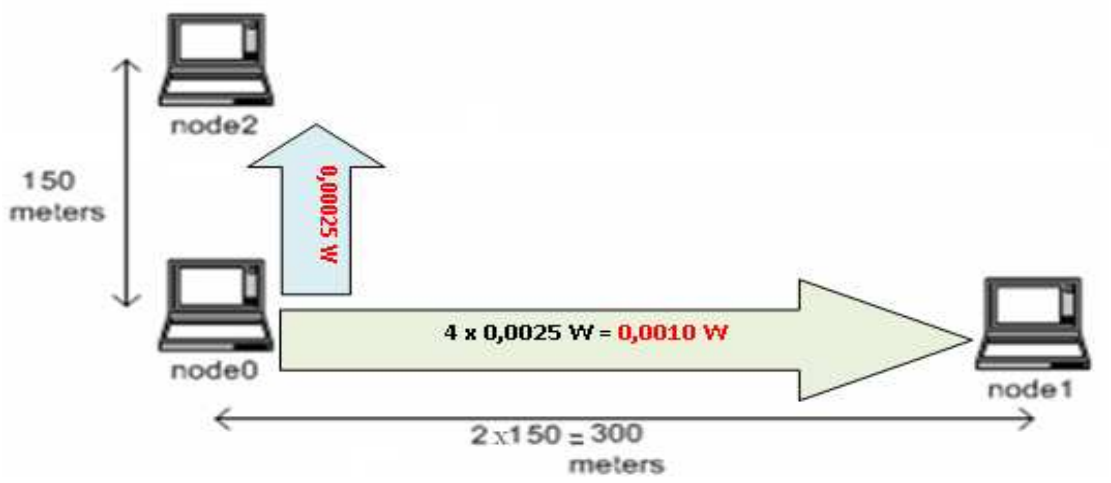


Figure 4.2 – Puissance de transmission pour différente distances

Cependant il est nécessaire de fournir une puissance minimale de transmission égale à *0,0001 Watt* pour atteindre un nœud récepteur distant à *300 mètres*. Donc on a besoin de quadrupler la puissance de transmission pour pouvoir doubler la distance.

$d_{réf}$ (mètres)	***Rx_power*** *(Tx_power=0,001 W)*	***Rx_power 2 (Watt)*** *(Tx_power=0,00025 W)*
50	3,95456255793006E-11	9,88640639482514E-12
60	2,74622399856254E-11	6,86555999640635E-12
70	2,01763395812758E-11	5,04408489531895E-12
80	1,54475099919143E-11	3,86187749797857E-12
90	1,22054399936113E-11	3,05135999840282E-12
100	9,88640639482514E-12	2,47160159870629E-12
150	4,39395839770006E-12	**1,09848959942502E-12**
200	2,47160159870629E-12	**Rupture du lien**
250	1,58182502317202E-12	**Rupture du lien**
300	**1,09848959942502E-12 ≈ -90dB**	**Rupture du lien**

Tableau 4.1 - Puissances de réception utilisant un *Tx_power* de *0,001W* et *0,0025W* avec une sensibilité de réception (*threshold*) d'au moins *-90 dB*.

Les puissances des signaux sont quantifiées en *Watts* ou en *Décibels* selon qu'on les considère une échelle logarithmique (*dB*) ou linéaire (*Watt*). Ainsi, on peut convertir le seuil de réception (*threshold*) du Décibel vers le Watt, en utilisant la formule (4.4) suivante:

$$\textit{Seuil de réception (watts)} = \frac{10^{threshold(dB)/10}}{10^3} \qquad (4.4)$$

Par exemple, pour convertir le seuil de réception de *-90 dB* en *Watts*, l'équation (4.4) va donner les résultats suivants :

$$\textit{Seuil de réception (-90 dB)} = 10^{-90/10} . 10^{-3}\ W$$

$$= 10^{-9}.10^{-3}\ W = 10^{-12}\ W$$

$$= \mathbf{1,0E\text{-}12\ W}$$

4.2.3 Détermination de la probabilité de rupture (*PR*)

La probabilité de rupture d'un lien *PR(i,j)* va être calculé en utilisant la puissance du signal reçu de *j (émetteur)* qui est récupérée à partir de la couche *MAC*, et de la puissance maximum de réception du nœud *i* (*récepteur*) c'est-à-dire la puissance de transmission utilisée. Ainsi la probabilité de rupture d'un lien *(i, j)* est déterminée par la formule (4.5) suivante :

$$\textit{La probabilité de rupture } PR(ij) = \frac{\textit{Puissance Max } P_{MAX}(i) - \textit{Puissance Reçue } P_r(j)}{\textit{Puissance Max } (i)} \qquad (4.5)$$

On sait que la *Puissance Max* $P_{MAX}(i)$ reçue par un nœud *i* ne peut pas dépassée sa puissance de transmission (P_t) qui est toujours fixe. La formule (4.5) peut alors être écrite sous la forme (4.6) suivante :

$$\textit{La probabilité de rupture } PR(ij) = \frac{\textit{Puissance transmission } P_t - \textit{Puissance Reçue } P_r(j)}{\textit{Puissance transmission } P_t} \qquad (4.6)$$

Ainsi (lien très stable) la probabilité de rupture d'un lien $PR_i(ij)$ tend vers zéro si la puissance reçue du signal $P_r(j)$ atteint sa valeur maximum c.à.d. la valeur de la puissance de transmission ($P_r(j) = P_t = P_{Max}(i)$), par contre (lien cassé) elle tend vers un si la puissance reçue du signal $P_r(j)$ est en dessous de son minimum ($P_r(j)$ < Seuil de sensibilité fixé).

4.2.4 Calcule de la stabilité d'un itinéraire

Les protocoles multi-sauts construisent leurs routes de manière à minimiser le nombre de sauts intermédiaires entre la source et la destination. Affin d'y parvenir, ils choisissent les nœuds les plus distants les un des autres. Lorsque les nœuds évoluent dans un milieu à forte atténuation (*Path Loss, Shadowing*, …) la probabilité que le signal émis rencontre des obstacles est fonction de la distance qu'il parcourt. Les données transportées sont donc sujette à de nombreuses perturbations et le taux d'erreurs des paquets y est important.

Notre solution est donc basée sur cette nouvelle métrique, la probabilité de rupture du lien *PR(ij*) au niveau d'un nœud *i* avec un nœud *j*, pour la construction de ses routes. Nous avons vu que cette probabilité est fonction de la distance qui sépare les nœuds communiquant et qu'il convient donc de la minimiser. C'est pour cela que nous avons d'abord calculé la probabilité de rupture des liens *PR(ij*) constituant la route, puis on en déduit la stabilité d'un itinéraire de bout en bout.

La stabilité d'un itinéraire $MP_{S,D}$ est une métrique multiplicative, autrement dit la stabilité totale de bout en bout entre une source « *S* » et une destination « *D* » est égale au produit des probabilités de rupture *PR(ij)* de tous les liens *(i,j)* constituants cette trajectoire.

$$MP_{S,D}(k) = \prod_{i\,,\,j \in [S,D]} PR(ij) \qquad (4.7)$$

Cette dernière expression nous indique bien que pour minimiser le $MP_{S,D}(k)$ de la $k^{ème}$ route il faut maintenir la probabilité de rupture *PR(ij)* de chaque lien la plus petite possible. Donc, notre objectif essentiel est de sélectionner un chemin dont le $MP_{S,D}(k)$ est le minimum $MP_{S,D}^{*}$ (itinéraire le plus stable) parmi tous les itinéraires trouvés durant le processus de découverte de routes.

$$MP_{S,D}^{*} = min\ \{\ MP_{S,D}(k)\ \} \qquad (4.8)$$

tel que $k = 1..n$, c'est le nombre d'itinéraires trouvés lors de l'opération de découverte de route initiée par le nœud source « *S* ».

Dans l'exemple de la figure 4.3 de la page suivante, le deuxième itinéraire (*AGFED*) est sélectionné par notre proposition car la valeur calculée de son $MP_{A,D}(2) = 1*PR(AG)*PR(GF)*PR(FE)*PR(ED) = 0,9999998660$ est inférieure à celle premier $MP_{A,D}(1) = 1*PR(AB)*PR(BC)*PR(CD) = 0,9999999703$. Donc, on estime que le chemin (*AGFED*) est plus stable que le second (*ABCD*) malgré que celui-ci soit plus long en nombre de sauts.

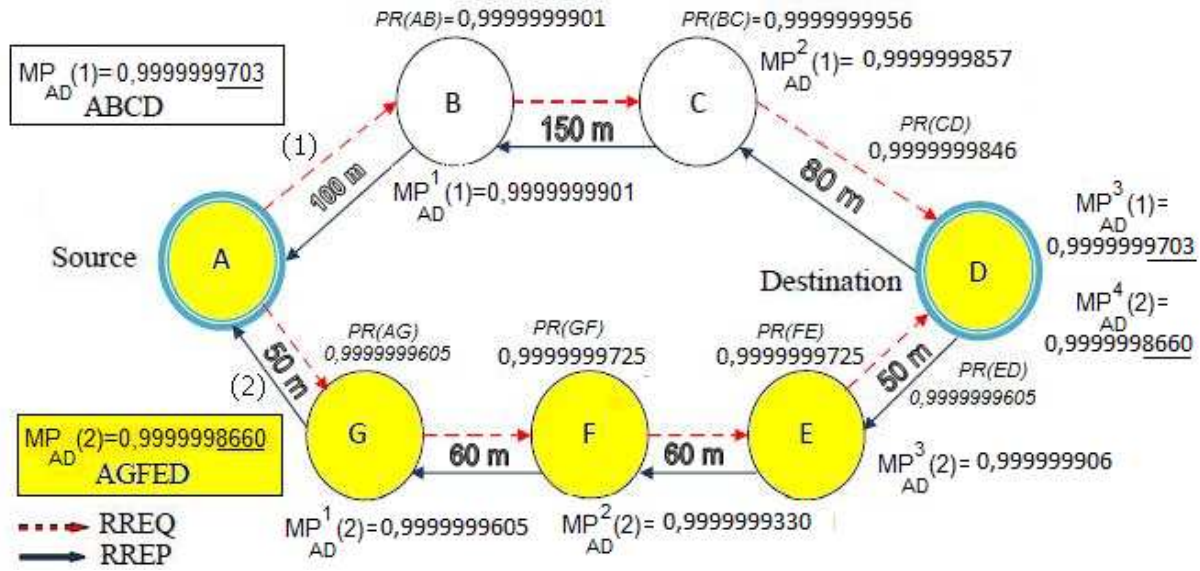


Figure 4.3 – Sélection du chemin le plus stable selon notre proposition ($MP_{AD}(k)$ le plus faible)

4.3 Intégration dan AODV :

Pour introduire la qualité de service dans *AODV* [23], un contrôle d'admission est nécessaire pour permettre la vérification des conditions de la *QoS* demandée. L'idée repose sur l'ajout d'un champ dans les paquets de contrôle. L'information que contient ce champ sera utilisée pour faciliter le contrôle d'admission effectué lors de l'établissement des routes. Selon le principe de fonctionnement adopté, nous avons modifié le format de deux paquets ; *RREQ* (le paquet de la requête de découverte de route) et *RREP* (le paquet de la requête de réponse de route).

Dans le contexte de cette étude, nous essayons dans ce qui suit de mettre en œuvre les modifications nécessaires pour l'utilisation du chemin le plus stable existant entre la source et la destination contenant les liens stables (en terme de puissance de signal reçu). Pour les extensions on s'inspire des travaux [49] faits par *MANet*.

Pour distinguer le protocole *AODV Standard* [23] de l'extension faite dans ce travail, nous avons appelé notre protocole: *AODV-SI* (pour *Ad-hoc On-demand Distance Vector with Stable Itinerary*), ou *AODV* avec Stabilité d'Itinéraire.

4.3.1 Extension de la *RREQ*

La requête de découverte de route est étendue pour inclure un nouveau champ (*MP-Q*), il spécifie le degré de stabilité de la trajectoire parcourue par les paquets *RREQ* (Tableau 4.2). La source initialise à un ce champs et diffuse la requête puis les nœuds intermédiaires calculent ce champ à partir de la formule (4.7). Ce champ sera mis à jour, au fur et à mesure, de la diffusion de la requête de recherche de routes de la source (le premier nœud) à la destination (le *Nième* nœud).

Type	***Flags***	***Reserved***	***Hop count***
RREQ- ID			
Destination IP Address			
Destination Sequence Number			
Originator IP Address			
Originator Sequence Number			
MP-Q (champ incluant la qualité de service)			

Tableau 4.2 - Format de message *RREQ* dans *AODV-SI*

- *Type (8 bits):* ce champ indique le type de paquet, dans ce cas il prend la valeur 1.

- *Flags ou Drapeaux (5 bits*): ce champ contient cinq flags *(J, R, G, D, U)* tel que ;

  - ***J** (Join flag) et **R** (Repair flag)* sont réservés pour le multicast ;
  - ***G** (Gratuitous RREP flag*) indique si un message *RREP* spécifique doit être envoyé à la destination dans le cas où un nœud intermédiaire possède un chemin à la destination.
  - ***D** (Destination only flag)* ce drapeau indique si seulement la destination qui doit répondre à la requête ou pas.
  - ***U** (Unknown sequence number*) indique le numéro de séquence de la destination est inconnu.

- *Réserved (11 bits*): initialisé à la valeur 0 et ignoré à la réception du message.

- *Hop Count (8 bits*): il contient le nombre de sauts parcourus par *RREQ*.

- *RRE- ID:* il identifie la requête parmi les requêtes envoyées par la même source.

- *Destination IP Address (32bits*)*:* l'adresse *IP* de destination pour laquelle une route est désirée.

- *Destination Séquence Number (32bits*)*:* Le dernier numéro de séquence reçu dans le passé par le créateur pour n'importe quelle route vers la destination.

- *Originator IP Adress (32bits*): l'adresse IP de la source de la requête de découverte de route.

- *Originator Sequence Number (32bits*)*:* Le numéro de séquence courant de la source contenue dans la table de routage du nœud source.

- ***MP-Q** (32 bits):* Il contient le degré de stabilité de la trajectoire parcourue par les requêtes de découverte de route *RREQ*.

4.3.2 Extension des messages *RREP*

Les messages *RREP* sont étendus pour inclure un nouveau champ (*MP-P*), il spécifie le degré de stabilité d'un itinéraire entre la source et la destination. La valeur de (*MP-P*) est calculée à partir de la formule (4.8) décrite précédemment. Lors de l'installation d'itinéraire, le nœud destination (le dernier nœud) extrait la valeur du *MP-Q* du paquet *RREQ* reçu, puis il l'insert de nouveau dans le champ (*MP-P*) du paquet *RREP* (tableau 4.3 de la page suivante) et l'envoi en mode unicast vers la source suivant le chemin inverse trouvé.

Type	***Flags***	***Reserved***	***Prefix Sz***	***Hop Count***
Destination IP address				
Destination Sequence Number				
Originator IP Address				
Lifetime				
MP-P (champ incluant la qualité de service)				

Tableau 4.3 - Format de message *RREP* de *AODV-SI*

- *Type* (8 bits): ce champ indique le type de paquet, dans ce cas il prend la valeur 2.

