Tables de matières

Liste des figures

Liste des tableaux

Introduction générale

On assiste ces dernières années à une importante évolution dans le domaine de la communication et de l'information, conduite par l'émergence des appareils de communications (téléphones cellulaires, PC portables, PDA, . . . etc.) et la complémentarité entre réseaux fixes et mobiles, ce qui a permis de rendre l'information accessible n'importe où et n'importe quand d'une manière très simple et avec un coût modeste.

Les réseaux sans fil ont connu une véritable explosion et un grand succès depuis la fin des années 90. Ces réseaux se différentient de ceux du monde filaire par leurs couches basses (physiques et liaisons), leurs support de transmission à base de signal radio et par la topologie très dynamique suite à la mobilité des éléments qui les composent.

Plusieurs types de réseaux sans fil ont été développés tels que les réseaux téléphoniques cellulaires, les réseaux Bluetooth, les réseaux locaux sans fil (WLAN), et les réseaux Ad hoc mobiles.

Les réseaux mobiles Ad hoc ou MANET (Mobile Ad hoc NETwork) constituent l'un des nouveaux champs de recherche les plus actifs du domaine des télécommunications. Les recherches dans ce domaine ont pris un grand essor avec l'arrivée des premières technologies radio, principalement la norme IEEE 802.11 et ses différentes dérivées. Cette norme a été standardisée en 1999 par l'IEEE (Institute of Electrical and Electronics Engineers), dans le but d'assurer la communication entre ordinateurs utilisant le médium radio.

Les réseaux Ad hoc sont constitués de stations (nœuds) mobiles reliées par des liens sans fil, équipées de cartes d'interface radio et des couches protocolaires adéquates pour communiquer entre elles si elles sont situées à la portée radio (i.e. rayon de communication par ondes radioélectriques ou ondes hertziennes). La portée des stations étant relativement limitée, le déploiement d'un réseau à grande échelle nécessite que le réseau MANET soit multi-sauts, c'est-à-dire que des stations intermédiaires fassent office de point de relais (routeurs). Les réseaux MANETs, grâce à leur auto-organisation, leur autonomie, et à l'absence d'infrastructure, peuvent facilement être déployés dans de nombreux domaines (opérations de secours, opérations militaires, enseignement ou les systèmes embarqués). Cependant, ils restent limités par différentes contraintes telles que la largeur de bande du support partagé, l'asymétrie de ses liens, le délai, l'autonomie et la capacité de ses nœuds, la mobilité, etc.

Ces dernières années, les recherches dans les réseaux Ad hoc, ont porté sur l'étude de leurs architectures, la qualité des services qu'ils offrent, la gestion de la mobilité, la transmission de données multimédia (voix, audio, vidéo), la modélisation du trafic, l'auto configuration des nœuds, le routage, etc. Parmi ces axes, la qualité de service et le routage constituent ceux qui soulèvent le plus de défis.

Les progrès des technologies numériques et de l'Internet ont fait émerger de nouvelles applications de multimédia et de temps-réel (vidéoconférence, téléphonie sur internet, vidéo sur demande…) qui posent de nouvelles contraintes au service de communication. La Qualité de Service (QdS) est au cœur de ces nouvelles demandes. Néanmoins, les ressources limitées des réseaux ad hoc rendent complexes le support de telles applications qui nécessitent des ressources importantes, car le réseau dans ce cas, doit optimiser un ensemble de paramètres, tel que le délai, la gigue, la bande passante, le taux de livraison de paquet, le taux de perte, etc. En plus, la nature partagée du canal de communication (air) et la mobilité des nœuds dans ces réseaux, leur organisation distribuée, le problème des nœuds cachés /exposés et la topologie dynamique causent la perte de paquets et influent sur la fiabilité des communications. Ceci apporte des défis supplémentaires pour supporter des services multimédia qui nécessitent une QdS et rendent sa garantie très délicate dans les réseaux ad hoc mobile.

La QdS dans les réseaux Ad hoc est actuellement faible et nécessite d'être améliorée. La plupart des solutions développées pour les réseaux filaires ne sont pas directement convertibles aux réseaux Ad hoc au vu des différences entre les deux types de réseaux.

De manière très simple, l'idée de la QdS, est d'offrir mieux que le service best effort de base afin que les applications ayant des exigences spécifiques fonctionnent correctement.

Le terme QdS peut regrouper une multitude de concepts distincts. Dans le domaine des réseaux, c'est la capacité à fournir un service adapté aux besoins spécifiques des applications (i.e. adapter le comportement du réseau aux besoins des applications).

D'un point de vue technique, la QdS proposée pour une application est caractérisée par un ensemble de paramètres, tel que le débit ou bande passante disponible (i.e. doit être suffisante pour absorber tout le trafic), le délai ou temps de réponse (i.e. le temps que requiert un paquet pour passer d'un bout à l'autre), la gigue (i.e. la variation maximale des temps de réponse des paquets dans le réseau), le taux de perte des paquets (i.e. le rapport entre le nombre de paquets émis et ceux reçus correctement) et la disponibilité du service rendu à tout instant.

La bande passante et le délai sont généralement les deux paramètres adoptés comme critère de QdS pour ce nouvelles applications (multimédia et temps réel).

La première proposition de QdS date de 1979 et le modèle à services intégrés IntServ fût le premier modèle de QdS. Après, et pour combler certains lacunes de l'IntServ, un modèle à différentiation de services DiffServ à vu le jour.

Plusieurs travaux pour le support de la QdS dans les réseaux Ad hoc ont été proposés. Ils sont globalement classés par couche de protocole, et se focalisent essentiellement sur des problèmes liés à une couche particulière indépendamment des autres couches.

Actuellement, il n'y a pas encore de protocole standard de QdS adapté aux spécificités du MANET, malgré tous les travaux qui ont été effectués.

Notre contribution dans cette thèse tente d'améliorer la QdS dans les réseaux Ad hoc. Pour cela, notre premier objectif a été l'étude de ces différents mécanismes de QdS, que ce soit dans les réseaux filaires ou dans les réseaux Ad hoc pour bien situer leurs carences. Ce qui nous permettra par la suite d'offrir des solutions optimales à des applications sensibles à certains facteurs de QdS.

Notre première contribution se situe au niveau de la couche réseau et porte spécialement sur la modification au protocole de routage pour supporter la qualité de service selon les différents paramètres. La seconde contribution porte sur la procédure DCF (Distributed Coordination Function) de la norme IEEE 802.11 au niveau de la couche MAC.

La fonction de base d'un protocole de routage est de déterminer une ou plusieurs routes entre deux nœuds désirant communiquer. Le protocole de routage dissémine des informations de routage nécessaires à l'obtention et à la maintenance des routes. Suivant le type de dissémination de l'information, ces protocoles peuvent être répertoriés en cinq grandes classes : proactifs, réactifs, hybrides, hiérarchiques et géographiques.

En 2003, le groupe de travail MANET a retenus quatre protocoles seulement pour une normalisation : Optimized Link-State Routing (OLSR), Ad Hoc On Demand Distance Vector (AODV), Dynamic Source Routing (DSR) et Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). OLSR, AODV et TBRPF ont d'ores et déjà franchi une étape vers la normalisation

L'étude de ces différentes approches nous a permis d'orienter nos travaux sur un protocole de routage réactif en l'occurrence l'AODV. Les routes déterminées par ce protocole ne répondent à aucunes exigences de QdS.

Pour apporter de la QdS à une route, le protocole de routage utilise une fonction poids dépendante de trois facteurs (la capacité d'un lien, la bande passante disponible et le nombre de voisins).

Nous avons choisi de baser nos contributions sur l'amélioration du protocole de routage AODV selon une série de modifications touchant en premier lieu, la réduction de la charge de contrôle par élimination des informations non nécessaires et limiter l'espace dans lequel ces informations doivent être routées. Le second point portera sur quelle stratégie doit on adopter lors de la rupture d'un lien d'une route utilisée durant une phase de transfert de données.

1. Augmenter la taille de la file d'attente associée au nœud pour recevoir le maximum de paquets et éviter leurs pertes en attendant que la réparation du lien cassé soit faite. Le problème ici est comment choisir cette taille par rapport au temps mis par l'opération de réparation. Il faut noter que la capacité fait défaut aux nœuds Ad hoc.
2. Comment aviser le plus tôt la source par émission d'un paquet spécial (erreur temporaire) pour qu'elle arrête ses transmissions.
3. Est-ce que une réparation à la source (AODV-SR) apporte une amélioration tout en minimisant le temps mis pour aviser la source par le nœud ayant détecté la rupture (compromis délai /perte).

Vu que les nœuds des réseaux Ad hoc sont mobiles, les déconnexions sont fréquentes. Une autre solution est de prévoir plus d'une route pour une destination donnée ou des multi chemins (M-AODV). Le problème lié

à cette solution est de répondre à la question comment trouver un compromis entre la disponibilité de routes et le volume considérable des paquets de contrôle nécessaire pour leur maintien et qui consomme de la bande passante (compromis débit/perte).