- *Flags* ou *drapeaux* (*2 bits*): ce champ contient deux *flags* ;

  - **R** *(Repair flag)* : utilisé pour le multicast.
  - **A** (*Acknowledgment required*): indique si la source doit envoyer un acquittement pour le message *RREP*.

- *Reserved* (*9 bits*): initialisé à la valeur 0 et ignoré à la réception du message.

- *Préfix Sz* (*5 bits*): si la valeur de ce champs est différente de zéro, ce dernier indique que le nœud prochain peut être utilisé pour chaque nœud demandant cette destination et qui possède la même valeur de *Prefix Sz*.

- *Hop Count* (*8 bits*): il contient le nombre de sauts entre la source jusqu'à la destination.

- *Destination IP Address (32 bits)*: l'adresse *IP* de la destination du paquet *RREQ*.

- *Destination Sequence Number (32 bits)*: le numéro de séquence de la destination associé à cette route.

- *Originator IP Adress (32 bits)*: l'adresse *IP* du nœud qui crée la requête.

- *Lifetime (32 bits)*: le temps pour lequel chaque nœud qui reçoit *RREP* considère que la route est valide.

- ***MP-P** (32 bits):* Il contient le degré de stabilité d'un itinéraire possible entre la source et la destination. C'est à partir de cette valeur que la source va sélectionner un chemin.

4.4 Mécanisme de découverte des routes dans AODV-SI

Puisque le protocole *AODV-SI* est une extension du protocole *AODV* [23], il garde la plupart de ces mécanismes de fonctionnement avec quelques modifications pendant la diffusion de la requête de découverte des routes. En premier lieu, un nœud répand une requête de découverte de route (*RREQ*), dans le cas où il aurait besoin de connaître une route vers une certaine destination et qu'une telle route n'est pas disponible. Cela peut arriver si la destination n'est pas connue au préalable, ou si le chemin existant vers la destination a expiré sa durée de vie ou il est devenu défaillant. Par la suite, un contrôle d'admission est effectué, celui-ci consiste à vérifier si la valeur de la probabilité de rupture du lien *PR(ij)* est en dessous d'un seuil fixé préalablement. Si la vérification est admise, la *RREQ* est rediffusée sinon elle est détruite. En fin, la réservation n'est faite que si la *RREP* de la destination est reçue par la source.

Dans *l'AODV-SI*, le champ numéro de séquence destination (*Destination Sequence Number*) du paquet *RREQ*, contient la dernière valeur connue du numéro de séquence, associé au nœud destination. Cette valeur est recopiée de la table de routage. Si le numéro de séquence n'est pas connu, la valeur nulle sera prise par défaut (*Destination Sequence Number* = 0). Le numéro de séquence source (*Originator Sequence Number)* du paquet *RREQ* contient la valeur du numéro de séquence du nœud source.

En opposition avec l'*AODV Standard* [23], si un nœud intermédiaire a une route disponible vers la destination, ce nœud ne peut pas envoyer un *RREP* destiné à la source car ce nœud ne connaît pas à priori l'état de l'ensemble des mobiles intermédiaires en aval sur la route. Dans le but de prévenir cette situation, le drapeau « *D* » de *RREQ* est activé indiquant que seule la destination peut répondre par l'envoi d'un paquet *RREP*.

Le champ *MP-Q* contient le degré de stabilité d'une trajectoire parcourue par les requêtes de découverte de route *RREQ*. Le nœud source diffuse d'abord un paquet de requête *RREQ* avec un *MP-Q* égal à un à tous ses voisins. Quand un nœud *i* de transit (intermédiaire) reçoit un *RREQ*, il récupère d'abord la puissance du signal reçu du nœud émetteur *j* (à partir de sa couche *MAC*) et calcule la probabilité de rupture du lien *PR(i,j).* Si cette valeur est en dessous du seuil fixé, la requête de découverte de route *RREQ* est rejeté sinon la nouvelle valeur du *MP-Q* est calculée suivant la formule (4.7), c'est-à-dire, la probabilité de rupture du lien *PR(i,j)* et multipliée par l'ancienne valeur du *MP-Q* reçu. Cette valeur sera réaffectée au champ *MP-Q* du prochain paquet *RREQ* puis ce dernier sera diffusé. Ces calcules sont effectués à chaque saut, et ainsi de suite, jusqu'à ce que le nœud destination soit localisé.

Quand la destination reçoit la *RREQ*, et si le contrôle d'admission est vérifié, la valeur finale du champ *MP-Q* (degré de stabilité de l'itinéraire trouvé) sera stockée dans *MP-P* de la requête de réponse de route *RREP* puis cette dernière sera envoyée en mode *unicast* vers la source par le chemin inverse pour la réservation et la validation de la route concernée.

Ainsi, après la diffusion du *RREQ*, la source attend tous les paquets de réponse de route (*RREP*). Si ces derniers ne sont pas reçus durant une certaine période (appelée *RREP_WAIT_TIME*), la source peut rediffuser une nouvelle requête *RREQ*. Sinon la source sélectionne l'itinéraire correspondant à la *RREP* ayant la plus petite valeur du champ *MP-P* suivant la formule (4.8) décrite précédemment (plus *MP-P* est petit plus l'itinéraire est stable).

C'est à ce moment que les ressources nécessaires sont réservées définitivement sur chaque nœud. La communication entre source et destination peut alors avoir lieu au débit requis par la source, jusqu'à ce que l'une des extrémités ferme la connexion, ou jusqu'à ce que la route utilisée se brise ou se dégrade.

4.5 Mécanisme de Maintenance des routes dans AODV-SI

Lorsqu'un nœud *i* lié au nœud *i+1*, le long d'une route active, voit la puissance du signal émis par le nœud *i+1* décroitre au dessus d'un seuil fixé préalablement, le nœud *i* précédant la cassure va essayer d'effectuer une réparation locale ou une réparation de bout en bout, s'il se trouve au moins à mis chemin entre la source et la destination. Cette anticipation de réparation de route représente, en quelque sorte, une prédiction de rupture.

Par contre, dans *l'AODV Standard* [23], la détection de la défaillance des routes est effectuée par une transmission périodique du message «*HELLO*» (qui est un *RREP* avec un *TTL* égal à *1*). Si trois messages «*HELLO*» ne sont pas reçus consécutivement à partir d'un nœud voisin, le lien en question est considéré défaillant. Ainsi, la période d'attente pour la recherche d'un nouvel itinéraire va être plus longue que l'*AODV-SI*.

Pour réparer la route localement, le nœud incrémente le numéro de séquence de la destination et initie un processus de découverte de route. La dissémination du *RREQ* se fera cependant à une plus petite échelle en limitant le champ *TTL* (*TIME_TO_LIVE*) du paquet. Si le nœud reçoit un *RREP*, il pourra mettre à jour ses informations de routage. En contre partie, si le nœud ne reçoit pas de *RREP* dans les délais autorisés pour la réparation, il effectue alors une réparation de bout en bout.

Lors de la réparation de bout en bout, le nœud concerné marque dans sa table les routes utilisant ce voisin comme invalide et délivre un message d'erreur « *RERR* » aux voisins en amont de la route. Chaque nœud recevant le message *RERR* exécutera le même ensemble d'opérations précédentes et continuera de propager le paquet *RERR* jusqu'à ce qu'il aboutisse à la source. Seule la source initie de nouveau une procédure de découverte de route après avoir reçu le message d'erreur.

4.6 Conclusion

Dans le but d'améliorer les performances du réseau Ad-Hoc, et particulièrement pour augmenter la stabilité des routes, nous avons étudié une extension du protocole de routage classique *AODV* [23] afin de supporter cette contrainte de qualité de service. Grâce à la simplicité de sa mise en œuvre, elle nécessite peu de changements dans le fonctionnement de base du protocole *AODV* et donc ne nécessite pas beaucoup de modifications dans le code source. Ainsi, nous avons décrit les spécifications de l'extension du protocole *AODV* pour garantir la *QoS* en termes de stabilité d'itinéraire. Et par la suite, nous avons explicité les processus d'installation et de maintenance des itinéraires pour trouver des itinéraires ayant un degré important de stabilité.

Dans le chapitre suivant, nous allons réaliser une série de simulations à l'aide de l'outil *OPNET Modeler* pour mettre en relief les performances des résultats de notre proposition (le protocole *AODV-SI)* en les comparant avec ceux du protocole *AODV* standard. Nous exposerons également, les surcoûts engendrés par l'extension pour les mêmes paramètres et dans les mêmes conditions de simulation.

Chapitre 5

Etude et Simulation du Protocole *AODV* Modifié

5.1. Introduction

Ce présent chapitre couvre, l'étude et l'évaluation des performances du protocole de routage *AODV-SI*. Pour cela, une série de simulations a été faite. L'objectif de cette évaluation est de comparer les performances de l'*AODV-SI* avec celles du protocole *AODV Standard* [23]. Le moyen dont nous disposons pour faire ces tests est le recours au simulateur *OPNET Modeler* [50] qui nous permet de simuler nos protocoles dans un environnement mobile avec de nombreux scénarios et en variant différents paramètres (mobilité, trafic, …).

Pour tester l'efficacité d'un réseau on fait souvent appel à la simulation. En effet, il serait très coûteux voire même impossible de mettre en place un réseau réels afin de tester certains critères. Dans le monde de la simulation des réseaux informatique, beaucoup de travaux ont participé à l'enrichissement de ce domaine. La majorité des travaux de contributions effectués sont d'ordre pédagogique et sont élaborés par des laboratoires informatiques à travers le monde. Beaucoup d'axes de recherche se basent sur ces simulateurs. Les simulateurs les plus répandus sont :

- ***Omnet++*** : Utilisé sur la plate-forme *Microsoft* W*indows*, *Unix*. Sa licence est gratuite pour les universitaires et pour toute utilisation non lucrative. Il ne semble pas particulièrement prévu pour les réseaux sans fil.

- ***NS-2*** : Utilisé sur la plate-forme *Unix* (*Linux*, *Solaris*, *Mac OS*), *Microsoft Windows*. Sa licence est gratuite. *NS-2* est très utilisé pour les réseaux filaires et les réseaux Ad-Hoc. Toutefois les modèles de couche physique sont simplistes. Le développement des protocoles s'effectue en *C++* et en *OTCL*. Le résultat de la simulation étant essentiellement composé d'un fichier retraçant l'ensemble des envois, de réceptions et de suppressions de paquets. Du fait de sa popularité, de nombreux protocoles sont a priori disponibles pour *NS-2*, et quelques protocoles spécifiques aux réseaux de capteurs sont disponibles.

- ***GlomoSim*** : Utilisé sur la plate-forme *Unix* avec une licence gratuite pour les universitaires. Peu de modèles semblent disponibles. Le moteur de *GlomoSim* est basé sur la bibliothèque *Parsec* (le langage de programmation de *GlomoSim*). Le simulateur peut donc être parallélisé, et l'apprentissage de cette API peut se révéler difficile.

- ***Jist/SWANS*** : Développé sous le langage *Java*. Sa licence est gratuite. *Jist* est le moteur de simulation, *SWANS* est le simulateur (l'interface). *Jist* permet d'utiliser, comme générateur de trafic, n'importe quelle application Java. Il souffre cependant du manque de modèles lié à sa jeunesse. Les protocoles sont conçus comme des composants indépendants interconnectés par des interfaces.

- ***OPNET*** : Utilise la plate-forme *Microsoft Windows (NT, 2000, XP)* et *Solaris*. Le développent s'effectue en *C++*, au travers de l'interface du logiciel. L'approche orientée objet associée à des éditeurs graphiques intégrés simplifie la composition des réseaux et des équipements. Il est réputé dans l'industrie pour la modélisation et la simulation de réseaux.

Le modèle du protocole *AODV* [23] de *MaNet* [2] est disponible sous « *OPNET Modeler 14.0* ». Ainsi, dans notre simulation on met à contribution ce dernier pour analyser quelques propriétés du protocole de routage *AODV-SI*.

Nous abordons dans ce qui suit l'outil de simulation *OPNET Modeler* et l'intérêt de la simulation. Plusieurs modèles de propagation radio et des modèles de mobilité supportés par *OPNET Modeler* sont listés pour permettre le choix d'un modèle de simulation suivant lequel les métriques de performance du protocole *AODV-SI* sont évaluées. Nous décrivons ensuite les différentes métriques mesurées ainsi que le modèle de simulation adopté et nous présentons les résultats obtenus suivis d'interprétations et de discussions.

5.2 Présentation d'OPNET

Initialement développé au *MIT* (*Massachusetts Institute of Technology*) et commercialisé en 1987 comme le premier simulateur de réseaux. C'est ainsi qu'est né *OPNET*, qui s'est imposé dans le monde de la recherche et du développement. *OPNET* est une offre de logiciels de modélisation et de simulation de réseaux s'adressant à différents publics : « *OPNET Modeler* » pour la communauté de la recherche scientifique, « *SP Guru* » pour les opérateurs, et « *IT Guru* » pour les entreprises.

OPNET Modeler [50] est un simulateur à évènements discrets qui offre un environnement graphique pour permettre la modélisation, l'étude, la simulation et l'évaluation des performances des réseaux et des protocoles de communication avec une grande flexibilité. Il permet la modélisation de toutes les couches du modèle *OSI* d'un système de communication grâce à des bibliothèques de modèles de nœuds et de liens prédéveloppés (routeurs, commutateurs, stations de travail, serveurs, téléphone portable, satellite, liaison point à point, liaison par bus et liaison satellite) et de protocoles (*TCP/IP*, *FTP*, *FDDI*, Ethernet, *ATM*…). Le module spécifique « *Wireless Radio* » permet la simulation des réseaux de radiocommunication courtes et longues distances ainsi que les réseaux satellitaires.

Des chercheurs de *l'INPL* (Institut *National Polytechnique de Lorraine à Nancy-France*), ont fait usage *d'OPNET* sur des réseaux à communication de paquets pour étudier le respect des contraintes temps réel. Le groupe de recherches *IETF* (*The Internet Engineering Task Force*), utilise aussi *OPNET* comme un outil de simulation.

5.3 Les fonctionnalités principales d'OPNET

OPNET Modeler [50] est utilisé par les entreprises technologiques les plus performantes pour accélérer leurs procédés de recherches et développements. Il continue d'être la référence sur l'état de l'art avec les fonctionnalités principale suivantes :

- Modeler est le moteur le plus modulable et plus évolutif des moteurs de simulation. Pour des modèles de réseaux filaires et sans fil; il permet d'utiliser des *Runtimes* de simulation qui utilisent des techniques d'accélérations.

- Les modèles hiérarchisés de réseaux gère des topologies complexes avec un nombre de sous réseaux imbriqués illimité.

- La programmation des modèles est orientée objet. Les équipements et les protocoles sont programmés sous forme de modules de classes incluant héritage et organisation des classes.

- La modélisation est claire et simplifiée. C'est le cas pour la modification du comportement des différents objets au « niveau processus » et l'intégration entre eux au « niveau équipements ». Il en est de même au « niveau réseau » pour la création des liens entre les équipements. Il est possible de lancer scénarios au sein des projets afin de pouvoir comparer les différentes architectures.

- Modeler reproduit chaque comportement sur une base de code *C* ou *C++* (*FSM finite state machine*). Chaque élément est contrôlable par l'utilisateur.

- Support complet pour la réalisation de protocoles. Plus de 1000 fonctions sont incluses et les librairies assurent l'aide à la réalisation des protocoles.

- Réseaux sans fils point à point et multipoints: Le comportement des liens réseaux est ouvert et programmable. Les caractéristiques des délais, de la disponibilité, des débits des liens sont modifiables. Ceci comprend le niveau physique et les modifications des caractéristiques en fonction de l'environnement. La librairie « *Longley-Rice* » est intégrée en standard dans *Modeler* sans fil. Il en est de même pour les librairies « *TIREM* » et « *Free Space* ».

- *Modeler* est la plateforme la plus évoluée intégrant de la simulation séquentielle et parallèle *DES* (*discrete event simulation*) sous la forme hybride et analytique tout en intégrant un noyau 64 bits et un bus *HLA*. L'option *SITL* (*system in the loop*) permet de bénéficier d'une liaison entre le monde réel et le monde virtuel.

- Totalement ouvert, *Modeler* possède des *API* pour des ajouts de programmes complémentaires. Pour assurer la confidentialité et la protection du savoir, les modèles peuvent être encryptés. Le code source est disponible pour tous les modèles.

- Debugger Intégré : Il permet de valider rapidement une simulation ou de situer les problèmes, s'il y en a.

- Outil d'analyse intégré : Interface simple pour visualiser les résultats d'une simulation. Interface permettant de visualiser des séries, des courbes, des fonctions de probabilité. Le tout exportable en bilan ou au format *XML*.

- Animations : Il est possible de visualiser le comportement d'un modèle sous forme d'une animation. Il est possible de visualiser graphiquement les données statistiques pendant l'exécution de la simulation.

- Importation de données: Les formats texte, *XML* et les outils standard tels que ceux de *Cisco*, *HP*, *CA*, *NetScout*, *BMC*, *Concord* (*CA*), *cflow*, *tcp-dump* et autres sont supportés.

- Librairie complète incluant les applications plusieurs tiers, la voix, *http, TCP, IP OSPF, BGP, EIGRP, RIP, RSVP, Frame Relay, FDDI, Ethernet, ATM, 802.11* sans fil, *802.16, WiMax , IPv6, MPLS, DOCSIS, UMTS, IP Multicast, Circuit Switch, MANET, Mobile IP, IS-IS*, les caractéristiques des satellites et bien d'autres. Tous ces modèles de normes sont fournis sous forme de *FSM* (*finite state machine*) avec le code source.

- Les équipements réseaux : La librairie des modèles standards inclut de nombreux équipements des constructeurs sous une forme générique dont des routeurs, des *Switches*, des stations de travail, des générateurs de paquets. L'ensemble s'assemble rapidement pour réaliser vos propres modèles en utilisant le module «*Device creator*». Il est possible d'agréer du trafic venant d'un *LAN* ou d'un réseau matérialisé par un nuage.

- Modélisation de réseaux de mobiles : Il existe des modèles de réseaux cellulaires, de mobiles Ad-Hoc, de *LAN* sans fil *Wifi*, de réseaux sans fil *WiMax* et de nombreux réseaux sur base d'équipements liés aux mobiles. Contrôle de chaque équipement, position dynamique ou trajectoire prédéfinie. Il est possible d'ajouter des cartes telles que *CADRG/CID* et des données venant d'autres fonds d'écrans contextuels comme *DTED*, *USGS* ou *OpenFlight* afin de mieux prendre en compte les considérations liées aux effets du terrain.

- Windows 2000, XP, Linux et Sun Solaris: Les modèles peuvent être partagés en étant indépendants des plateformes. La version sous Windows Vista et en cours de qualification.

- Gestion souple des licences: Droit d'utilisation la licence transportable. Fourniture automatique de numéro de licence via Internet et gestion web du management des licences.

5.4 Concepts de bases du simulateur OPNET

Cette section explique les concepts de bases et le fonctionnement général du simulateur *OPNET Modeler*. Ce simulateur est basé sur une série d'éditeurs hiérarchisés et interfaces graphiques qui parallélisent la structure du réseau réel, des équipements et des protocoles (le domaine réseau, le domaine équipement et le domaine processus).

5.4.1 Présentation des interfaces de l'outil *OPNET*

Parmi les nombreuses interfaces que propose *OPNET Modeler* au démarrage, nous trouvons :

- **Editeur de projet :**

L'interface principale du logiciel permet d'implanter des modèles issus des bibliothèques *OPNET* ainsi que des modèles créés par l'utilisateur, de configurer puis lancer des simulations, et de visualiser les résultats des simulations. La figure 5.1 de la page suivante illustre les principales fonctions de cette interface sous formes d'icônes (Ouvrir la palette d'objet, Vérification des liens, Mise en panne d'un appareil ou d'un lien collectés, Remise en marche d'un appareil ou d'un lien, Visualiser les graphiques et statistiques, Visualiser le rapport le plus récent, Zoom +/-, Retour au réseau supérieur, Lancer la simulation, Visualiser tous les graphiques…).

Figure 5.1 – Editeur de projet

- **Editeur de réseaux :**

Le domaine réseau définit la topologie du réseau de communication dans l'éditeur de projet (*Network Model Editor*). Les entités communicantes du réseau (Figure 5.2) sont des nœuds (routeurs, hôtes, commutateurs…) reliés entre eux par des liens. Chaque entité est affectée à un modèle qui spécifie ces fonctions dans la fenêtre éditeur de nœud (*Node Model Editor*).

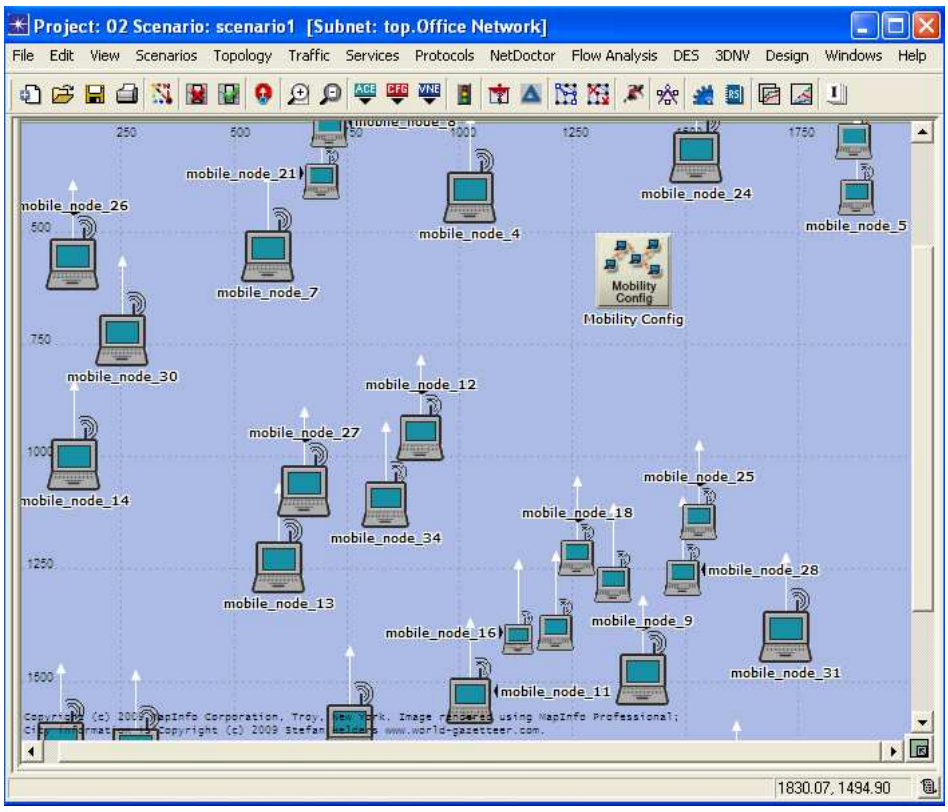


Figure 5.2 – Editeur de réseau

Les méthodes de « copier/coller » des équipements et des liens à partir des palettes d'objets de l'éditeur permettent de réaliser le réseau. Il est également possible d'utiliser la fonction d'importation rapide de l'ensemble de la configuration. Il est possible d'utiliser la librairie *OPNET Modeler* ou de réaliser sa propre palette intégrant ses propres équipements et liens.

L'éditeur de projets intègre également un environnement géographique avec des caractéristiques dimensionnelles reprenant les composants utiles pour la modélisation de réseaux câblés ou sans fils. L'utilisation du menu Protocoles permet de configurer rapidement les protocoles et d'activer des paramètres particuliers.

- **Editeur d'équipements :**

Le domaine équipement permet de concevoir dans la fenêtre de l'éditeur d'équipement (*Node Model Editor*) les éléments qui peuvent envoyer et recevoir des données et peuvent également être connectés dans le domaine réseau (entités communicantes). Un équipement (Figure 5.3) est formé d'un ensemble de blocs fonctionnels appelés modules de processus (processus, files d'attente, générateurs, émetteurs, récepteurs, antennes….) qui sont développés par un éditeur de processus (*Process Model Editor*). Chaque module peut générer, envoyer et recevoir des paquets des autres modules pour réaliser sa fonction. Les Modules représentent les applications, les couches de protocoles, les algorithmes, les ressources physiques comme les buffers, les ports *TCP* et les bus de données.

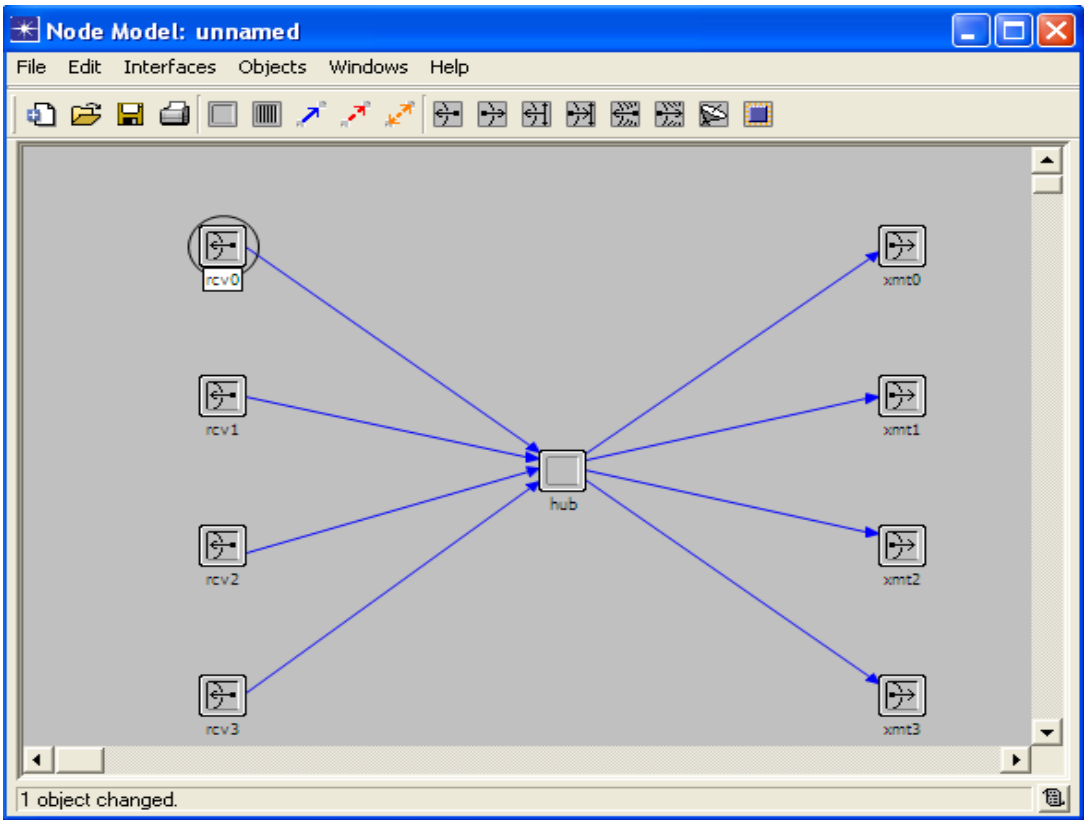


Figure 5.3 – Editeur d'équipements

- **Editeur de processus :**

Le domaine processus définit le comportement d'un module appartenant à un équipement par un diagramme d'états finis (*FSM : finite state machine*) qui décrivent les conditions pour passer d'un état à un autre (Figure 5.4). Les états sont codés en langage *C* ou *C++*.

Il est alors nécessaire de compiler le processus afin de le rendre interprétable et utilisable dans le modèle d'équipement. Chaque *FSM* peut définir des variables d'état et peut faire des appels aux codes dans les librairies disponibles. L'éditeur de processus est utilisable pour développer un modèle complètement nouveau de processus en partant des spécifications papiers.

Les Framework de la modélisation de processus supportent le traitement complet du multi-thread et les ordinateurs dont les architectures sont parallèles.

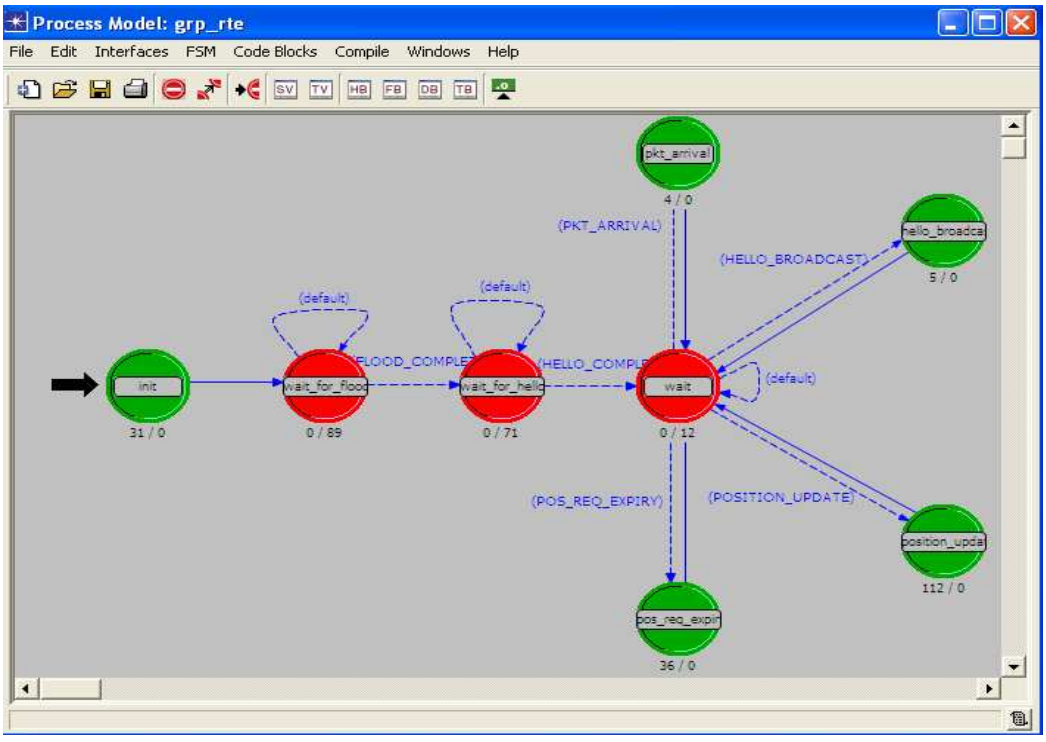


Figure 5.4 - Editeur de processus

- ***Antenna Pattern* :**

Cette interface permet de modéliser une antenne pour la radiocommunication par son diagramme de rayonnement *3D* et ses coordonnées polaires.