La dernière proposition dans le contexte du routage et qui consiste spécifiquement en la modification faite sur l'AODV est de lancer en parallèle à l'opération de transfert de données, une action de prédiction sur les positions des nœuds en se basant sur la qualité du signal. Cette action permet de nous renseigner si un nœud est proche, loin ou s'il n'est plus dans le rayon de portée. Cette technique est utilisée pour prévoir à l'avance de nouveaux liens plus stables que ceux faisant partie de la route ou il y a une forte probabilité de déconnexion (PF-AODV).

Les mécanismes de la norme IEEE 802.11 rencontrent un grand défi pour le support de la QdS puisque la DFC telle que définit actuellement ne peut supporter une différentiation pour les différentes classes de trafic et d'un autre coté la PCF peut fournir seulement un support de QdS pour trafic temps réel. Par conséquent la norme IEEE 802.11 est conçue pour un service best effort et non pour les applications de multimédia avec des exigences de QdS.

Le protocole de routage avec QdS, détermine les routes qui répondent aux exigences de QdS telles que le délai et la bande passante sans assurer leurs réservations. Dans une méthode d'accès au support avec contention comme CSMA/CA, la réservation est très difficile à cause de la présence de collisions. Notre seconde contribution à ce niveau pour un cas avec contention est d'améliorer la procédure d'accès au médium DCF en proposant une nouvelle forme d'incrémentation du Backoff pour trouver les meilleures valeurs de la fenêtre de contention (CW) pour minimiser le nombre de collisions et par conséquent les retransmissions qui réduisent la bande passante utile d'un réseau et accroissent, aussi, le délai de transfert des paquets de données, pour mieux permettre le respect des contraintes de QdS.

Cette thèse est structurée en deux parties (l'état de l'art et la contribution) composée chacune de trois chapitres. Les réseaux locaux sans fil et les différents concepts liés à ce type de réseaux, la norme IEEE 802.11 et les réseaux mobiles Ad hoc sont décrit dans le premier chapitre. Le second chapitre traite les protocoles de routage Meilleur Effort existant dans le contexte Ad hoc avec leur classification. Le troisième chapitre présente la notion de QdS dans les réseaux Ad hoc. Dans ce chapitre nous présenterons la définition de la QdS et les modèles de QdS pour les réseaux Ad hoc, les protocoles de routage avec QdS, et enfin un bilan sur les travaux récents de QdS pour MANET. Le quatrième chapitre décrit le protocole de routage AODV sur lequel nos contributions se sont basées. Le cinquième chapitre est consacré à la description de notre apport dans cette thèse. Dans ce chapitre, on décrit les différentes modifications faites sur l'AODV au niveau routage puis la proposition faite dans la couche MAC. La validation des différentes propositions par simulation sous le simulateur NS2 fera l'objet du chapitre six.

Pour finir ce travail, nous dressons une conclusion sur les travaux présentés dans ce manuscrit ainsi que les perspectives et les orientations pour la poursuite de ce travail de recherche.

Partie 1 : Etat de l'art

Chapitre 1 : Les environnements sans fil

I.1. Introduction

Les premières réussites de communication sans fil ont vue le jour vers les années 90 avec les premiers téléphones dotés de combinés sans fil caractérisés par une autonomie très limitée et une faible portée de communication. Cependant, la véritable révolution des réseaux de données sans fil a eu lieu au début des années 2000 avec le développement de la norme 802.11 (a, b, g, etc..) offrant des débits et une qualité de services pas loin des réseaux filaires traditionnels. Les premières normes (802.11a et 802.11g) proposent des débits allant jusqu'à 54 Mbit/s tandis que les dernières évolutions (802.11n) projette de dépasser le seuil de 500 Mbit/s. Grâce à cette évolution, ces réseaux sans fil se sont imposés comme une solution facile à déployer et à mettre en œuvre dans divers domaines de : l'enseignement (collaboration), la gestion des catastrophes naturelles, militaire, l'agriculture, et l'environnement. Ils sont utilisés surtout comme réseaux d'accès (passerelle) à la toile internet que ce soit pour les entreprises, les universités et même les particuliers.

Plusieurs réseaux sans fil ont été mises en place, chacun ayant ses propres caractéristiques (rayon de portée, qualité de service, sécurité, etc…) et dédié à un type particulier d'application.

Il est à noter que le fait que la technologie sans fil n'est pas toujours mobile, par exemple : un téléphone sans fil de type DECT rattaché à une station de référence n'est pas mobile, il en est de même avec une liaison Wi-Fi, une fois qu'elle est établie, le terminal est rattaché à un point d'accès qu'il ne peut changer sans établir une nouvelle liaison, et donc rompre la précédente [1].

Dans la suite de ce chapitre nous présenterons ces différents standards tout en mettant l'accent sur les réseaux locaux sans fil (WLAN) relatifs à la norme IEEE 802.11 et nous focaliserons notre étude sur la catégorie des réseaux sans fil sans infrastructure : les réseaux ad hoc.

I.2. Définition d'un réseau sans-fil

Un réseau sans fil est un réseau dans lequel au moins deux composants sont capables de communiquer entre eux sans liaison filaire grâce à des ondes radioélectriques (radio et infrarouge).

Ils permettent la connexion d'équipements distants en leur offrant la possibilité de se déplacer dans un périmètre géographique plus ou moins étendu (caractéristique de mobilité) [2].

I.3. Catégories de réseaux sans fil

On distingue habituellement plusieurs catégories de réseaux sans fil. Cette classification se fait en générale selon la portée et/ou le débit. A chaque catégorie correspond un standard (une norme), une technologie et un type d'application (Figure 1. 1).

On trouve principalement les réseaux WPAN, WLAN, WMAN et WWAN, mais certains auteurs ajoutent à ces derniers la catégorie HAN (Home Area Network).

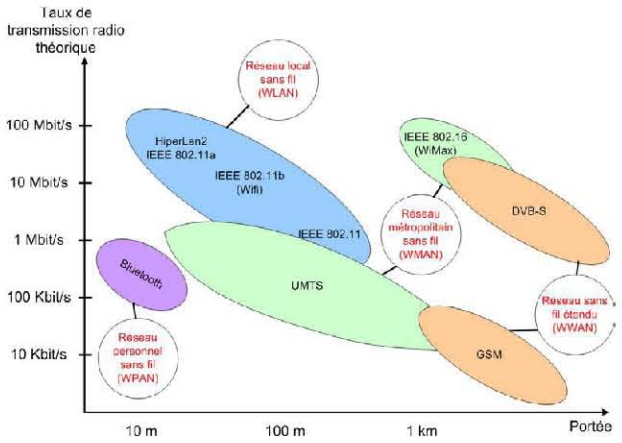


Figure 1. 1: Catégorie de réseaux sans-fil [3]

I.3.1. WPAN

Les réseaux sans fil personnels ou Wireless Personal Area Network (WPAN) (appelé également réseau individuel sans fil ou réseau domestique sans fil), sont des réseaux à très faible portée de l'ordre d'une dizaine de mètres autour de l'usager. Ils permettent de connecter plusieurs équipements informatiques entre eux (PCs, périphériques, PDAs), ou à faire communiquer entre eux des composants domestiques à proximité d'une personne (oreillette, téléphones portables, etc.…). Ils apportent une grande souplesse d'utilisation par rapport à la connexion filaire.

Les technologies utilisées dans ces réseaux offrent des débits faibles et consomment peu d'énergie, ce qui favorise leurs intégration dans de petits équipements autonomes sans fil (i.e. les PDA) et les utiliser aussi pour des réseaux de capteurs.

Pour mettre en œuvre de tels réseaux, les deux principales technologies sont le Bluetooth et le ZigBee (IEEE 802.15(x)). La technologie infrarouge ou IrDA est également utilisée dans ce type de réseaux. Elle est cependant plus sensible aux perturbations lumineuses et nécessite une vision directe entre les éléments souhaitant communiquer ce qui la limite bien souvent à un usage de type télécommande [1][3][4].

I.3.2. WLAN

Les WLAN (*Wireless Local Area Network*) sont des réseaux sans fil qui ont les meilleures performances en débit et en portée. Ils sont un domaine des télécommunications en pleine expansion. Ils offrent de nombreux avantages: mobilité des équipements informatiques, compatibilité des débits avec les applications informatiques actuelles, utilisation des bandes de fréquences libres de droit d'utilisation; infrastructure légère ou inexistante et une mise en œuvre aisée. Mais ils sont moins sûrs et peu fiables.

Ils sont généralement utilisés soit dans des endroits privés (bureaux, salles de conférences), soit dans un environnement public (aéroports, hôtels, restaurants), avec des restrictions. De plus, ils permettent l'interaction avec des infrastructures filaires.

Ces réseaux sont principalement basés sur la technologie IEEE 802.11(x) (Wi-Fi) soutenue par Wireless Ethernet Compatibility Alliance (WECA) ou sur la technologie HiperLan (1 et 2) soutenue par l'European Telecommunications Standards Institute (ETSI).

La différence des WLAN par rapport aux WPAN est une meilleure portée et un meilleur débit [1][3][4].