- ***Modulation Curve*** **:**

Donne une visualisation du taux d'erreur binaire *BER* (*Bit Error Rate*) en fonction du rapport signal sur bruit *SNR* (*Signal Noise Rate*) pour différents types de modulations (*BPSK, MSK, FSK,* …).

- ***Analysis Configuration*****:**

Permet la configuration du stockage des résultats issus des simulations, sous différentes formes.

- ***Simulation Sequence*****:**

Permet de paramétrer la ou les simulations *OPNET* en temps et attributs des modèles (types de liens, d'antenne, de services …).

- ***Packet format editor:***

Permet de définir la structure interne des formats de paquets (pour les réseaux filaires et sans fil) en décrivant en détails tous les champs qui les composent.

5.4.2 Modèles de propagation radio sous *OPNET Modeler*

Le simulateur utilise le modèle de propagation par défaut (modèle en espace libre « *Free Space* ») pour tout calcul d'atténuation du signal. Le module de modélisation de terrain *TMM* (*Terrain Modeling Module*) comprend plusieurs modèles de propagation (par exemple: *Free Space, Longley-Rice, Hata, CCIR, Walfisch-Ikegami, and TIREM Propagation Model*). Pour des terrains particuliers, *OPNET* peut créer des modèles additionnels spécifiques.

- ***Le modèle de propagation en espace libre (Free space model):***

Ce modèle considère le cas idéal où il y a un seul chemin de propagation entre l'émetteur et le récepteur et qu'il est en vue directe. L'équation (5.1) ci-dessous permet de calculer la puissance du signal reçu en environnement libre à une distance « *d* » de l'émetteur.

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{16 \pi^2 d^2 L} \qquad (5.1)$$

Où P_t est la puissance d'émission, G_t et G_r les gains respectifs des antennes de l'émetteur et du récepteur. L (avec $L >= 1$) est la perte du système, et λ est la longueur d'onde.

Ce modèle de propagation représente les zones de communication comme un cercle autour de l'émetteur. Si un récepteur est dans ce cercle il reçoit tous les paquets, s'il est en dehors il n'en reçoit aucun.

- ***Le modèle de propagation Longley-Rice:***

Ce modèle prédit l'atténuation des transmissions à longue distance à travers des terrain irégulier et les espaces libres. Le modèle est conçu pour les fréquences de *20 MHZ* à *40 GHZ* et pour les chemins de *0,1 Km* à *2000 Km*. Ce modèle est répandu, et aprouvé par *FCC* comme méthode de prédiction de la couverture de la zone de diffusion [51]

- ***Le modèle de propagation Hata:***

Le modèle est largement utilsé pour la modèlisation dans un environement urbain. Ce modèle possède un paramètre, appelé « *Area Type* ». Ce paramètre spécifie la taille relative et la distribution des constructions. On peut affecter à ce paramètre l'ensemble des valeurs prédéfinies suivantes : « *large city* », « *small city* », « *suburban* », et « *open areas* » (grande cité, petite cité, sous-zone ou espace ouvert).

- ***Le modèle de propagation CCIR:***

Le modèle *CCIR* (publié par : le Comité Consultatif International des Radiocommunications) est une version simplifiée du modèle de propagation *Hata*. Ce modèle possède un seul paramètre « *Building Percentage* » représentant le pourcentage de constructions couvrant cette zone. Le paramètre par défaut représente une cité de petite à moyenne taille et avec un taux de construction de *15,8%* dans cette zone.

- ***Le modèle de propagation Walfisch-Ikegami:***

Le modèle de propagation *Walfisch-Ikegami* est un modèle semi-déterministe prévu pour les applications cellulaires utilisant des cellules larges ou moyenne dans les zones renfermant des constructions. Il est mieux adapté pour les antennes des stations de bases situées à une bonne hauteur.

Les paramètres de ce modèle sont : *Path Loss* (atténuation), Hauteur nominale des constructions, séparation des constructions, largeur de la rue, et l'angle d'incidence de l'onde avec la rue.

- ***Le modèle de propagation TIREM:***

Le modèle *TIREM* (*Terrain Integrated Rough Earth Model)* prédit la l'atténuation de la propagation des radios fréquence (*RF*) de 1 à 40 GHZ et à travers les surfaces de la mer et des terrains irréguliers. Les fonctionnalités de *TIREM* incluent les modèles *TIREM3* et *TIREM4*. *TIREM3* est le modèle de propagation standard aux états unis. *TIREM4* est une version optimisé de *TIREM3* permettant le calcul rapide des valeurs du *path loss* (atténuation*)*. Les paramètres de ce modèle permettent de spécifier :

- La conductivité de l'endroit (moyenne sol = *0.005*, moyenne eau = *5.0*),
- La perméabilité relative (moyenne sol = *15.0*, moyenne eau = *81.0*),
- L'humidité (supérieur à zéro, en gramme / mètre cube),
- La réfractivité de la surface (Intervalle de : *250.0 à 400.0*),
- La résolution (distance en mètre entre les échantillons de terrains).

5.4.3 Modèles de mobilité sous *OPNET Modeler*

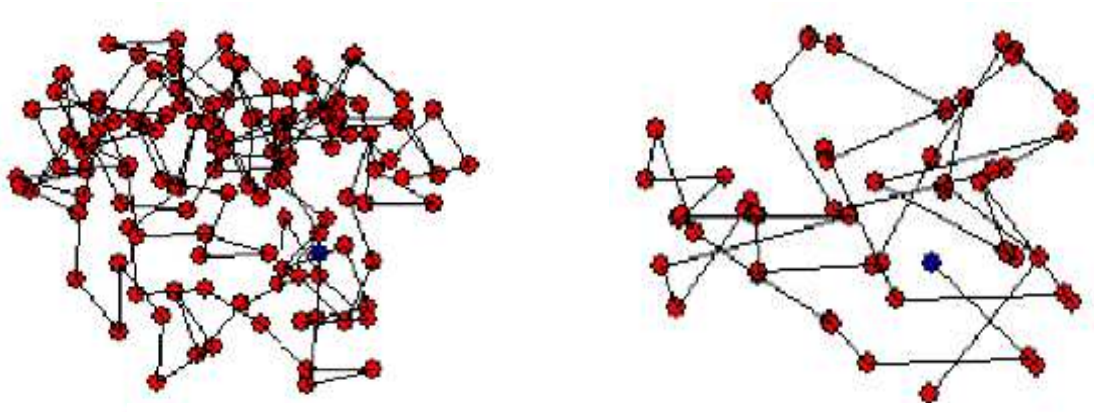
Afin d'évaluer les performances d'un protocole, ce dernier doit être testé sous plusieurs modèles de mobilité des nœuds. Un tel modèle doit être capable d'imiter le mouvement des nœuds tel que souhaité dans un certain scénario. Ainsi, l'évaluation d'un protocole de routage Ad-Hoc dépend du choix d'un modèle de mobilité pour illustrer les mouvements réalistes des utilisateurs.

Les modèles de mobilité d'entités représentent les nœuds mobiles dont les mouvements sont indépendant les uns des l'autres. D'autre part, les modèles de mobilité de groupes représentent les nœuds mobiles dont les mouvements dépendent les uns des autres et ils tendent à être plus réalistes dans les applications impliquant la communication de groupe.

- ***Random Walk (RW)***

Dans ce modèle *RW* [52], chaque nœud dans la simulation choisit aléatoirement un angle de direction entre 0 et 2π et une vitesse entre *Vmin* et *Vmax*. Le déplacement du nœud se fait pendant un temps *t* ou d'une distance *d*. Ce modèle était premièrement décrit mathématiquement par *Einstein* en 1926. A la fin de son voyage, le nœud choisit une nouvelle direction et une nouvelle vitesse et se déplace de nouveau.

On peut bien remarquer que le changement de la direction et de la vitesse sont absolument aléatoire et indépendant du choix précédent. La figure 5.5.A montre le mouvement d'un nœud qui voyage pendant un temps de *t* secondes alors que dans la Figure 5.5.B, le nœud se déplace d'une distance *d* définie.



-A- **-B-**

Figure 5.5 – trajectoires du *Random Walk* [52]
(-A- suivant la distance *d*, -B- suivant un temps *t*)

- ***Random Waypoint(RWP)***

Ce modèle était d'abord utilisé par *Johnson et Maltz* [19] dans l'évaluation du protocole de routage *DSR* et était ensuite raffiné par les mêmes auteurs. Le « *Random Waypoint*» [52] est conçu pour modéliser tous les scénarios dans lesquels, les nœuds se déplacent vers une destination, prennent un repos en arrivant, avant de se déplacer vers une autre destination et ainsi de suite. Dans ce modèle chaque nœud choisit aléatoirement, comme destination un point de coordonnées *(x, y)* dans la zone de la surface de simulation, et une vitesse entre *0* et *Vmax.* Le nœud voyage vers la destination choisie avec la vitesse choisie. A l'arrivée, le nœud prend un temps de repos avant de choisir à nouveau une nouvelle destination et une nouvelle vitesse pour répéter le même processus.

Des études ont été faites sur ce modèle puisqu'il est le modèle le plus utilisé dans les simulations dû à la facilité de son implémentation. Certaines études ont traité l'initialisation de ce modèle et le temps de convergence des simulations dans le cas où les nœuds commencent par prendre un temps de repos. Le «*Random Waypoint*», dans sa forme courante, n'atteint pas un état d'équilibre, mais plutôt que la vitesse diminue sans interruption pendant que la simulation progresse, ce qui peut fausser les résultats. En se basant sur les analyses de l'état d'équilibre du modèle, les auteurs [19] proposent une solution simple qui est de choisir une valeur strictement positive pour la vitesse minimale.

On remarque que le *Random Waypoint* est proche du *Random Walk* à la différence près que la destination choisie est toujours un point intérieur à la surface de simulation, ce qui élimine tout effet de bord (Figure 5.6).

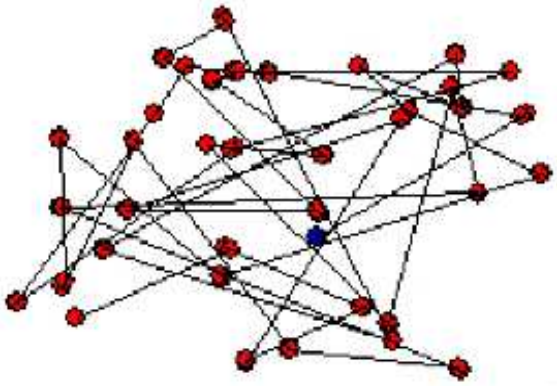


Figure 5.6– trajectoires du *Random Waypoint* [52]

- ***Random Direction (RD)***

Le *Random Direction* [52] a été créé pour éviter l'effet de la concentration des nœuds au centre produit par le *Random Waypoint*. Dans ce modèle, chaque nœud choisit aléatoirement, comme dans le *Random Walk*, une direction qui a un angle entre 0 et 2π et une vitesse entre *Vmin* et *Vmax*.

La différence entre ce modèle et le *Random Walk* est qu'ici le nœud ne voyage pas pendant un certain temps ou d'une certaine distance mais se déplace suivant la direction choisie jusqu'à atteindre le bord de la surface de simulation où il prend un temps de repos. Une fois le temps de pause terminé, le nœud choisit de nouveau et aléatoirement une nouvelle direction et une nouvelle vitesse et répète le même processus.

La figure 5.7 montre un nœud utilisant le *Random Direction* comme modèle de mobilité. La position initiale du nœud est au centre de la surface de simulation. Le nœud commence à se déplacer et chaque fois il se déplace jusqu'au bord où il prend un temps de repos avant de changer sa direction et sa vitesse.

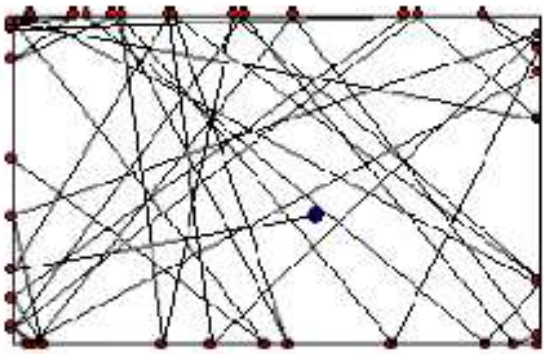


Figure 5.7– trajectoires du *Random Direction (RD)* [52].

5.4.4 Cycle traditionnel d'un projet sous *OPNET*

Il y a six phases dans le cycle de modélisation et de simulation sous OPNET. On commence par la phase de création de projets (*Project creation*), puis la création ou/et la mise à jour d'un scénario (*Scenario creation/update*), ensuite le choix des statistiques à collecter (*Choose statistics*), après soit le lancement des simulations (*Simulation execution*) soit la duplication du scénario (*Scenario duplication*) et enfin la comparaison des résultats (*Results analysis*). Si on a besoin de mise à jour du scénario, on recommence à partir de la deuxième phase (figure 5.8).

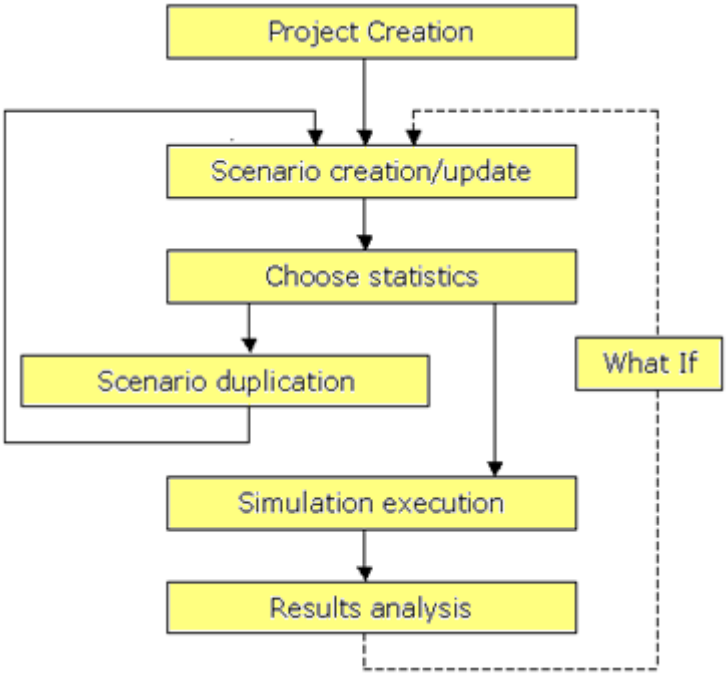


Figure 5.8 – Etapes de modélisation et simulation d'un projet

5.5 Objectifs de la simulation

Le but général de notre simulation est d'analyser un certain nombre de propriétés qui nous paraissent utiles dans la *QoS*, qu'offre notre proposition :

- évaluer les performances du protocole *AODV-SI* en le comparant au protocole *AODV standard*;
- évaluer les insuffisances ou les surcoûts du protocole *AODV-SI* par rapport au protocole *AODV standard*.

5.5.1 Métriques de la simulation

Dans le but de tester le protocole suivant les objectifs précédents, les simulations sont faites par rapport à six métriques classées selon deux groupes (métriques de performances, métriques d'insuffisances ou de surcoûts).

5.5.1.1 Métriques de performances

Nous avons choisi ces derniers car ils montrent une vision globale sur les améliorations possibles apportées à cette application par le nouveau mécanisme de routage avec *QoS* mis en place.

Ainsi, les paramètres choisis pour l'évaluation des performances de l'extension *AODV-SI* par rapport à ceux du protocole *AODV Standard* [23] sont:

- ***Le nombre de paquets de données reçus***

Cette métrique représente le nombre de paquets de données délivrées à toutes les destinations suivant les itinéraires choisies par toutes les sources dans le réseau. Cependant, dans les communications sans-fil, les paquets peuvent ne pas être livrés (une perte de paquets) due à des collisions aussi bien qu'aux ruptures de liens entre les nœuds.

Nous calculons la moyenne du nombre de paquets de données reçues par toutes les destinations pendant la durée de simulation du protocole *AODV Standard* [23] et l'*AODV-SI*. Cette métrique nous permet de connaitre le débit et la qualité de réception des données.

- ***Le nombre d'erreurs de route:***

Cette métrique représente le nombre de paquets d'erreurs de route envoyés par tous les nœuds du réseau qui ont détecté une coupure des routes parcourues par un flux de données. Un nœud doit diffuser un message « *Hello* » à tous ses voisins pour vérifier et confirmer l'accessibilité à ces derniers. S'il a des données à transmettre via un nœud et que l'accessibilité à ce dernier n'a pas été validée, il entame une procédure de réparation locale. Si cette dernière échoue encore, il envoi un message d'erreur de route à tous les nœuds utilisant ce saut pour atteindre les différentes destinations.

Nous calculons la moyenne du nombre d'erreurs de route générées pendant la simulation du protocole *AODV Standard* [23] et l'*AODV-SI*. Cette métrique nous permet de connaitre la fiabilité des connexions établies à travers le réseau au cours du temps, dans notre cas, plus le nombre d'erreurs est petit plus la stabilité des routes est grande.

- ***Le nombre de réponses de route:***

Cette métrique représente le nombre de paquets de réponses de routes envoyés par tous les nœuds destinations du réseau. Quand la destination reçoit une requête de découverte de route, elle envoie une réponse de route à la source.

Nous calculons la moyenne du nombre de paquets de réponses de route envoyés par les destinations vers les sources pendant toute la durée de simulation des deux protocoles *AODV Standard* [23] et l'*AODV-SI*. Cette métrique représente l'efficacité du protocole en termes de choix des chemins optimaux (sélection du plus stable).

- ***Le nombre de sauts par route :***

Cette métrique représente le nombre de sauts, correspondants aux itinéraires des différentes destinations, dans la table de routage de tous les nœuds du réseau.

Nous calculons la moyenne du nombre de sauts par route de tous les itinéraires générés pendant la durée de simulation du protocole *AODV Standard* [23] et l'*AODV-SI*. Cette métrique nous permet de connaitre la stabilité des connexions établies à travers le réseau au cours du temps, dans notre cas, plus le nombre de sauts est important plus les nœuds sont proches les uns des autres et par conséquent les liens sont plus forts.

5.5.1.2 Métriques d'insuffisances et de surcoûts

Effectuer les contrôles d'admission et faire les prédictions de rupture de route engendrent un surcoût en délai d'acheminement et en trafic de contrôle dans le réseau. On propose de représenter les paramètres pour l'évaluation des surcoûts (insuffisances) de l'*AODV-SI* comparés à ceux du protocole *AODV Standard* [23]:

- ***Le délai de découverte de route :***

Cette métrique représente le temps de découverte de route vers une destination spécifique pour tous les nœuds dans le réseau. Le temps de découverte de route à une destination spécifique est le temps que met une requête de découverte de route pour parcourir tous les nœuds intermédiaires jusqu'à atteindre la destination plus le temps que met la réponse de route à atteindre la source.

Nous calculons la moyenne du délai de sélection de route de touts les nœuds sources de trafic durant la simulation du protocole *AODV Standard* [23] et l'extension *AODV-SI*.

- ***Le trafic de contrôle envoyé :***

Cette métrique représente le nombre de paquets de contrôle utilisés par le protocole pour l'établissement et le maintient des routes entre les nœuds sources et destinations du réseau. Un nœud doit diffuser les paquets de découverte de route *RREQ*, il doit aussi transmettre ou retransmettre les paquets de réponse de route *RREP*, il doit encore envoyer les paquets d'erreurs *RERR* et il doit finalement diffuser les messages « *Hello* » à tous ses voisins. Tous ces paquets représentent le trafic de contrôle envoyé.

Nous calculons la moyenne du nombre de paquets de contrôle générées pendant la simulation du protocole *AODV Standard* [23] et l'*AODV-SI*. Cette métrique nous permet de connaitre les couts de sélection et de maintient des routes.

- ***Le délai de bout en bout des paquets:***

Cette métrique représente le temps qui sépare le moment d'envoi d'un paquet de la couche transport de la source et le moment de réception de ce paquet par la couche transport de la destination. Il inclut le temps de latence pour la découverte de routes, le temps de passage dans les files d'attente des nœuds intermédiaires et le temps de transmission d'un saut vers un autre.

Nous calculons la moyenne du délai de bout en bout par rapport à tous les paquets générés pendant la simulation du protocole *AODV Standard* [23] et l'*AODV-SI*.

5.6 Modèle de simulation

Les simulations sont faites sur « *OPNET Modeler 14.0* » sous le système d'exploitation «*Microsoft Windows XP – Service Pack 2* ». Nous allons présenter les différents scénarios et modèles utilisés dans notre expérimentation.

5.6.1 Scénarios de simulations

Dans les scénarios de nos simulations, la topologie est de *50 nœuds* (*MANET Station*) dans laquelle ils bougent aléatoirement dans un environnement de taille *1000 x 1000 m* (*Mobility Configuration*). Le nœud de contrôle *Rx_Group_Config* (figure 5.9, dans la page suivante) est mis en place pour accélérer le processus de simulation. Il est configuré pour éliminer toutes les réceptions qui n'atteignent pas le seuil minimal de la puissance de réception fixé (soit *-90 dB*, dans nos simulations).

Nous effectuons des simulations d'une durée de *300 secondes* pour les différents scénarios de notre expérimentation. Les scénarios se caractérisent par le même nombre de nœuds, le même nombre de connexions de trafic, le même modèle de mobilité et de terrain (propagation). Les statistiques utilisées dans l'analyse représentent la moyenne des résultats.

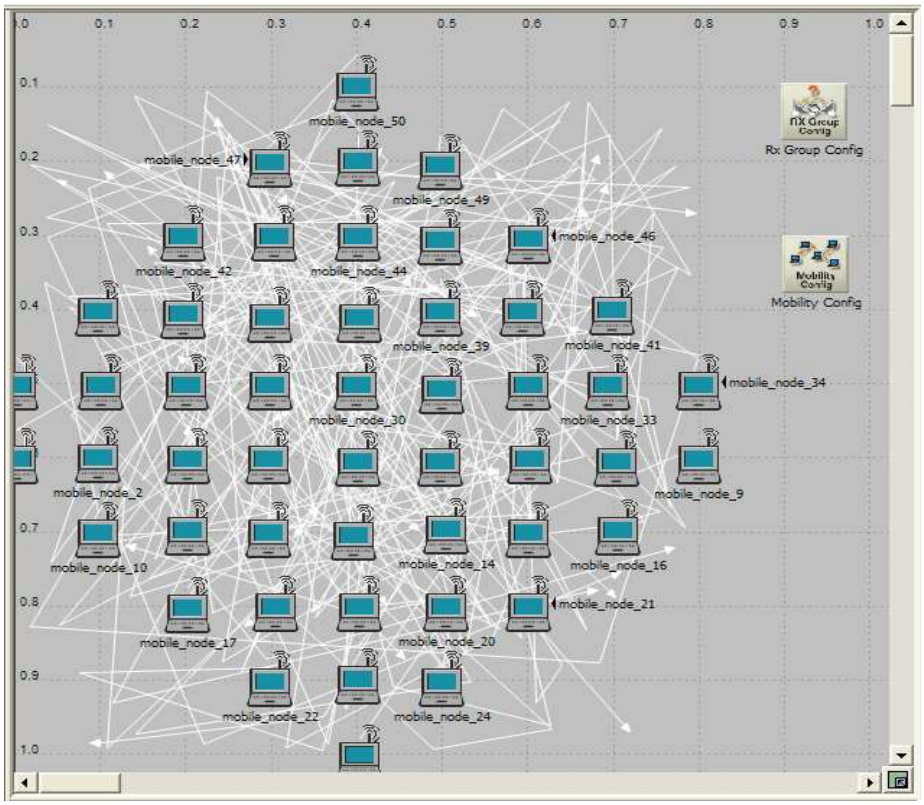


Figure 5.9 – Scénario d'un réseau de 50 nœuds (*MANET station*) mobiles

Au début de chaque simulation, les paramètres des protocoles (*AODV*) de chaque nœud mobile doivent être configurés suivant le tableau 5.1 comme suit :

Paramètres d'un nœud	Valeurs
Temps d'expiration de l'entrée de la table (*Active route Timeout*)	*3 secondes*
Nombre de fois pour lequel une nouvelle *RREQ* est émise à une destination	*5 fois*
Nombre de paquets *Hello* perdus avant d'affirmer la rupture du lien	*2 Hellos*
Durée de vie au départ d'un paquet (*TTL Start*) en nombre de sauts	*1 saut*
Durée de vie incrémentée d'un paquet (*TTL Increment*) en nombre de sauts	*2 sauts*
Durée de vie maximum d'un paquet (*TTL Threshold*) en nombre de sauts	*7 sauts*
Nombre de sauts maximum entre la source et la destination	*34 sauts*

Tableau 5.1 - Paramètres de configuration des protocoles de routage

5.6.2 Modèle de propagation

Dans toutes ces simulations on utilise les paramètres par défaut pour le médium et le modèle de propagation radio. Le médium utilise la technique d'étalement de spectre par séquence directe (*DSSS*) avec une bande *2,4 MHZ* divisée en *14* canaux de *22 MHZ* et le seuil de la puissance de réception est fixé à -90 dB. Le module de modélisation de terrain *TMM* (*Terrain Modeling Module*) utilise le modèle *TIREM3* «*Terrain Integrated Rough Earth Model*» avec comme paramètres : *Conductivité = 0.005, Perméabilité = 15.0, Humidité = 10.0, Réfractivité = 250.0 et Résolution = 100.0*. Le protocole *IEEE 802.11b* est utilisé comme protocole d'accès au médium.

Les valeurs présentées dans le tableau 5.2 sont les paramètres spécifiques à notre modèle de propagation.

Paramètres de propagation	Valeurs
Modèle de propagation	*TIREM3*
Protocole *MAC (protocole d'accès au médium*)	*IEEE802.11b*
Bande passante de la couche physique (*Data Rate*)	*11 Mbps*
Puissance de transmission (*Transmit Power*)	*0,001 Watt*
Seuil de réception d'un paquet (*Packet Reception-Power Threshold*)	*-90 dB*
Portée maximum de la communication directe entre deux nœuds	*300 m*

Tableau 5. 2 - Paramètres de propagation dans le réseau

5.6.3 Modèle de mobilité

L'impact du mouvement des nœuds sur une simulation est très important. La figure 5.9 du scénario de simulation (page 102) montre les trajectoires effectuées par les nœuds mobiles (*MANET Station*) suivant les paramètres du nœud de contrôle *Mobility_Config*.

Dans nos simulations, les nœuds mobiles utilisent « *Random Way Point* » comme modèle de mobilité. Ce modèle est utilisé souvent dans l'analyse de performance des protocoles de routage. Ainsi, on a choisi de paramétrer la simulation suivant une mobilité moyenne dans laquelle les mobiles se déplacent avec une vitesse uniformément distribuée entre *0* et *20 m/s*. Le temps de pause, quant à lui, est uniforme variant entre *0* et *20 secondes*.

Les valeurs présentées dans le tableau 5.3 sont les paramètres spécifiques à notre modèle de mobilité.

Paramètres de mobilité	Valeurs
Modèle de mobilité (*suivant le temps de simulation*)	*Random Way Point*
Vitesse de la mobilité (*Speed, en mètre/sec)*	*Uniforme entre 0 et 20*
Temps de pause (*Pause Time, en seconde*)	*20 secondes*
Début de la mobilité (*Start Time, en seconde*)	*Uniforme entre 0 et 20*
Fin de la mobilité (*Stop Time, en seconde*)	*Fin simulation (300)*

Tableau 5.3 - Paramètres de mobilité du nœud

5.6.4 Modèle de trafic de données

Le paramètre suivant à définir est l'établissement des différents trafics entre les nœuds du réseau. On a choisi des sources de trafics à débit constant *CBR* (*Constant Bit Rate*) dont le fonctionnement est assez simple [26] : les paquets ont une taille fixe et sont envoyés à un rythme continu, l'intervalle d'envoi entre deux paquets est constant. De plus, la source d'un message n'essaie pas de savoir si son paquet a bien été reçu.

Afin de créer un trafic dans le réseau, des paramètres spécifiques au trafic (*MANET Traffic Generation Parameters*) sont affectées à *10 nœuds* émetteurs. Chaque nœud émetteur est configuré pour créer un trafic constant (*CBR*) vers des destinations aléatoires, dés le début de la simulation et ce, jusqu'à la fin de cette dernière.

Dans ces simulations, la taille des paquets est constante toujours égale à 1280 octets et la fréquence d'envoi est de 10 paquets/secondes. Le débit envoyé pour chaque nœud émetteur source est donc égal à :

Débit envoyé = fréquence (Source) x taille (Paquet en octets) x 8bits

$= 10 \times 1280 \times 8$ *bit/sec*

$=$ ***100 kbits/sec***

Les paramètres de configuration du trafic de données dans réseau sont résumés dans le tableau 5.4 ci-dessous :

Paramètres de trafic	Valeurs
Nombre de connexion dans le réseau (*ou émetteurs*)	*10 flux*
Début de transmission des paquets (*Start Time, en seconde*)	*Début simulation (0s)*
Fin de transmission des paquets (*Stop Time, en seconde*)	*Fin simulation (300s)*
Durée entre deux paquets (*Inter-arrival Time, en seconde*)	*0,1 seconde*
Taille d'un paquet (*Packet Size, en octets*)	*1280 octets*
Adresse de destination (*Destination IP-Address*)	*Aléatoire*
Segmentation des paquets	*Désactivée*

Tableau 5.4 - Paramètres de trafic dans le réseau.

5.7 Résultats et interprétations

Nous allons commencer par présenter les résultats des performances apportées aux applications par le nouveau mécanisme de routage *AODV-SI* sous forme de graphes illustratifs.

Pour prendre une décision sur la qualité de ces performances, nous les comparons avec ceux du protocole de routage *AODV Standard* [23] qui est déjà implémenté dans *l'OPNET* [50]. Par la suite, nous allons révéler par le même biais, les surcoûts engendrés par le protocole *AODV-SI* par rapport au protocole *AODV Standard* [23].

5.7.1 Le nombre d'erreurs de route

Comme on l'a déjà mentionné précédemment, cette métrique représente le nombre de paquets d'erreurs de route envoyés par tout nœud du réseau dont l'accessibilité au prochain nœud n'a pas été pas validée par les messages « *Hello* », due à une coupure de route utilisée par au moins un trafic de données.