I.3.3. WMAN

Les réseaux métropolitains sans fil ou Wireless Metropolitan Area Network (WMAN) également appelés boucle locale radio (BLR) étaient à l'origine prévus pour interconnecter des zones géographiques difficiles d'accès à l'aide d'un réseau sans fil. Ces réseaux sont basés sur la technologie IEEE 802.16 connue sous le nom commerciale WiMAX (Worldwide Interoperability for Microwave Access). Ils ont une portée de l'ordre de quelques dizaines de kilomètres (50km) et un débit théorique pouvant atteindre 50 Mbit/s. Cette technologie est destinée principalement aux opérateurs de télécommunication [4].

I.3.4. WWAN

Les WWAN (Wireless Wide Area Network) sont, comme leur nom l'indique, des réseaux de grandes dimensions (plusieurs kilomètres) ou réseaux cellulaires mobiles. Ces technologies nécessitent une infrastructure importante et l'intervention d'opérateurs de téléphonie portable comme par exemple Mobilis, Nedjma, etc...

Les principales technologies sont les suivantes :

- ✓ GSM (Global System for Mobile Communication)
- ✓ GPRS (General Packet Radio Service)
- ✓ MTS (Universal Mobile Telecommunication System)

Les WWAN présentent des débits faibles et des portées très grandes [1].

I.3.5. Les réseaux domestiques

Les HAN (*Home Area Network*) sont des réseaux qui couvrent une localisation fixe. Leur portée peut aller jusqu'à quelques dizaines de mètres. Leur débit varie en fonction de la distance. Ce type de réseaux est essentiellement est utilisé pour la connexion des appareils domestiques [1].

Le tableau ci-dessous (Tableau 1. 1) récapitule les caractéristiques des quatre principales catégories de réseaux sans fil.

	WPAN	WLAN	WMAN	WWAN
Standard	IEEE 802.15	IEEE 802.11	IEEE 802.16	IEEE 802.20
Technologies	Bluetooth HomeRF HR-WPAN Zig-Bee	Wi-Fi HiperLAN	WiMax	GSM GPRS UMTS
Couverture	Quelques dizaines de mètres	Une certaine de mètres	Quelques dizaines de kilomètres	Une certaine de kilomètres
Débit	700 kb/s	De 1 à 54 Mb/s	10 Mb/s	De 10 à 385 Mb/s
Application	Point à point	Réseaux d'entreprise	Fixe	GSM PDA

Tableau 1. 1 : Classification des réseaux sans-fil [2]

I.4. Contraintes et problèmes spécifiques des réseaux sans fil

Si ces réseaux comportent des avantages indiscutables, un certain nombre d'inconvénients existent jusqu'à présent [3][5].

- **Interférences et atténuation** : l'atténuation du signal est proportionnelle à la distance, par contre les interférences sont dues aux bandes de fréquences proches. Ces dernières augmentent le nombre d'erreurs sur une transmission et réduisent les performances d'un lien radio.
- **Liens asymétriques** : la liaison entre l'émetteur et le récepteur et vice versa n'est pas toujours la même.
- **Nature half-duplex des liaisons** : en général, un nœud ne peut réaliser les taches d'émission et d'écoute du canal au même temps (une à la fois) car le signal émis est plus fort que celui reçu.
- **Portée limitée** : un champ de communication limité engendré par l'atténuation du signal.
- **Fiabilité** : le taux d'erreurs est plus important que celui rencontré dans les réseaux filaires. La nature même de ces pertes est différente. En effet, dans un réseau filaire, les pertes sont souvent dues à des congestions contrairement aux réseaux sans fil où les pertes sont majoritairement dues à des problèmes de transmission du signal.
- **Débit** : Il est beaucoup plus faible que celui que l'on trouve dans les réseaux filaires même si les premiers réseaux haut-débit sans fil commencent à voir le jour (500 Mbit/s annoncés pour la 802.11n).

I.5. Les modes de mise en réseau

Les deux modes les plus connues dans les réseaux sans fil [6] sont le mode avec infrastructure et le mode sans infrastructure ou ad hoc (Figure 1. 2).

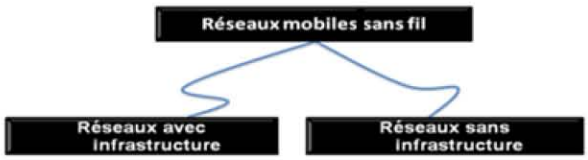


Figure 1. 2: Décomposition des réseaux mobiles sans fil.

- Les réseaux avec infrastructure qui utilisent généralement un modèle de communication cellulaire. Ils connaissent une très forte expansion à l'heure actuelle (les réseaux GSM par exemple) mais requièrent une importante infrastructure logistique et matérielle fixe.
- Les réseaux sans infrastructure ou les réseaux ad hoc qui essaie d'étendre les notions de la mobilité à toutes les composantes de l'environnement. Ils ne requièrent aucune infrastructure, ni de contrôle centralisé.

I.5.1 Les réseaux sans fil avec infrastructure

Les réseaux sans fil avec infrastructure sont essentiellement des réseaux cellulaires. Ce type de réseau est composé de deux ensembles d'entités distinctes: les stations de base (SB) (les point d'accès ou "unités fixes") d'un réseau de communication filaire classique (Wired network), et les "unités mobiles" (UM) (Wireless network). Les SB sont munis d'une interface sans fil pour la communication directe avec les UM localisées dans une zone géographique limitée, appelée cellule (Figure 1. 3). Les UMs peuvent émettre et recevoir des messages à l'intérieur de la cellule où elles sont rattachées, alors que les SB sont interconnectés entre elles à travers un réseau de communication filaire ou sans fil, généralement fiable et d'un débit élevé [6].

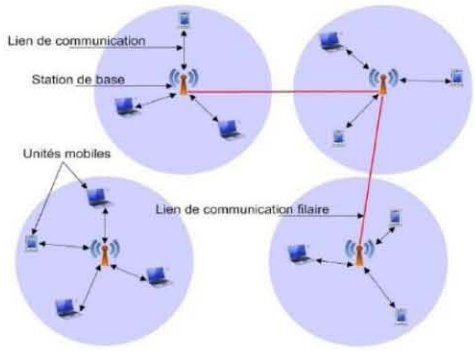


Figure 1. 3 : Réseau sans fil avec infrastructure [2].

Dans ce modèle, une UM ne peut être, à un instant donné, directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres UM à travers la SB à laquelle elle est directement rattachée [6].

Ces réseaux sont souvent retenus pour leur simplicité d'administration. En effet comme les points d'accès sont fixes et n'ont pas de problème d'énergie, la topologie du réseau demeure dans l'ensemble assez stable. De plus, les problèmes de routage (Chapitre 2) sont ici réduits à leur plus simple expression dans la mesure où toutes les communications (sur la partie sans fil) se font en un bond (un saut).

Il existe cependant des problèmes ouverts auxquels s'intéressent de nombreux travaux de recherche.

Parmi lesquels on peut citer celui du changement de réseau aussi appelé handover ou handoff.

Le handover survient lorsqu'un nœud se déplace et quitte la zone de couverture d'une cellule pour entrer dans la zone de couverture de la cellule voisine. Lorsqu'un nœud change d'opérateur en changeant de cellule on dit alors qu'il effectue une forme particulière de handover appelée roaming [3].

I.5.2 Les réseaux sans fil sans infrastructure ou ad hoc

Le réseau **ad hoc** est un réseau sans aucune infrastructure préexistante et aucun support administratif pour sa mise en œuvre (Figure 1. 4). C'est un réseau dont la topologie est variable dans le temps et imprévisible du fait de la mobilité de ses composants (à un moment donné, un mobile peut rejoindre ou quitter le réseau ad hoc). C'est un réseau qui ne comporte pas des stations de base, tous les unités sont mobiles et communiquent entre elles à travers leurs interfaces radios de manière directe. L'absence des SBs dans le réseau oblige les UMs à se comporter comme des routeurs pour la découverte et la maintenance des chemins du réseau [2][5].

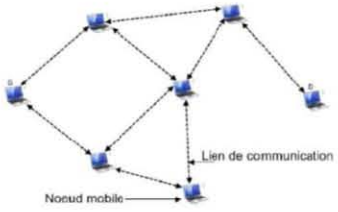


Figure 1. 4: Réseaux ad hoc [2]

Dans ce type de réseau (Figure 1. 4), chaque nœud du réseau joue à la fois le rôle d'élément terminal (émetteur ou récepteur) et le rôle de routeur pour relayer les messages de ses voisins vers un nœud qui n'est pas situé dans le voisinage immédiat.

Le groupe de travail Mobile Ad hoc NETworks (MANET) de l'Internet Engineering Task Force (IETF) a formalisé l'ensemble des caractéristiques des réseaux ad hoc dans la RFC (Request For Comment) 2501 [7].

Les réseaux ad hoc ont la capacité de se relier aux réseaux filaires via une passerelle (Figure 1. 5) dans le but d'avoir accès aux services de ces derniers [8].