Les figures 5.10 représente la moyenne des paquets d'erreurs de route engendrée par les nœuds qui ont détecté une coupure des routes et dont la réparation locale n'a pas réussie à trouver un autre itinéraire.

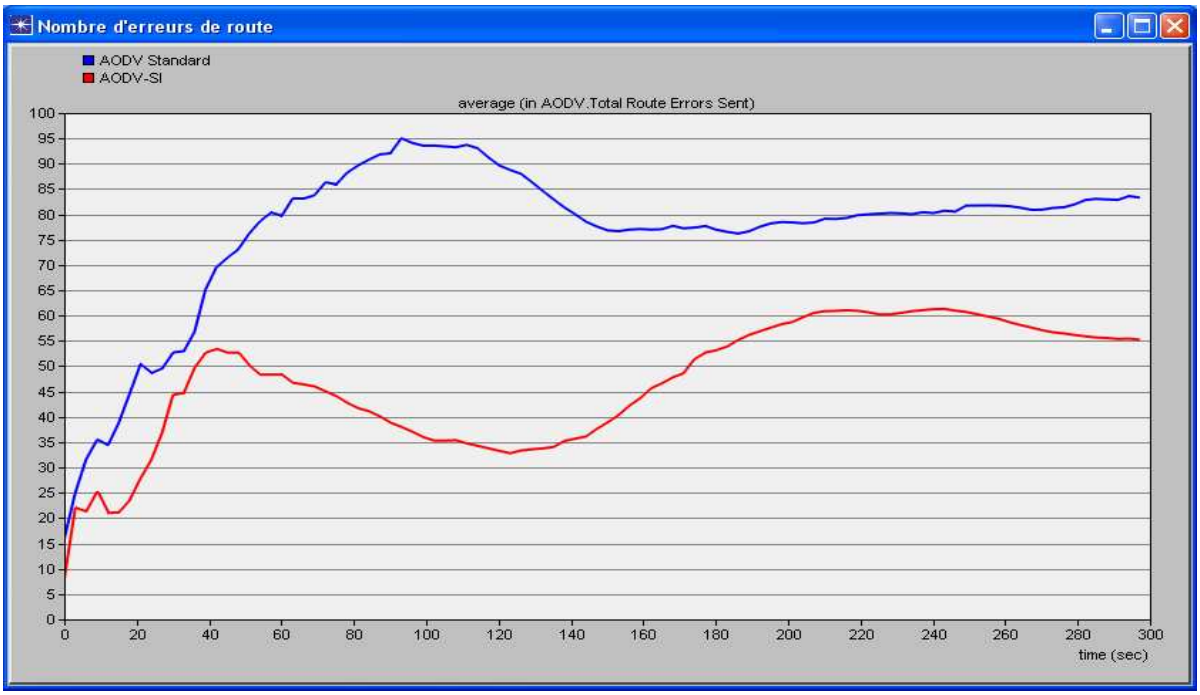


Figure 5.10 – Moyenne d'erreurs de route envoyées
avec *AODV* vs *AODV-SI*

Nous observons que les courbes illustrant le nombre moyen de paquets d'erreurs augmentent fortement au début pour les deux protocoles (*entre 0 et 40 secondes pour l'AODV-SI ; entre 0 et 100 secondes pour l'AODV Standard*) puis les variations sont moins brusques durant le reste du temps. Ceci peut être expliqué par le changement de la topologie. Quand les nœuds sont mobiles, ils génèrent plus de cassures de routes ce qui nécessite des phases de découverte de route répétées. Ces dernières engendrent des trafics de contrôle additionnels qui consomment plus de bande passante et laissent moins de capacité pour les trafics de données. L'*AODV-SI* affiche des résultats meilleurs que le protocole *AODV Standard* [23] durant toute la simulation.

La diminution considérable du nombre moyenne de paquets d'erreurs lors de utilisation du protocole de routage *AODV-SI* est justifiée premièrement par l'utilisation d'un mécanisme de sélection des routes stables, et deuxièmement par la prédiction de route permet de déclencher la découverte de route avant la rupture réelle du lien. Ainsi, les coupures d'itinéraires sont réduites par conséquent le nombre de paquets d'erreurs de route se voit diminué.

5.7.2 Le nombre de paquets de données reçus

Toujours pour le même scénario, on obtient les résultats présentés dans la figure 5.11, ci-dessous. Ils représentent l'évolution de la moyenne et du nombre de paquets de données reçues avec les protocoles *AODV Standard* [23] vs *AODV-SI* au cours du temps de la simulation.

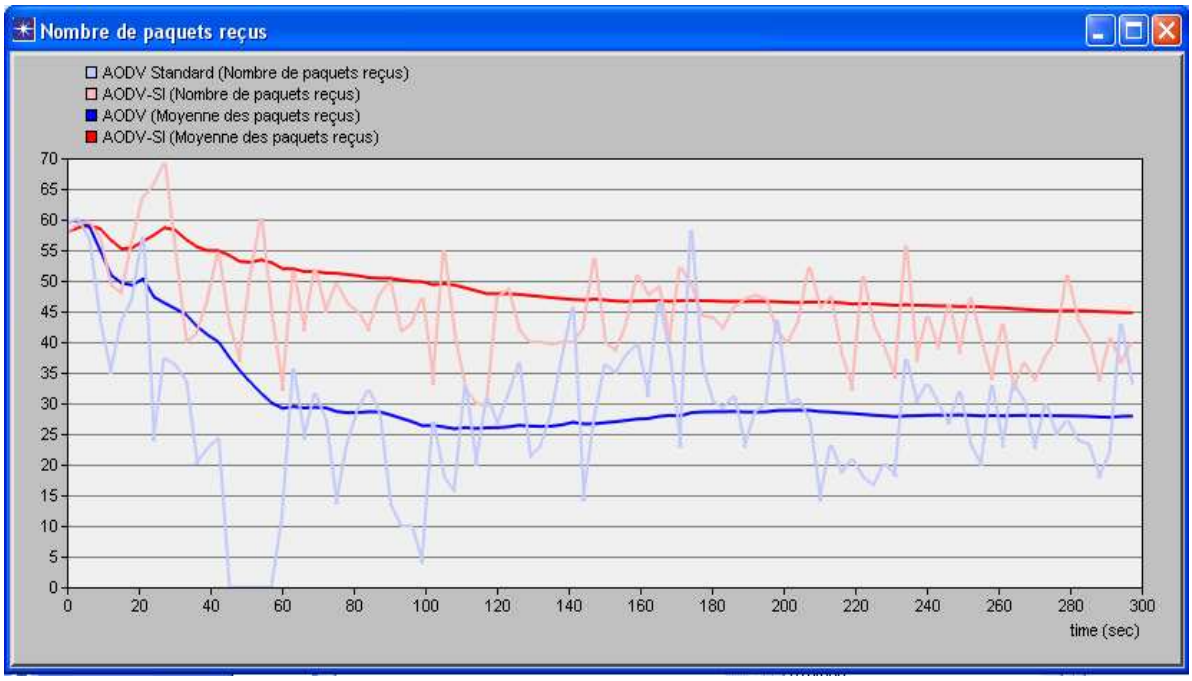


Figure 5.11 – Nombre et moyenne de paquets de données correctement reçus *avec AODV* vs *AODV-SI.*

Les graphiques illustrés par les courbes dont les lignes sont en couleurs foncées représentent les moyennes de paquets de données correctement délivrés avec les protocoles *AODV Standard* (bleu foncé) et *AODV-SI* (rouge foncé). Comme présenté dans la figure, les performances du protocole *AODV Standard* [23] se dégradent rapidement comparé au protocole *AODV-SI* qui affiche des meilleures performances.

Nous observons que l'*AODV-SI* offre une moyenne de réception de paquets de données qui commence à partir d'une moyenne de 60 et diminue jusqu'à une moyenne de 45 paquets sur un total de 100 paquets de données générés par seconde dans tout le réseau (donc 25% paquets sont perdus par rapport au début). Par contre, avec le protocole *AODV Standard* [23], la réception moyenne des paquets de données descend jusqu'à 26 paquets (donc 53% des paquets sont perdus par rapport au début). Ainsi, une amélioration de 28% du trafic reçu est constatée pour le protocole *AODV-SI* dans cette simulation.

Les graphiques illustrés par les courbes dont les lignes sont en couleurs estompées représentent le nombre totale des paquets de données correctement délivrés avec les protocoles *AODV* (bleu clair) et *AODV-SI* (rouge clair).

Nous observons qu'avec le protocole *AODV Standard* [23] les interruptions de connexions subies dans le réseau sont plus importantes et plus fréquentes qu'avec le protocole *AODV-SI*. Ainsi, on voit qu'avec l'*AODV Standard*, toutes les connexions s'arrêtent qu'à partir de 45ème et jusqu'à la 56ème seconde. Le nombre de paquets descend à zéro dans la courbe en bleu clair, alors qu'avec l'*AODV-SI* les connexions sont maintenues plus longtemps et les coupures sont moins fréquentes.

La dégradation de *l'AODV Standard* [23] est justifiée toujours par le choix des plus courts chemins sans aucune contrainte. En effet, un minimum de nœuds intermédiaires entre la source et la destination correspond à l'existante de liens plus longs (grandes distances) entre les nœuds intermédiaires, et comme la distance entre les nœuds est l'un des facteurs influant sur la qualité du canal, alors les liens longs ont une mauvaise qualité de réception et fonctionnent à bas débits.

5.7.3 Le nombre de réponses de route

La figure 5.12, ci-dessous, résume les résultats obtenus des réponses de route générées par les nœuds destinations avec le même scenario de simulation précédent.

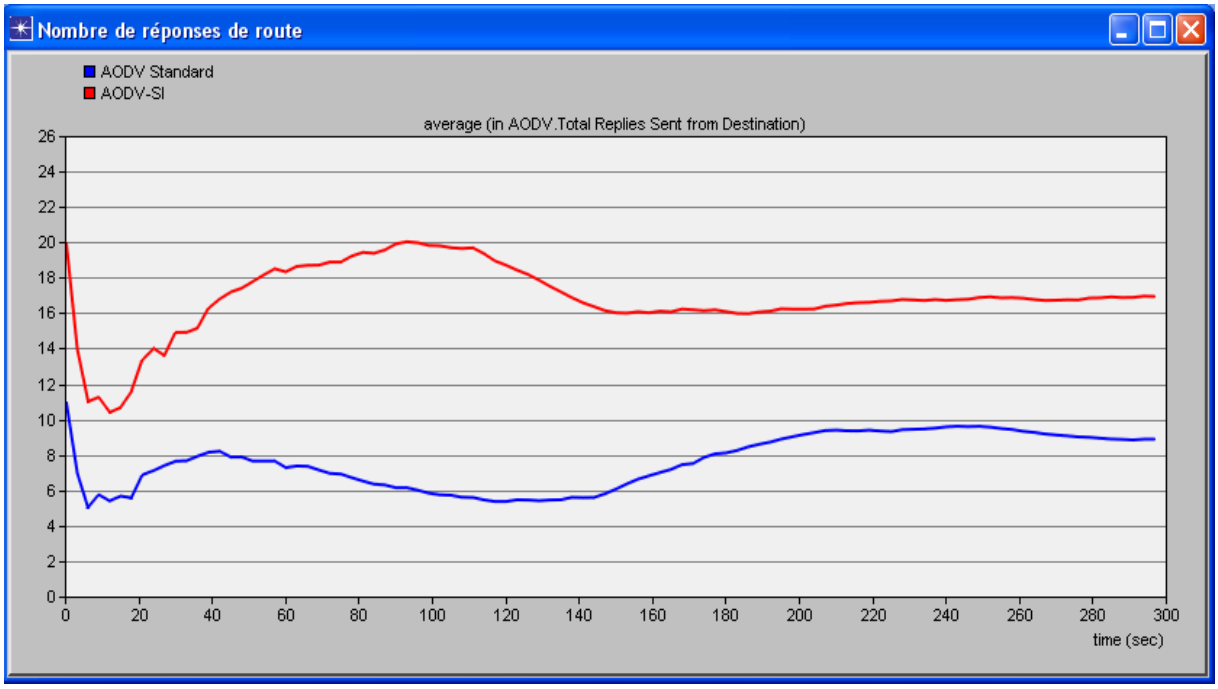


Figure 5.12 – Moyenne de réponses de route générée
avec *AODV* vs *AODV-SI*

Nous observons une évolution croissante du nombre de paquets de réponse de route générées dans le réseau avec le protocole *AODV-SI* en comparaison avec le protocole *AODV Standard* [23]. On constate que la moyenne des réponses de route est comprise entre *10* et *20* paquets pour l'*AODV-SI,* alors qu'elle commence à *5* et ne dépasse pas *10* paquets pour l'*AODV Standard.*

Cette évolution est justifiée par la découverte de plusieurs routes différentes entre la source et la destination soit une moyenne de deux routes par destination et qui, par la suite, permet à la source de sélectionner l'itinéraire le plus stable pour acheminer les paquets de données le plus longtemps possible.

L'*AODV Standard* [23] se contente d'une seule réponse de route (généralement, la première) pour commencer l'acheminement du trafic de source vers la destination. La route choisie correspond toujours au plus court chemin en nombre de sauts donc sans aucune exigence de qualité de service ce qui entraîne une dégradation de ses performances.

5.7.4 Le nombre de sauts par route

La figure 5.13, ci-dessous, réunit les résultats obtenus de la moyenne globale et de l'évolution de du nombre de sauts moyen par route reliant les nœuds sources aux nœuds destinations avec le même scenario de simulation précédent.

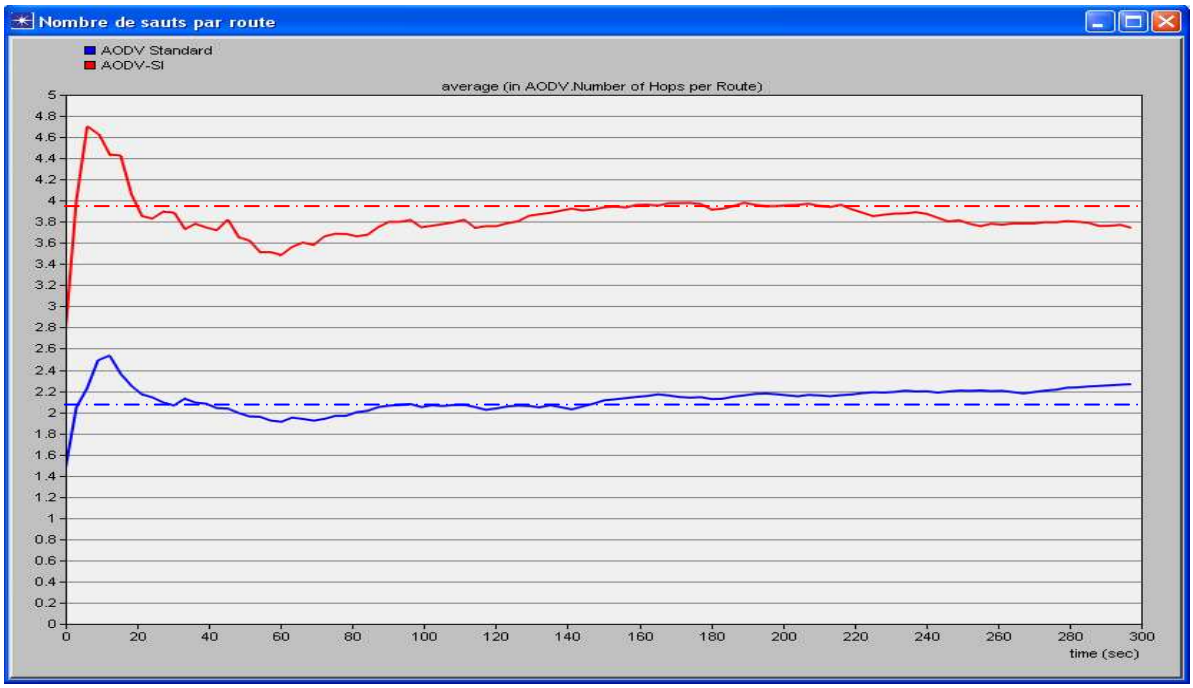


Figure 5.13 – Moyenne du nombre de sauts obtenue
avec *AODV* vs *AODV-SI*

Nous observons que l'*AODV-SI* utilise une moyenne globale de quatre sauts par route (trait en pointillés, rouge) alors que l'*AODV Standard* n'utilise que deux sauts par route (trait en pointillés, bleu). Ces résultats confirment que le protocole *AODV-SI* effectue le choix de route selon la contrainte de stabilité des liens, c'est-à-dire, les liens correspondant à de petites distances entre les nœuds intermédiaires de la source et la destination.