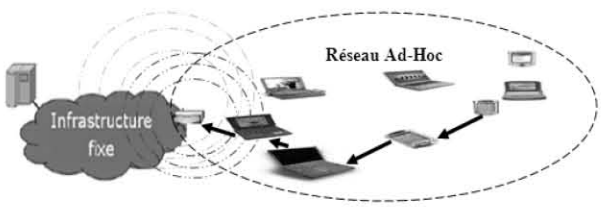


Figure 1. 5: Extension d'une infrastructure en utilisant un réseau ad hoc [5]

Les réseaux mobiles sont caractérisés par de fréquentes déconnexions et des restrictions sur les ressources utilisées (l'énergie par exemple), surtout si tous les usagers du système sont mobiles ce qui est le cas pour les réseaux ad hoc. Ces limitations transforment certains problèmes, ayant des solutions évidentes dans les réseaux classiques, en des problèmes complexes et difficiles à résoudre. Parmi ces problèmes figure le problème de routage [5][6].

Parmi les réseaux ad hoc on trouve les sous types [9]:

1. **MANET** (Mobile ad hoc Networks) : Ils mettent en avant la mobilité des nœuds en cours d'utilisation du réseau.
2. **VANET** (Vehicular ad hoc Networks) : Variante des précédents où les nœuds sont intégrés à des véhicules mobiles.
3. **WSN** (Wireless Sensor Networks) ou **Réseaux de capteurs** : Les nœuds dans ce types de réseaux sont des capteurs dispersés dans une zone donnée et ayant pour but la réalisation de mesures physiques. Ils sont caractérisés par une mobilité généralement faible ou nulle et une énergie faible, mais par contre ils disposent d'une capacité de traitement meilleure que ceux des MANETs.

I.6. Présentation de la norme 802.11 dans les WLAN

La norme 802.11, comme toutes les normes définies par le comité 802, couvre les deux premières couches du modèle OSI (i.e. la couche physique (niveau 1) et la couche liaison de données (niveau 2)). Cette norme dite originale ou initiale date de 1997 pour un débit allant jusqu'à 2 Mbit/s. Depuis, plusieurs extensions ont été introduites permettant des améliorations et des modes de fonctionnement plus performants.

- **Couche liaison** : est subdivisée en deux sous couches (Figure 1. 6) : LLC (Logical Link Control) et MAC (Medium Acces Control). La sous-couche LLC, définie par la norme 802.11, est identique à la couche 802.2 permettant une compatibilité avec n'importe quel autre réseau 802, tandis que la sous-couche MAC est redéfinie par la norme 802.11 (Niveau 2). Elle caractérise l'accès au média de façon commune aux différentes normes 802.11 physiques. Elle est équivalente à la norme 802.3 Ethernet avec des fonctionnalités nécessaires aux transmissions radio. De plus la couche MAC définit deux méthodes d'accès différentes, la DCF (Distributed Coordination Function) ou CP (Contention Period) appelée aussi mode d'accès à compétition, et la PCF (Point Coordination Function) ou CFP (Contention Free Period) appelée mode d'accès contrôlé. La méthode DCF est similaire à Ethernet permettant le transport des données asynchrones où les stations ont une chance égale d'accéder au support. La seconde méthode est le PCF, fondée sur l'interrogation à tour de rôle des stations, ou polling, contrôlée par le point d'accès. Parmi les variantes qu'on trouve dans cette couche on a : la 802.11e (2003) pour le support de la QoS ; la 802.11i (2003) pour l'amélioration de la sécurité et la 802.11f (2003) pour la gestion des changements de point d'accès (Handovers).
- **Couche physique** : définit la technique de transmission (modulation des ondes radioélectriques), l'encodage et la signalisation de la transmission. Elle est divisée en deux sous couches (Figure 1. 6). PLCP (Physical Layer Convergence Protocol) s'occupe de l'écoute du support et de la signalisation en fournissant un CCA (Clear Channel Assessment) à la couche MAC et PMD (Physical Medium Dependent) chargée du traitement de l'encodage des données et la modulation.

  Dans cette couche on trouve : 802.11a (2001) baptisée Wifi5, la 802.11b (1999) connue sous le nom WiFi, la 802.11g (2003) une amélioration de b et sans doute la plus répandue et la 802.11n (2006-2007) dédiée pour un haut débit (World-Wide Spectrum Efficiency : WWiSE ou TGn Sync).

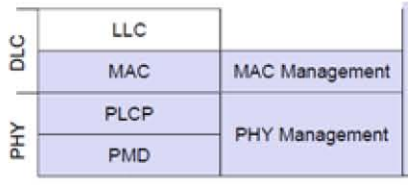


Figure 1. 6: Couches 1 & 2 de la norme 802.11

Le standard IEEE 802.11 regroupe différentes extensions [10]. Dans ce qui suit, une brève description pour chacune des extensions est donnée :

- **802.11b** : (WiFi pour *Wi*reless *Fi*délité) ou 802.11HR (High Rate) fut la première norme des WLAN utilisée par un grand nombre d'utilisateurs (la plus répandue actuellement). A l'heure actuelle, la norme 802.11b est remplacée par la 802.11g, plus rapide. La norme 802.11b permet l'interopérabilité entres les

différents matériels existants, elle propose un débit théorique de 11 Mbit/s (6 Mbit/s réels) avec une portée pouvant aller jusqu'à 300 mètres (en théorie) dans un environnement dégagé. La plage de fréquences utilisée est la bande des 2,4 GHz avec 3 canaux radio disponibles. Son inconvénient est le risque d'interférence avec les appareils fonctionnant aux mêmes fréquences (four à micro onde, matériel sans fils, ...). Cette norme utilise toujours une bande ISM (*Industrial, Science Medical de* 2.4 GHz) et une modulation DSSS (Direct Sequence Spread Spectrum).

- **802.11a** : La norme IEEE 802.11a permet d'obtenir un haut débit (dans un rayon de 10 mètres : 54 Mbit/s théoriques, 27 Mbit/s réels). Elle utilise une bande de fréquences appelée UNII (Unlicensed National Information Infrastructure) divisée en trois parties ne nécessitant aucunes autorisations :
  - UNII-1 (de 5,150 à 5,250 GHz), exploitée en intérieur.
  - UNII-2 (de 5,250 `à 5,350 GHz), exploitée en intérieur et extérieur.
  - UNII-3 (de 5,725 à 5,825 GHz), exploitée en extérieur.

  Son avantage est qu'elle dispose d'une plus grande bande passante par rapport aux normes 802.11b/g et des débits plus importants que la 802.11b (54Mb/s). La 802.11a utilise une technique de modulation OFDM (Orthogonal Frequency Division Multiplexing) différente des autres normes physiques.
- **802.11c** : n'a pas d'intérêt pour le grand public. Il s'agit uniquement d'une modification de la norme 802.11d afin de pouvoir établir un pont avec les trames 802.11 (niveau liaison de données). Elle analyse les procédures de connexion entre les points d'accès.
- **802.11d** (internationalisation) : cette norme 802.11d est un supplément à la norme 802.11 dont le but est de permettre une utilisation internationale des réseaux locaux 802.11. Elle consiste à permettre aux différents équipements d'échanger des informations sur les plages de fréquences et les puissances autorisées dans le pays d'origine du matériel.
- **802.11e** : vise à donner des possibilités en matière de qualité de service (QdS) au niveau de la couche *liaison de données*. Ainsi, cette norme a pour but de définir les besoins des différents paquets en termes de bande passante et de délai de transmission de manière à permettre, notamment, une meilleure transmission de la voix et de la vidéo. Elle prévoit des communications planifiées, dans des intervalles de temps ou aucun trafic n'est transmis.
- **802.11f** : c'est une recommandation à l'intention des vendeurs de points d'accès pour une meilleure interopérabilité des produits. Elle propose le protocole Inter-access point roaming protocol permettant à un utilisateur itinérant de changer de point d'accès de façon transparente lors d'un déplacement, quelles que soient les marques des points d'accès présentes dans l'infrastructure réseau. Cette possibilité est appelée *itinérance* (ou *roaming en anglais*)
- **802.11g** : validée en juin 2003, elle est actuellement la plus répandue dans les produits commerciaux. Elle offre un haut débit (54 Mbit/s théoriques, 25 Mbit/s réels) sur la bande de fréquences des 2,4 GHz. La norme 802.11g a une compatibilité ascendante avec la norme 802.11b (i.e. matériels conformes à la 802.11g restera fonctionnels sous la 802.11b).
- **802.11h** : son but est de mieux gérer la puissance d'émission et la sélection des canaux dans la bande des 5 GHz que ce soit qu'on est à l'intérieur ou à l'extérieur de bâtiments. Elle vise à rapprocher la norme 802.11 du standard Européen (HiperLAN 2, d'où le *h* de 802.11h) et être en conformité avec la réglementation européenne en matière de fréquence et d'économie d'énergie.
- **802.11 i** : elle a pour but d'améliorer la sécurité des transmissions (gestion et distribution des clés, chiffrement et authentification). Cette norme s'appuie sur l'AES (Advanced Encryption Standard) et propose un chiffrement des communications pour les transmissions utilisant les standards 802.11a, 802.11b et 802.11g.
- **802.11 k** : apporte des améliorations dans le domaine de la mesure des ressources radio, dans le but d'arriver à une meilleure gestion du réseau. Elle définit quelles sont les informations qu'il faut rendre disponibles pour la gestion et la maintenance des WLAN.
- **802.11IR** : elle a été élaborée de manière à utiliser des signaux infra-rouges (IR). Cette norme est désormais dépassée techniquement.
- **802.11 m** : elle se charge de la maintenance, des corrections, des ajouts, clarifications et interprétations des documents relatifs à la famille de spécifications 802.11.

- **802.11 n** : est disponible depuis le 11 septembre 2009. Le débit théorique atteint les 600 Mbit/s (débit réel de 100 Mbit/s dans un rayon de 90 mètres) grâce aux technologies MIMO (Multiple-Input Multiple-Output) et OFDM (Orthogonal Frequency Division Multiplexing). Des équipements qualifiés de « pré-N » sont disponibles depuis 2006.

  Le 802.11n a été conçu pour pouvoir utiliser les fréquences 2,4 GHz ou 5 GHz. Les premiers adaptateurs 802.11n actuellement disponibles sont généralement simple-bande à 2,4 GHz, mais des adaptateurs double-bande (2,4 GHz ou 5 GHz, au choix) ou même double-radio (2,4 GHz et 5 GHz simultanément) sont également disponibles. Le 802.11n pourra en théorie atteindre une capacité totale effective de presque un gigabit par seconde [12].
- **802.11 s (réseau Mesh)** : est actuellement en cours d'élaboration. Le débit théorique atteint aujourd'hui 10 à 20 Mbit/s. Elle vise à implémenter la mobilité sur les réseaux de type ad hoc. Tout point qui reçoit le signal est capable de le retransmettre. Elle constitue ainsi une toile au dessus du réseau existant.

Le Tableau 1. 2 ci-dessous donne un récapitulatif pour les différentes normes vues précédemment:

Norme	Description	Statut
802.11a	Nouvelle couche physique : env. 54 Mbit/s sur bande U-NII	Finalisée
802.11b	Nouvelle couche physique : env. 11 Mbit/s sur bande ISM	Finalisée
802.11c	Incorporation des fonctionnalités de 802.1d (pontage)	Finalisée
802.11d	Internationalisation	Finalisée
802.11e	Travaux sur la qualité de service (QoS)	En cours
802.11f	Itinérance (roaming)	En cours
802.11 g	Nouvelle couche physique : env. 54Mbits/s sur bande ISM	Finalisée
802.11h	Harmonisation de 802.11a avec norme européenne HiperLAN	En cours
802.11i	Amélioration des mécanismes de sécurité	En cours
802.11j	Harmonisation de 802.11a avec normes japonaises	En cours
802.11k	Radio Resource Measurement (info. radio fournies par équip.)	En cours
802.11m	Amélioration du standard 802.11 et des normes finalisées	En cours
802.11n	Nouvelle couche physique : 100 Mbit/s	En cours

Tableau 1. 2: normes 802.11x

I.7. Technologies utilisées dans le monde sans fil

Il y a différentes technologies et les plus importantes sont :

- **Bluetooth** a été développé pour les réseaux personnels (PAN). Il offre des communications à courte portée allant du mètre à une centaine de mètres environ et des débits faibles ou moyens entre toute sorte d'équipements. Il travaille dans la bande ISM des 2.4 GHz. Il est géré par le groupe de travail 802.15.

  Les réseaux Bluetooth sont construits de manière centralisée. Un maître élu peut prendre en charge jusqu'à huit esclaves et forme ainsi un piconet. Il contrôle toutes les transmissions en interrogeant régulièrement les esclaves pour savoir s'ils ont des données à envoyer (polling). Plusieurs piconets peuvent être reliés afin de former une structure plus grande appelée scatternet (Figure 1. 7) [13].

  Parmi les dérivées du Bluetooth, on trouve [14]:
  - ✓ 802.15.1 : traduit les spécifications Bluetooth en spécifications de type 802,
  - ✓ 802.15.2 : étudie les configurations des réseaux personnels et notamment la coexistence de ces réseaux avec les réseaux locaux sans fil du type de 802.11,
  - ✓ 802.15.3 : travaille sur les spécifications d'un réseau personnel à haut débit de l'ordre de 20 Mbit/s,
  - ✓ 802.15.4 : étudie les spécifications pour des réseaux personnels à bas débit de l'ordre de 200 Kbit/s.

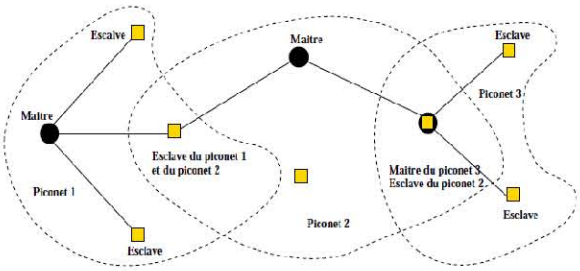


Figure 1. 7 : Schéma de connexion de terminaux Bluetooth [14]

- **Home RF** est un standard développé en 1998 par le "Home Radio Fréquency Working Group". Il utilise comme le Wi Fi la bande de fréquence de 2.4Ghz et offre un débit théorique de 10 Mbits/s mais en pratique, il est de 3 à 4 Mb/s. Sa portée varie entre 50 et 100m.
- **Opnair** est un standard proche du 802.11b, il utilise la même bande de fréquence de 2.4Ghz et propose un débit de 1.6 Mb/s.
- **Hiperlan** (High Performance Local Area Network) : est une technologie développée par ETSI (European Technical Standard Institute). Elle existe en deux versions : Hiperlan 1 et Hiperlan 2 qui peuvent fonctionner ensemble. Elle utilise une bande de fréquence proche de 5 Ghz et offre un débit théorique de 20 Mb/s pour Hiperlan 1 et 54 Mb/s pour Hiperlan 2. Sa portée dépend du milieu (environ 50 mètres et 100 mètres respectivement). Cependant dans les milieux dégagés (type point à point) la connexion sera meilleure que le Wi fi.
- **Hyperlan1** est l'équivalent de la norme 802.11. Il a une architecture totalement décentralisée (Figure 1. 8). Il n'y a pas de notion de point d'accès mais les nœuds HiperLAN 1 peuvent cependant avoir des rôles de passerelles. Il est resté au stade de prototype dans les laboratoires. Les caractéristiques les plus marquantes d'HiperLAN 1 sont [13]:
  - un mécanisme évolué d'accès au médium permettant d'obtenir des garanties de QdS surtout pour les flux multimédias;
  - la possibilité d'étendre le réseau au delà de la portée radio, par sauts successifs (fonctionnement semblable aux réseaux ad hoc).

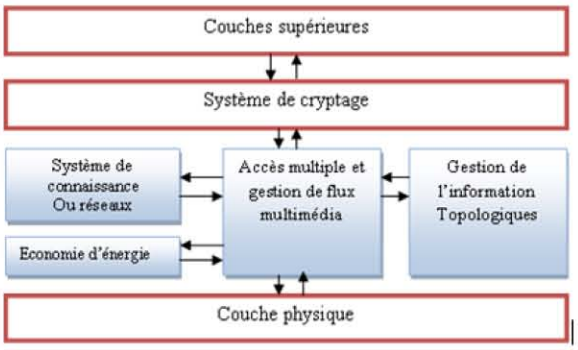


Figure 1. 8: Organisation d'HiperLAN 1[13]

Hyperlan2 compte concurrencée la 802.11 (a et g). Il est basé sur une centralisation poussée avec une architecture très différente du type 1 (Figure 1. 9). Les points d'accès AP (Access Points) ou CC (Central Controler) sont reliés entre eux par une infrastructure réseau filaire ou non. Pour accéder aux ressources du réseau, les mobiles doivent s'attacher à ces points d'accès.

HiperLAN 2 peut aussi fonctionner sans infrastructure fixe, mais dans ce cas, il est différent d'un réseau ad hoc au sens MANET. Dans ce mode, les mobiles pourront communiquer soit directement entre eux (un saut), ou par l'intermédiaire du CC (deux sauts) qui est chargé de l'ordonnancement des communications dans la zone qu'il gère. Ces communications se font grâce à des trames de durée fixe (2 ms) véhiculant soit les informations de contrôle du point d'accès, soit les données. Il faut noter que la couche physique d'HiperLAN 2 est très semblable à celle de 802.11a. Donc HiperLAN 2 est peu adapté aux réseaux ad hoc [13].

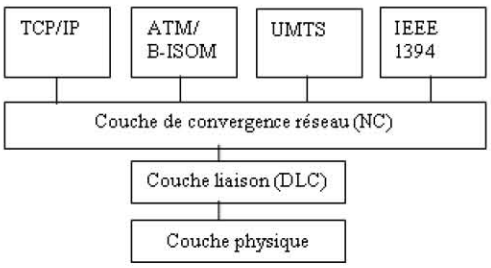


Figure 1. 9: L'organisation générale d'HiperLAN 2 [13]

- **WiFi** (Wireless Fidelity) : est une technologie standard d'accès sans fil à des WLAN, c'est le nom commercial du standard IEEE 802.11b développé en 1999. Il est le plus utilisé dans le monde. Il utilise la bande de fréquence de 2.4Ghz et offre un débit théorique de 11 Mb/s. Sa portée varie entre 10 et 200 m. Ce standard a été développé pour favoriser l'interopérabilité du matériel des différents fabricants [11][13].
- **WiMAX (802.16)**: est l'abréviation pour *Worldwide Interoperability for Microwave Access*. Il s'agit d'un standard de réseau sans fil métropolitain créé par les sociétés Intel et Alvarion en 2002 et ratifié par l'IEEE (*Institute of Electrical and Electronics Engineer*) sous le nom IEEE 802.16. Plus exactement, **WiMAX** est le label commercial délivré par le *WiMAX Forum* aux équipements conformes à la norme IEEE 802.16, afin de garantir un haut niveau d'interopérabilité entre ces différents équipements.