Cependant, l'*AODV Standard* [23] recherche le plus cours chemin en nombre de sauts entre la source et destination ce qui influe sur la qualité de réception des données et la durée de vie des liens dans le réseau.

5.7.5 Délai de découverte de route

La figure 5.14, ci-dessous, représente les résultats obtenus des délais de découverte de route effectués par les nœuds sources avec le même scenario de simulation précédent.

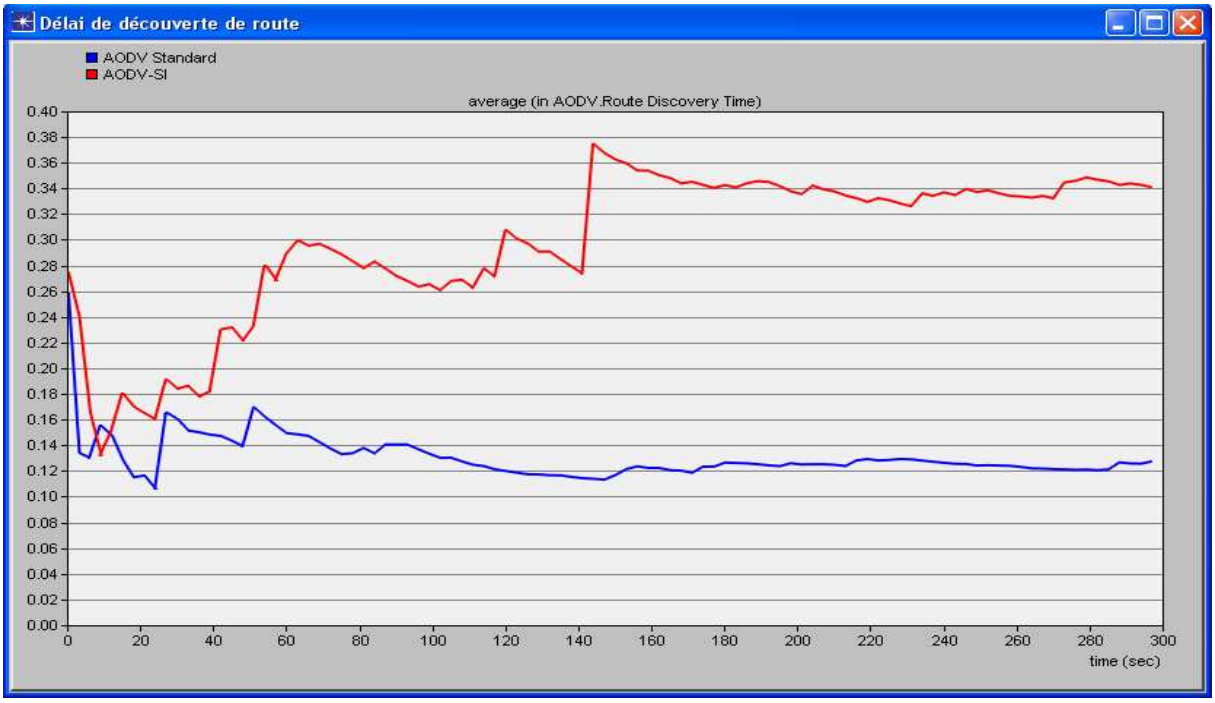


Figure 5.14 – Moyenne des délais de découverte de route obtenue avec *AODV* vs *AODV-SI*

Nous observons une augmentation importante des délais moyens de découverte de route dans le réseau avec le protocole *AODV-SI* en comparaison avec le protocole *AODV Standard* [23]. On constate que le protocole *AODV-SI* prend trois fois plus de temps que *l'AODV Standard* [23] pour découvrir les routes (pour *AODV ≈ 0,12 sec*, par contre pour *AODV-SI ≈ 0,34 sec*).

Ce grand délai peut être expliqué de deux manières, premièrement pour la même requête de découverte de route l'*AODV-SI* doit accepter toutes les réponses qui arrivent et deuxièmement le nombre de sauts constituant les itinéraires est important engendrant ainsi des délais d'attentes dans les files de tous les nœuds intermédiaires entre la source et destination.

5.7.6 Trafic de contrôle envoyé

La figure 5.15, ci-dessous, montre le nombre de paquets de contrôle générés par les deux protocoles de routage dans tout le réseau et avec le même scenario de simulation précédent.

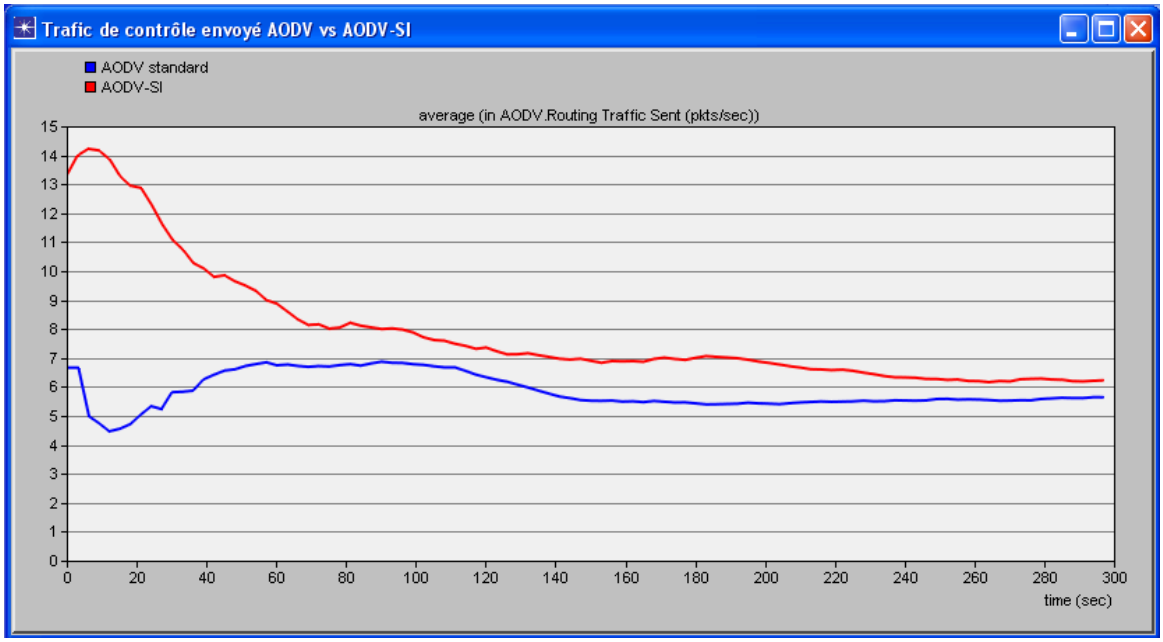


Figure 5.15 – Moyenne du trafic de contrôle envoyé
avec *AODV* vs *AODV-SI*

Au début du temps de simulation, nous observons un accroissement important des paquets de contrôle dans le réseau avec le protocole *AODV-SI* en comparaison avec le protocole *AODV Standard;* puis à partir de la 50ème seconde les deux protocoles génèrent presque la même charge de trafic de contrôle.

On peut commenter ces résultats par le fait que le nombre de requêtes de découvertes engendrées par l'*AODV-SI* parcourent plus de saut pour atteindre la destination (dans section 5.7.4, on a obtenu deux fois plus de sauts), ajoutant encore à ça le nombre multiple de réponses de route reçues par la source (dans la section 5.7.3, on a obtenu deux fois plus de réponses de route) et enfin il faut dire que les scénarios sont conçues et paramétrés de telle sorte que tous les trafics soient générées au même instant dans le réseau (au début de la simulation) .

Cependant, ce surcoût n'est généré que durant une petite période au début de tout le temps de la simulation puis il diminue suffisamment pour devenir presque égale à celui conçue par le protocole *AODV Standard* [23].

5.7.7 Délai de bout en bout des paquets

La figure 5.16, ci-dessous, représente le nombre de paquets de contrôle générés par les deux protocoles de routage dans tout le réseau et avec le même scenario de simulation précédent.

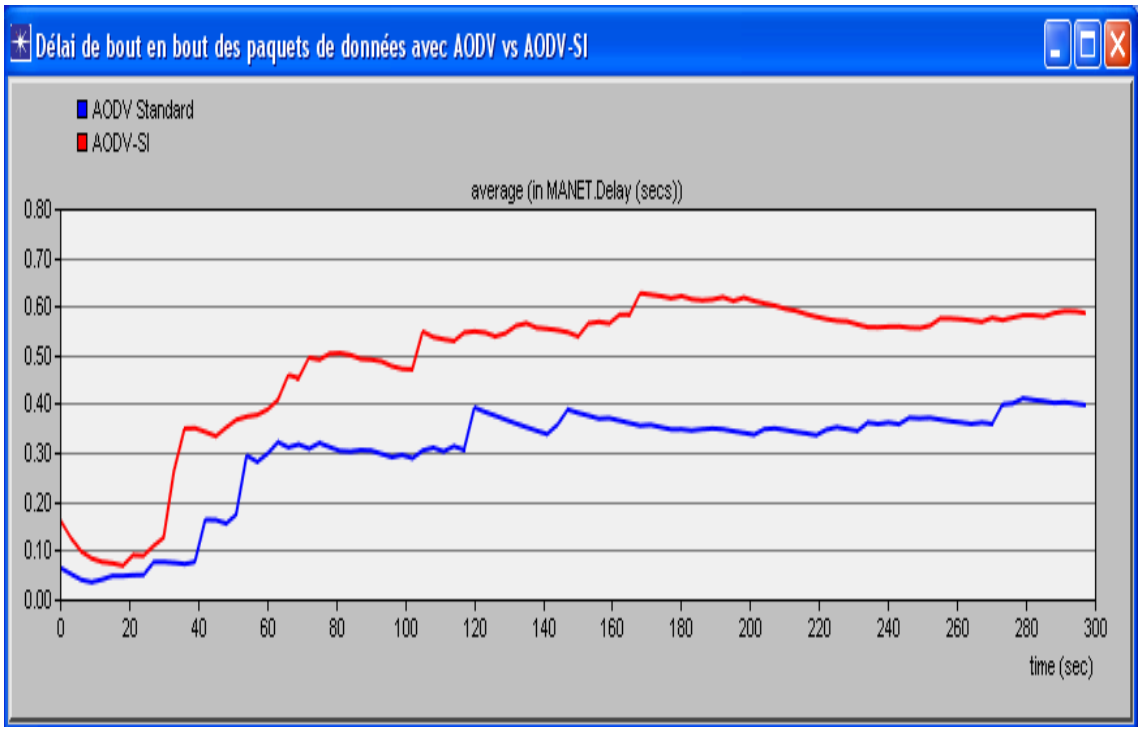


Figure 5.16 – Moyenne du délai de bout en bout obtenue avec *AODV* vs *AODV-SI*

Nous observons une faible augmentation de la durée de bout en bout des paquets de données dans tout le réseau du protocole *AODV-SI* en comparaison avec le protocole *AODV Standard* [23] pendant toute la période de la simulation.

Cette petite différence peut être commentée par le faite que les différents paquets de données acheminés par les différentes routes installées par l'*AODV-SI* parcourent plus de sauts pour atteindre la destination.

5.8 Conclusion

Les simulations réalisées dans ce présent chapitre nous ont montré les différences entre les deux protocoles *AODV Standard* [23] et *AODV-SI*. Ainsi, nous avons pu dégager les améliorations apportées par notre proposition et les surcoûts générés par celle-ci afin d'assurer une forte stabilité des itinéraires et par là, une grande fiabilité du réseau.

Nous concluons que la mise en œuvre d'un tel mécanisme de routage offre une importante qualité de service pour les applications qui exigent des liaisons durables et un trafic important dans le réseau.

Conclusion et perspectives

Conclusion et perspectives

Le réseau Ad-Hoc *(MANET)* consiste en un ensemble de nœuds autonomes, auto-organisables et auto-opérationnels. *MANET* [2] est caractérisé par des liens fragiles, des liens avec moins de bande passante, des nœuds avec des contraintes d'énergie, des nœuds avec moins de capacité mémoire et de puissance de traitement que dans les réseaux filaires. Malgré toutes ces contraintes, il a plusieurs avantages et de multiples domaines d'applications. Ses utilisations ou applications spécifiques sont différentes que celles des réseaux fixes, et parfois même impossibles à réaliser avec ces derniers. Plusieurs solutions de routage ont été proposées pour ces réseaux mais aucune d'entre elles ne paraît satisfaire la diversité des exigences de qualité de service (*QoS*), dans ces réseaux.

Dans ce mémoire nous avons traité l'un des problèmes de routage avec qualité de service des réseaux sans fil Ad-Hoc, soit, le problème lié à la contrainte de stabilité des liens entre deux nœuds mobiles. Après avoir arboré l'évolution des réseaux sans fil et tout particulièrement le fonctionnement des réseaux Ad-Hoc, par la suite nous avons identifié les problèmes liés aux réseaux Ad-Hoc tout en explicitant les principaux mécanismes de la norme *IEEE 802.11* usités, pour l'accès au support de transmission.

Ce travail nous a permis de cerner un certain nombre de protocoles de routage multi-sauts, classés selon quatre familles principales conformément à leurs stratégies de routage et leurs modes de fonctionnement.

Par la suite, nous avons étudié les différentes catégories de solutions de qualité de service dans *MANET,* soient *:* les protocoles de signalisation, les protocoles de routage avec *QoS*, la différenciation des services et enfin les modèles de *QoS.*

Ainsi, nous nous sommes inspirés de l'architecture routage avec qualité de service (*QoS*) pour définir un de nos objectifs qui consiste à trouver une solution pour améliorer la qualité de service en termes de stabilité des itinéraires dans le réseau.

Notre contribution n'est autre qu'une suggestion d'une solution appelée : *Protocole AODV-SI.* Cette solution est basée sur une extension du protocole de routage classique *AODV* qui incorpore un premier mécanisme de contrôle d'admission prenant en compte des informations de la couche physique (puissance du signal de réception) pour la sélection des liens stables lors du processus de découverte et d'installation de route, suivi d'un second mécanisme de prédiction de rupture de routes lors de la maintenance de celles-ci.