  L'objectif du WiMAX est de fournir une connexion internet à haut débit sur une zone d'un rayon de couverture de plusieurs kilomètres (50 kilomètres) pour un débit théorique de 70 Mbit/s. Le WiMAX possède l'avantage de permettre une connexion sans fil entre une station de base (*Base Transceiver Station*, notée *BTS*) et des milliers d'abonnés sans nécessiter de ligne visuelle directe (*Line Of Sight notée LOS*). Dans la réalité le WiMAX ne permet que de franchir de petits obstacles (i.e. arbres, maisons) mais ne peut en aucun cas traverser les collines ou les immeubles. Le débit réel n'excédera pas 20 Mbit/s.

  Le cœur de la technologie WiMAX est la **station de base**, c'est-à-dire l'antenne centrale chargée de communiquer avec les **antennes d'abonnés**. On parle ainsi de liaison *point-multipoints* pour désigner le mode de communication du WiMAX.

  Les révisions du standard IEEE 802.16 sont en deux catégories [13]:

  WiMAX fixe, également appelé *IEEE 802.16-2004*. Il est prévu pour un usage fixe avec une antenne montée sur un toit (i.e. antenne TV). Le WiMAX fixe opère dans les bandes de fréquence 2.5 GHz et 3.5 GHz, pour lesquelles une licence d'exploitation est nécessaire, ainsi que la bande libre des 5.8 GHz.

  WiMAX mobile (*WiMAX portable*), également baptisé *IEEE 802.16e*, prévoit la possibilité de connecter des clients mobiles au réseau internet. Le WiMAX mobile ouvre ainsi la voie à la téléphonie mobile sur IP ou plus largement à des services mobiles de haut débit.

  Parmi les dérivées du WiMAX on trouve : IEEE std 802.16a (bandes de fréquences comprises entre 2 et 11 GHz), IEEE 802.16b (bandes de fréquences comprises entre 10 et 60 GHz), IEEE std 802.16c (bandes de fréquences libres) et IEEE 802.16d (IEEE std 802.16-2004) est la révision intégrant les standards 802.16, 802.16a et 802.16c en date du 1er octobre 2004.

I.8. Structure de protocole 802.11

La norme 802.11 s'attache à définir les couches basses du modèle OSI (Figure 1. 10) pour une liaison sans fil utilisant des ondes électromagnétiques, 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) ;
  - 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. La norme 802.11 propose donc en réalité trois couches (une couche physique appelée PHY et deux sous-couches relatives à la couche liaison de données du modèle OSI), définissant des modes de transmission alternatifs que l'on peut représenter de la manière suivante :

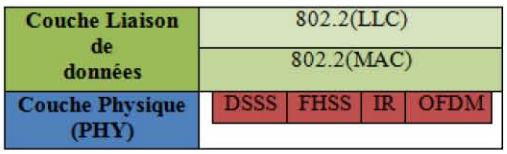


Figure 1. 10 : les couches de la norme 802.11

I.8.1. La couche physique

Elle définit en général, les aspects électriques, mécaniques et fonctionnels de l'accès au canal de communication, ainsi que les protocoles d'échange de données via le réseau [15].

La norme physique 802.11 (ratifiée en 1997) propose deux couches : FHSS et DSSS à base d'une transmission à modulation de fréquence associée à une modulation de phase et une couche basée sur la technique de transmission à infrarouge (IR). Dans la norme 802.11a, on trouve une autre couche dite OFDM.

I.8.1.1. Différentes couches physiques : techniques de transmission

- **FHSS** (Frequency Hoping Spread Spectrum) ou Étalement de Spectre avec Saut de Fréquence : a été créée et brevetée en 1942. Cette technologie utilise la technique de saut de fréquence. Son principe est de diviser la bande de fréquence (bande passante) définie de 2,400 à 2,4835 GHz en 79 sous canaux, de 1 MHz de largeur de bande offrant chacun, un débit allant de 1 à 2 Mbps et le saut se fait toutes les 300 à 400 ms. L'émetteur et le récepteur s'accordent sur une séquence de saut de fréquence porteuse et les données sont envoyées successivement sur les différents sous canaux en évitant (de manière temporaire) d'utiliser les sous canaux fortement perturbés [15].

  Au départ cette technique était utilisée à des fins militaires afin de crypter la transmission mais les séquences de fréquences étant aujourd'hui standardisées, donc divulguées, la norme 802.11 l'utilise pour remédier au phénomène d'interférences. De plus la norme Bluetooth utilise cette technique mais avec des séquences de saut différentes [13].

- **DSSS** (Direct Sequence Spread Spectrum): Le DSSS est une couche physique utilisant une technique radio. C'est une technologie de transmission par spectre étalé, où la porteuse est successivement modulée par l'information et par un code pseudo aléatoire de débit beaucoup plus important. Le signal résultant occupe donc une bande très importante.

  Dans cette technique, la bande des 2.4 GHz est divisée en 14 sous canaux de 22MHz fournissant un signal très bruité surtout entre canaux adjacents seuls trois d'entre eux étant entièrement isolés.

  Cette technique offre des débits de transmission allant de 5.5 à 11 Mbps. Avec comme avantages :

  - Une densité spectrale faible du signal transmis, car ce dernier est large bande,
  - Une sécurité assurée, tant que le code d'étalement reste secret,
  - Une tolérance obtenue vis à vis du multi-trajet en choisissant des codes avec des facteurs d'auto-corrélation faibles.

  Cette technique est moins sensible aux interférences dues aux fréquences parasites à faible largeur spectrale [13][15].

- **IR** (Infra Red) : elle se base sur la transmission en utilisant la lumière infrarouge. Elle est simple, peu réglementée et peu coûteuse. Malgré que la lumière infrarouge possède une large bande passante offrant par conséquent des débits relativement importants, la portée de ce type de communications reste faible (utilisée dans le cas ou les distances entres les différentes stations sont faibles). Elle offre un débit qui peut atteindre les 2Mbit/s en utilisant une technique de modulation appelée PPM (*Pulse Position Modulation)*. Ce type de connexion tend à être remplacé par les voies de communications hertziennes car la nature du medium (lumière) pose problème (sensibles aux interférences lumineuses). En effet, les appareils connectés doivent toujours être en face l'un de l'autre, ce qui n'est pas toujours le cas [15][16].

- **OFDM** (Orthogonal Frequency Division Multiplexing) ou Multiplexage par Répartition Orthogonale de la Fréquence : est une technique née dans les années 50 - 60. Cependant, c'est en 1980 qu'on a commencé à prendre conscience de l'intérêt que représentent l'OFDM et ses applications. Cette technologie représente une technique de modulation numérique des signaux, utilisée entre autres pour les systèmes de transmissions mobiles à haut débit de données. Elle consiste à répartir le signal sur un grand nombre de sous porteuses orthogonales modulées individuellement à bas débit.

L'OFDM est particulièrement bien adapté aux réseaux locaux ou métropolitains mais pas pour les réseaux à grandes échelles. Elle est actuellement utilisée dans plusieurs applications telles que l'ADSL ou le câble pour la diffusion des données, du son ou de l'image. Mais, de plus en plus, cette technologie s'oriente vers les systèmes de communications sans fil. Ainsi, les normes 802.11a et 802.11g peuvent offrir des débits théoriques jusqu'à 54 Mbps ce qui n'est pas pour la norme 802.11b qui n'est pas OFDM, le débit se limite à 11 Mbps [15].

A l'origine, la bande de fréquence utilisée était celle des 900MHz. Les extensions successives ont ajouté d'autres couches physiques afin de permettre des débits de plus en plus élevés.

La version de l'année 1999 du standard 802.11 a remplacé la bande des 900 MHz par la bande ISM (Industry, Scientific and Medical) (2.4 GHz) qui dispose de 14 canaux radio de 20 MHz de largeur.

Suivant les réglementations, ils ne sont pas tous utilisables dans tous les pays (Tableau 1. 3).

Pays (organisme régulateur)	Bandes de fréquence
Etats-Unis (FCC)	2.400 - 2.485 GHz
Europe (ETSI)	2.400 - 2.435 GHz
Japon (MKK)	2.471 - 2.497 GHz
France (ART)	2.400 - 2.454 GHz à 100 mW 2.454 - 24835 GHz à 10 mW en extérieur 2400 - 2483.5 GHz a 100mW en intérieur

Tableau 1. 3: Réglementations de la bande ISM [13]

De plus, il faut préciser que ces canaux ne sont pas tous indépendants les uns des autres. Ils se chevauchent en partie. Il faut également noter que la bande ISM tend à être saturée car de nombreux équipements l'utilisent (dont Bluetooth, 802.11b et 802.11g).

La figure (Figure 1. 11) récapitule les différentes technologies et débits possibles pour chacune d'entre elles.

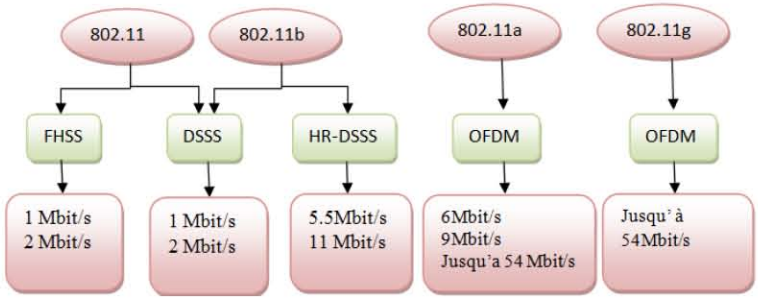


Figure 1. 11: Récapitulatifs des technologies et des débits possibles

I.8.2. La couche mac 802.11

L'une des particularités du standard 802.11 est qu'il définit deux mécanismes d'accès fondamentalement différents au niveau de la couche MAC. Le premier est le DCF (Distributed Cordination Function), qui correspond à une méthode d'accès assez similaire à celle des réseaux traditionnels supportant le best-effort. Le DCF a été conçu pour prendre en charge le transport de données asynchrones, dans lequel tous les utilisateurs qui veulent transmettre des données ont une chance égale d'accéder au support.

Le second mécanisme d'accès est le Point Coordination Function (PCF). Fondé sur l'interrogation à tour de rôle des terminaux, ou polling, sous le contrôle du point d'accès, la méthode PCF est conçue essentiellement pour la transmission de données sensibles, qui demandent une gestion de la QdS et opère en mode synchrone. PCF est utilisé pour les applications temps réel, telles que la voix ou la vidéo. Un réseau en mode ad hoc utilise uniquement le DCF, tandis qu'un réseau en mode infrastructure utilise à la fois le DCF et le PCF [17].

- **DCF** (Distributed Coordination Function) : La méthode d'accès DCF est une méthode d'accès en mode Best-Effort, c'est à dire sans priorité et sans garantie. Elle est très efficace pour transporter les trafics ne nécessitant pas de garantie sur la latence de transmission ou le débit offert (téléchargement, navigation sur Internet, etc.). Elle s'appuie sur l'algorithme CSMA/CA, qui est l'une des nombreuses variantes de la méthode d'accès au médium CSMA. Cette variante présente la particularité d'être adaptée au médium sans fil dans la mesure où, dans le cas d'un médium sans fil, il est difficile de détecter une collision

instantanément. Contrairement à un médium filaire, il n'est pas possible, sur un médium sans fil, d'entendre le médium en même temps que l'on transmet des données, ceci à cause des phénomènes d'aveuglement et d'affaiblissement, alors qu'en filaire, Ethernet utilise la variante CSMA/CD, qui permet de détecter une collision quasi-instantanément : la comparaison de ce qui est émis et de ce qui est entendu permet de savoir si il y a collision alors que dans le cadre de CSMA/CA, la collision est détectée ultérieurement par la non réception d'un message d'acquittement.

- **PCF** (Point Coordination Function): La PCF appelée mode d'accès contrôlé. Elle est fondée sur l'interrogation à tour de rôle des stations, ou polling, contrôlée par le point d'accès. Une station ne peut émettre que si elle est autorisée et elle ne peut recevoir que si elle est sélectionnée. PCF est aussi un mode dans lequel les stations de base ont la charge de la gestion de l'accès au canal dans leur zone de couverture pour les mobiles qui leur sont rattachés. Cette méthode est conçue pour les applications temps réel (vidéo, voix) nécessitant une gestion du délai lors des transmissions de données [18].

I.8.2.1.Description du mode DCF

Les fonctions principales de la sous couche MAC de 802.11 en mode DCF sont : le protocole CSMA/CA, l'algorithme de Backoff, les espaces inter trames (IFSs), le mécanisme RTS (*Request To Send*) et CTS (*Clear To Send*) et le mécanisme de fragmentation.

I.8.2.1.1. Les espaces inter trames

Comme pour la norme 802.3 Ethernet, l'espace entre trames (**InterFrame Space : IFS)** joue un rôle crucial pour coordonner l'accès au médium. Le protocole 802.11 CSMA/CA dans le mode DCF utilise différents espaces entre trames (IFS) (i.e., SIFS, PIFS, DIFS et EIFS) comme indiqués dans la figure 1.14.

Les espaces entre trames (IFSs) sont en fait des périodes d'inactivité sur le support de transmission qui permettent de gérer l'accès compétitif par les stations au médium, d'instaurer un système de priorités lors d'une transmission. L'IFS est calculé à base d'une unité de temps dite « slot time ». Le standard définit cinq types d'espace entre trames pour contrôler l'accès au médium, c'est-à-dire permettant d'instaurer des priorités entre les accès. Ces temporisations sont utilisées par la méthode DCF et par les autres fonctions d'accès (i.e. PCF, EDCA) [18][19]:

- **SIFS** (Short InterFrame Space) sera utilisé pour séparer deux trames faisant partie d'un même échange atomique ; par exemple entre la transmission d'une trame et la transmission de l'acquittement correspondant, entre la transmission d'une trame RTS et la trame CTS correspondante, entre deux trames faisant partie d'une même rafale. SIFS est le plus petit des IFSs, il permet ainsi d'empêcher qu'un échange en cours ne soit interrompu par une autre transmission. Sa durée est définie par la valeur de aSIFSTime. SIFS représente la plus haute priorité et assure qu'une station est capable de finir la séquence de transmission de trame avant qu'une autre station puisse accéder au médium.
- **PIFS** (PCF InterFrame Space) est la temporisation utilisée par l'élément central du réseau (le point d'accès : AP) pour intervenir au profit d'une scrutation (polling). Cette temporisation est utilisée par la fonction PCF et par son équivalent HCCA dans la norme 802.11e. Elle est d'une durée supérieure à SIFS (afin d'éviter qu'une scrutation n'interrompe un échange en cours) mais inférieure à DIFS (afin de rendre une scrutation plus prioritaire qu'un accès avec contention). Après l'expiration de cet intervalle, n'importe quelle trame du mode PCF peut être transmise.
- **DIFS** (DCF InterFrame Space) est la temporisation minimale de médium libre qu'une fonction d'accès avec contention (DCF) doit attendre avant de pouvoir entamer une procédure d'accès. Il est plus grand que PIFS. Après l'expiration de cet intervalle, n'importe quelle trame du mode DCF peut être transmise, de façon asynchrone selon le mécanisme du Backoff de CSMA/CA. Donc DIFS a la plus faible priorité.
- **AIFS** (Arbitration InterFrame Space) est utilisée par EDCA, il représente l'équivalent de DIFS dans DCF, à savoir une temporisation de séparation des trames représentant une partie du mécanisme de différenciation introduit par EDCA dans la norme 802.11e.
- **EIFS** (Extended InterFrame Space) est une temporisation, d'une durée supérieure à DIFS, utilisée suite à une réception erronée par une station afin d'éventuellement permettre à une autre station du réseau d'acquitter la réception du paquet qui, pour la première station, contenait des erreurs.

I.8.2.1.1 Problème du CSMA dans le cas des réseaux sans fil

Si les mécanismes de détection de collisions s'avèrent adaptés pour un réseau local câblé, ils ne le sont pas, en général, pour les réseaux radio. Plusieurs raisons sont à citées :

- Dans un environnement sans fil, on ne peut pas être sûr que toutes les stations s'entendent entre elles (ce qui est l'hypothèse de base du principe de détection de collision), et le fait que la station voulant transmettre teste si le support est libre, ne veut pas forcement prédire qu'il est libre du coté récepteur.
- **Problème des stations cachées** : Ce problème se produit quand deux stations ne peuvent pas s'entendre l'une et l'autre du fait que la distance qui les sépare est trop grande ou qu'un obstacle les empêchent de communiquer entre elles mais elles ont des zones de couverture qui se recoupent. Si les stations A et C ne font que la détection de porteuse en écoutant le canal, n'étant pas en mesure de s'entendre l'une et l'autre, elles vont s'autoriser à émettre des paquets au même temps à une station B située dans l'intersection des zones de couverture, ce qui va provoquer une collision. On dit que les stations A et C sont cachées l'une par rapport à l'autre (Figure 1. 12).