Nous avons évalué et comparé le protocole *AODV-SI* avec le protocole *AODV* standard, en utilisant le simulateur de réseau *OPNET Modeler*. Les résultats obtenus montrent une amélioration considérable des performances du réseau (total d'erreurs de route, volume du trafic délivré, total de réponses de route) par rapport à l'*AODV* classique. On peut conclure que, malgré les quantités importantes de trafics de contrôles générés, ce protocole constitue une bonne solution pour la sélection des itinéraires stables et durables dans le réseau.

Dans le futur, on se propose d'intégrer dans le protocole *AODV* une réservation effective de la bande passante, qui se basera sur la dissémination des messages « hello » à deux sauts et de changer le mécanisme de maintenance des routes comme déjà expliqué. Il va falloir songer également à surveiller la garantie de la qualité de service sur les routes actives en envoyant des messages d'erreur suite à la dégradation de la qualité de service sur ces routes. Nous suggérons d'étudier les résultats de simulation des réseaux dans lesquels circulent des trafics privilégies et des trafics dits « best effort » afin de réguler la consommation de la bande passante entre les deux types de trafics afin de permettre aux applications d'être adaptatives en changeant leur débit en fonction de l'état du réseau.

Sur le plan de la sécurité, des axes de recherche intéressants, peuvent être entrevus dans le futur : inclure un certains nombre de critères de sécurité et de gestion des ressources en énergie lors de la sélection des routes pour la transmission des données, comme prolongation à la solution *AODV-SI,* développée sera sans aucun doute d'un apport de *QoS* irréfutable.

Bibliographie

[1] M. Rahnema. « *Overview of the gsm system and protocol architecture* ». *IEEE Communications Magazine*, 31(4) :92–100, Avril 1993.

[2] "*IETF, Mobile Ad hoc Network (manet)*". www.ietf.org/html.charters/manetcharter.html.

[3] S.Corson and J.Macker. "*Mobile Ad-Hoc Networking (MANET): Routing Protocol Performance Issues and Evolution Considerations*", *RFC 2501*, Janvier 1999.

[4] G. Zussman and A. Segall. "*Energy efficient routing in Ad-Hoc disaster recovery networks*", In *Proceedings of IEEE INFOCOM*, San Francisco, USA, 2003

[5] A. Gallais, F. Ingelrest, J. Carle and D. Simplot-Ryl. "*Preserving Area Coverage in Sensor Networks with a Realistic Physical Layer*". *In Proc. 26th Annual IEEE Conference on Computer Communications (INFOCOM 2007),* Anchorage, Alaska, 2007.

[6] ANSI/IEEE std 802.11, "*Wireless LAN Medium Access Control (MAC) and physical Layer Specifications*," tech. rep., 1999.

[7] Andrew Tanenbaum. Réseaux. Pearson Education, 4th edition, 2003.

[8] Tahiry Razafindralambo and Fabrice Valois, "*Stochastic study off back off algorithm in case of hidden terminals"*. In *IEEE Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC),* 2006

[9] Chun-Yen Hsu; Wu, J.-L.C.; Shun-Te Wang. « *Finding stable routes in mobile ad hoc networks* ». 18th International Conférence on Advanced Information Networking and Applications, 2004. AINA 2004.

[10] Rabah MERAIHI. « *Gestion de la qualité de service et contrôle de topologie dans les réseaux ad hoc* » thèse de doctorat, EST, France, janvier 2005

[11] C. Hedrick, *«Routing Information Protocol», RFC 1058*, Jun 1988

[12] J. Moy. "*Open Shortest Path First v2 (OSPF)*". *RFC 2328*, Avril 1998.

[13] C. Perkins, P. Bhagwat, "*Highly dynamic destination-sequenced-distance vector routing (DSDV) for mobile computers," ACM SIGCOMM*, vol. 24, no. 4, October 1994.

[14] C. Siva Ram Murthy and B. S. Manoj, *"Ad-Hoc Wireless Networks Architecture and Protocols*", pages 299-359. Prentice Hall, 2004.

[15] Tsu-Wei Chen and Mario Gerla. "*Global State Routing: A new routing scheme for Ad-Hoc wireless networks". Proceeding IEEE ICC'98*, Jun 1998.

[16] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. *"Fisheye state routing in Mobile Ad-Hoc networks"*. *In Proceedings of the 2000 ICDCS Workshops*, Taiwan, Avril 2000

[17] G. Pei and al. *"A Wireless hierarchical Routing Protocol with group mobility", Proc. IEEE WCNC 99,* New Orleans, LA, Septembre 1999.

[18] T. H. Clausen, G. Hansen, L. Christensen, and G. Behrmann, "*The Optimized Link State Routing Protocol, Evaluation Through Experiments and Simulation," Proceedings of IEEE Symposium on Wireless Personal Mobile Communications 2001*, Septembre 2001

[19] D.B. Johnson and D. A. Maltz, "*Dynamic Source Routing in Ad-Hoc Wireless Networks" (DSR) Mobile Computing, Kluwer Academic Publishers*, vol. 353, pp. 153-181, 1996.

[20] V. Park and M. Corson, "*A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," Proceedings of IEEE InfoCom*, pp. 1405-1413, Avril 1997.

[21] C-K. Toh, "*Associativity-Based Routing for Ad-Hoc Mobile Networks",(ABR), Wireless Personal Communications Journal*, vol. 4, no. 2, pp. 1-36, Mars 1997.

[22] R. Dube, C. D. Rais, K. Y. Wang, and S. K. Tripathi, "*Signal Stability-Based Adaptive Routing for Ad-Hoc Mobile Networks", (SSA), IEEE Personal Communications Magazine*, pp. 36-45, Février 1997.

[23] C. Perkins, E. Royer, and S. Das. "*Ad-Hoc On-demand Distance Vector routing (AODV)", Internet Draft, Internet Engineering Task Force*, Mars. 2001.

[24] Zygmunt J. Hass, Marc R. Pearlman and Prince Samar, "*The Zone Routing Protocol (ZRP) for Ad-Hoc Mobile Networks", Internet Draft, IEEE*, Juillet 2002.

[25] Raghupathy Sivakumar, Prasun Sinha and Vaduvur Baharghavan. "*Core Extraction Distributed Ad-Hoc Routing CEDAR", IEEE Infocom, pages 202-209*, Mars 1999.

[26] M. Jiang, J. Li and Y.C. Tay, "*Cluster Based Routing Protocol (CBRP)", Draft-ietf-manetcbrp-spec-01.txt*, Juillet 1999.

[27] M. Joa-Ng and I. Lu, "*A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad-Hoc Network" IEEE Journal on Selected Areas in Communications, vol. 17, no 8, pp. 1415-1425*, Aout 1999.

[28] Y. Ko and N. H. Vaidya, "*Location-Aided Routing (LAR) in Mobile Ad-Hoc Network," Proceedings of ACM MobiCom 1998, pp. 66-75*, Octobre 1998.

[29] S. Basagni, I. Chlamtac, B. A. Woodward, et al. "*A distance routing effect algorithm* for *mobility (DREAM)", In Proc. of ACM/IEEE Mobicom*, 1998.

[30] R. E. Bellman. "*On routing problem*". *Quarterly of Applied Mathematics*, 16(1), 1958.

[31] R. E. Bellman. « *The shortest path through a maze* ». *In Proc. of the International Symposium on the theory of switching*, 1959.

[32] G. Malkin. "*RIP version 2", IETF STD 56*, http://ietf.org, Novembre 1998.

[33] Shree Murthy and J. J. Garcia-Luna-Aceves. *"A routing protocol for packet radio networks". Proceeding of the IEEE Mobicom, pp 86-95,* Novembre 1995.

[34] James A.Freebersyser and Barry Leiner. *« A DoD Perspective on Mobile Ad-Hoc Networks »*. *Ad-Hoc Networking*, Addison Wesley, London 2001.

[35] QoS Forum. « *QoS protocols and architectures* ». White paper of QoS Forum, Juillet 1999. http://www.qosforum.com.

[36] S.B. Lee, and A. T. Campbell. « *INSIGNIA : In-band signaling support for QoS in mobile ad hoc networks* ». In 5th Int.Workshop on Mobile Multimedia Communication (MoMuc'98), Berlin, Allemagne, Octobre 1998.

[37] R. Sedgewick. *"Weighted graphs". Chapter 31, Addision Weseley*, 1983

[38] L. Kleinrock and K. Stevens. *"Fisheye : Alenslike computer display transformation". Technical report, UCLA, Computer Science Departement*, 1971.

[39] C. C. Chiang, H-K Wu, Winston Liu, and Mario Gerla. *"Routing in clustered multihop mobile wireless networks with Fading Channel", Proceedings of IEEE SICON 1997,* Avril 1997

[40] M. Gerla and J. Tsai. *"Multicluster, mobile, multimedia radio network". ACM-Baltaz Journal of Wireless Networks*, 1995.

[41] H. Xiao, K. G. Seah, A. Lo, and K. C. Chua, « *A flexible quality of service model for mobile ad-hoc networks* », *Vehicular Technology Conference Proceedings, IEEE* : 51st, Tokyo, 2000.

[42] Cheikh Sarr, Claude Chaudet, Guillaume Chelius, Isabelle Guérin Lassous, *« Bandwidth Estimation for IEEE 802.11-Based Ad Hoc Networks »*. IEEE Transactions on Mobile Computing, Octobre 2008.

[43] Lei Chen and Wendi B. Heinzelman, «*A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks*», *IEEE Network*, University de Rochester, USA Décembre 2007.

[44] V. D. Park and M. S. Corson. "*Temporally-ordered routing algorithme (TORA) Version1, functionnal specification". IETF, Internet Draft, draft-ietf-manet-tora-spec-02.txt,* Octobre 1999.

[45] A. Moussaoui, « *Routage QoS et Prédiction de Rupture de Route dans les Réseaux Ad-Hoc* », Mémoire de magistère, U.A.M.B, Bejaïa, Juillet 2006.

[46] A. Kamerman and L. Monteban, "*WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band*," Bell Labs Technical Journal, Juillet 1997.

[47] J.D. Parsons. « *The Mobile Radio Propagation Channel* ». Jhon Wiley and Sons Ltd, 2sd edition, 2000.

[48] Z.J. Haas, M. Perlman and P. Samar, "*The Inter zone Routing Protocol (IERP) for Ad-Hoc Networks." draft-ietf-manet-zone-ierp-01.txt, IETF MANET Working Group*, Juin 2001.

[49] C. Perkins and E. Belding-Royer. « *Quality of Service for Ad-Hoc On-Demand Distance Vector Routing* », (work in progress), draft-perkins-manet-*AODV*qos-02.txt. Octobre 2003.

[50] *AODV Contributed Model (NIST)*, OPNET Technologies Inc, www.opnet.com

[51] A. G. Longley, P. Rice. "*Prediction of Tropospheric Radio Transmission Loss Over Irregular Terrain, A Computer Method-1968*", *ESSA Technical Report ERL 79-ITS 67* Institute for Telecommunications Sciences. 1968.

[52] M. Sanchez and P. Manzoni. Anejos: « *Future Generation Computer Systems* », 17(5):573–583, 2001.

[53] E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, « *A framework for QoS-based routing in the Internet RFC 2386 », The Internet Engineering Task Force*, Août 1998]

[55] Dominique Dhoutaut, « *Etude du standard IEEE 802.11 dans le cadre des réseaux ad hoc: de la simulation à l'expérimentation* », INSA, Lyon, Décembre 2003.

[56] R. Braden, et al. « *Resource ReSerVation Protocol (RSVP) », RFC 2205, IETF*, Septembre 1997.

[57] R. Braden, D. Clark, and S. Shenker. «*Integrated services in the internet architecture*» : *an overview*. *RFC 1633, IETF,* Juin 1994. ftp://ftp.nordu.net/rfc/rfc1633.txt.

[58] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. « *An architecture for differentiated services*». *RFC 2475, IETF,* Décembre 1998. ftp://ftp.nordu.net/rfc/rfc2475.txt.

[59] I. Aad, C. Castelluccia, « *Differentiation mechanisms for IEEE 802.11* », *IEEE Infocom 2001*, Anchorage - Alaska, Avril 2001.

Résumé

Un réseau Ad-Hoc est un mode de fonctionnement particulier des réseaux sans fil, c'est donc une collection de nœuds mobiles à topologie variable et communicants entre eux sans point d'accès et sans administration centralisée. Ce type de réseaux est très facile à déployer et leurs utilisations semblent beaucoup plus étendues. Les communications multi-sauts y sont possibles grâce à des protocoles de routage spécifiques. Cependant, le processus de routage devient une tache complexe dans le cas ou certaines contraintes de qualité de services sont rajoutées au niveau des transmissions.

Nous avons étudié différents protocoles de routage, pour améliorer le routage dans les réseaux *Ad-Hoc* et afin de prendre en compte les contraintes stabilité des liens. Finalement, nous avons choisi d'utiliser le protocole de routage *AODV* (*Ad-hoc On-Demand Distance Vector*) car il présente un nombre important de caractéristiques qui s'adaptent aux réseaux sans fil.

Suite à une modification de ce protocole, plusieurs optimisations ont été apportées pour améliorer ses performances. *AODV* adopte la métrique de nombres de sauts qui n'est pas toujours la solution optimale, ainsi la route à tendance à contenir des liens fragiles et offre de faibles performances. Pour résoudre ce problème, on a utilisé la puissance du signal reçu pour la sélection d'itinéraires et pour la prédiction des ruptures de liens. En préférant les liens avec une plus grande puissance du signal, alors les chemins convergent progressivement à l'optimum, donc l'itinéraire sélectionné possède une forte stabilité et une plus grande durée de vie. *L'AODV* modifié est testé sous le simulateur « *OPNET Modeler »* qui stipule parfaitement bien que ce dernier apporte une qualité de service indéniable par rapport à *l'AODV* original.

Mots-clés : Réseaux Ad-Hoc, protocoles de routage, stabilité des liens, contrôle d'admission, qualité de service (*QoS).*

Abstract

An Ad-Hoc network is a particular mode of operation of wireless networks, so it's a collection of mobile nodes in variable topology and communicating with each other without access point and without centralized administration. This type of network is very easy to deploy and use seem to be much broader. Communications multi hops are possible through specific routing protocols. However, the routing process becomes a complex task in case of certain quality of service are added at the transmission.

We have study different routing protocols to improve the routing in the Ad-Hoc networks and to take care of the links stability constraints. Finally, we have chosen the use *AODV* (*Ad-hoc On-Demand Distance Vector*) routing protocol because it has a large number of features that adapt to wireless networks (*MANET*).

In this work, the protocol was changed and several optimizations have been added to improve its performance. *AODV* adopts the metric of the number of hops which is not always the optimal solution, and the road tends to contain links fragile and offers poor performance. To resolve this problem, we used the signal received power for the selection of routes and the prediction of links failures. In using the links with greater strength of the signal, then the paths gradually converge to the optimum, so the selected route has a high stability and greater durability. We evaluated the modified *AODV* with the « *OPNET Modeler »* simulator and found that it takes a great improvement and quality of service compared to the original *AODV*.

Keywords: Ad-Hoc networks, routing protocols, links stability, admission control, quality of service(*QoS*).