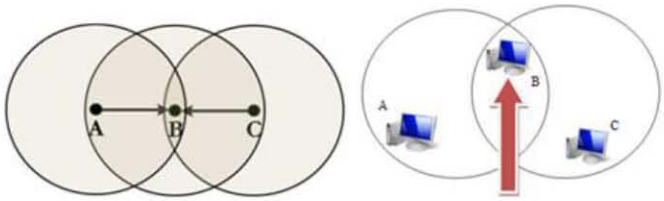


Figure 1. 12 : Le problème des nœuds cachés

- **Le problème des stations exposées**: ce problème arrive dans le cas où une station B transmet des données à une station A. Si une station C écoute le canal radio, elle détecte la communication en cours, et conclut qu'elle ne pourra transmettre des paquets à une station D, or si C transmettait, cela créerait des collisions seulement dans la région entre B et C et non dans les régions où D et A se situent (Figure 1. 13).

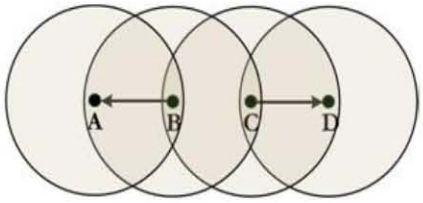


Figure 1. 13: Le problème des nœuds exposés

I.8.2.1.2 Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

CSMA/CA est un mécanisme d'écoute de la porteuse à accès multiple avec évitement de collision. L'idée retenue pour 802.11 est, lorsque le canal devient libre, d'attendre une période de durée aléatoire supplémentaire " Backoff " avant d'émettre. Ce mécanisme s'applique lorsque le canal devient libre aussi bien après une de nos propres émissions qu'après toute autre émission.

La station qui a choisie le plus petit Backoff va commencer à émettre. Les autres, dès qu'ils détectent le regain d'activité sur le canal stoppent la décrémentation de leur Backoff et entrent en période de defering (i.e. temps entre l'instant où la STA stoppe la décrémentation du Backoff et l'instant ou ce dernier atteint la valeur zéro bien sûr après re-lancement de la décrémentation). Les stations ne pourront reprendre leur décrémentation que si le canal redevient à nouveau libre pendant DIFS (Figure 1. 14) [13].

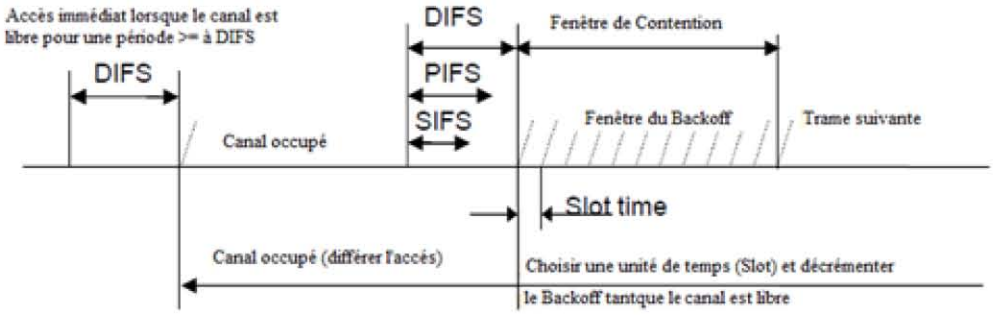


Figure 1. 14 : diagramme d'écoute et d'ajournement dans DCF (CSMA/CA) (extrait de IEEE 802.11)

Bien que cette méthode permette d'éviter les collisions, il est cependant possible que deux stations parviennent à avoir la même valeur du Backoff et émettent en même temps ce qui provoque une collision. Pour cela, CSMA/CA impose un accusé de réception pour chaque paquet de données reçu correctement.

a) Principe de l'accusé de réception ACK :

Une station voulant transmettre écoute le support, et s'il est occupé, la transmission est différée. Si le support est libre pour un temps spécifique DIFS, alors la station est autorisée à transmettre après une durée tirée aléatoirement en se basant sur l'algorithme de Backoff exponentiel (cf. § suivant). En retournant un accusé de réception (ACK), la destination confirme la bonne réception du paquet de données. La réception de l'ACK indiquera à la source qu'aucune collision n'a eu lieu mais dans le cas contraire, il retransmet le paquet jusqu'à ce qu'il y a succès ou abandon au bout d'un certain nombre de retransmissions fixé par la norme.

Remarque : c'est la couche MAC qui est informée des collisions par l'attente d'un accusé de réception (ACK) pour chaque paquet transmis. Dans le cas de non réception d'un ACK, la couche MAC retransmet le paquet sans avoir à passer par les couches supérieures, ce qui engendrait des délais significatifs.

Afin de surveiller l'activité du réseau, la sous couche MAC travaille en collaboration avec la couche physique qui utilise l'algorithme CCA (Clear Channel Detection) pour évaluer la disponibilité du canal. Pour savoir si le canal est libre, la couche physique mesure la puissance reçue par l'antenne appelée RSSI (Received Signal Strength Indicator). La couche physique détermine donc si le canal est libre en comparant la valeur du RSSI à un certain seuil et transmet par la suite à la couche MAC un indicateur de canal libre. Dans le cas contraire, la transmission est différée [20].

Dans la DCF (Figure 1. 15), une station doit écouter le canal avant d'initialiser l'envoi d'un paquet. Si le canal est libre pendant un temps DIFS, la station peut transmettre son paquet. Les stations en écoute constatent une émission, déclencheront pour une durée fixée leur indicateur de Virtual Carrier Sense (VCS) (appelé NAV : Network Allocation Vector) et utiliseront cette information pour retarder toute transmission prévue.

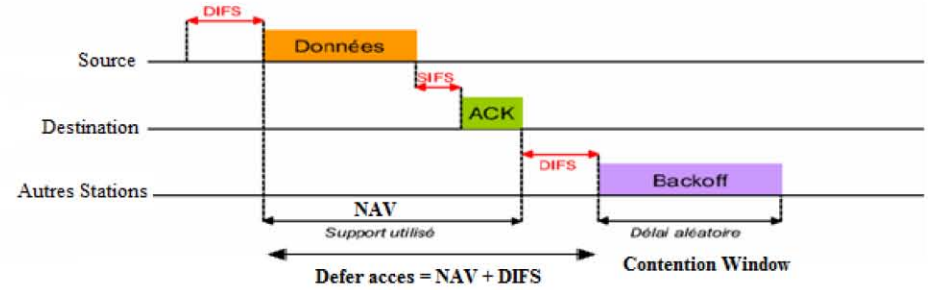


Figure 1. 15: Le mécanisme d'accès DCF (sans RTS/CTS) [17]

Les valeurs des différents PIFS et DIFS sont calculées de la manière suivante : PIFS=SIFS+*Slot Time et* DIFS= SIFS+ 2 * *Slot Time*. Où *Slot Time* est l'intervalle minimal entre deux opérations de détection physique de porteuse. Cette valeur est dépendante des caractéristiques de la couche physique considérée. C'est une constante spécifiée par le standard pour une couche physique donnée.

Les IFS permettent de définir des degrés de priorité. Lorsque plusieurs stations souhaitent émettre simultanément, la station souhaitant émettre les trames les plus prioritaires comme les acquittements pourra les envoyer en premier. Puis seront transmises d'autres trames jugées prioritaires comme celles liées à l'administration réseau ou au trafic qui a des contraintes de délai. Enfin, les informations les moins importantes concernant le trafic asynchrone seront émises après un temps d'attente plus long.

b) Algorithme de Backoff exponentiel BEB (Binary Exponenttiel Backoff)

Le Backoff est une méthode bien connue pour résoudre les différents entre plusieurs stations voulant accéder au support. Cette méthode demande que chaque station choisisse un délai d'attente aléatoire compris entre 0 et la taille d'une fenêtre de contention de valeur CW qui est égale à un certain nombre de slots, et d'attendre ce nombre de slots avant d'émettre si bien sûr aucune autre station n'a pas accédé au support avant elle.

La durée d'un slot (SlotTime) est définie de telle sorte que la station sera toujours capable de déterminer si une autre station a accédé au support au début du slot précédent. Cela divise la probabilité de collision par deux (Figure 1. 16).

Le Backoff exponentiel signifie qu'à chaque fois qu'une station choisit un slot et provoque une collision, la durée d'attente aléatoire est augmentée exponentiellement (Figure 1. 17).

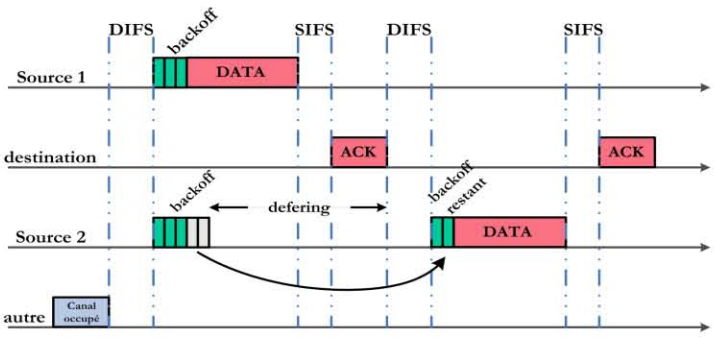


Figure 1. 16 : l'algorithme de backoff et le defering [13]

Le standard 802.11définit l'algorithme de Backoff exponentiel devant être exécuté dans les cas suivant :

- Quand la station écoute le support avant la première transmission et que le support est occupé,
- Après chaque retransmission,
- Après une transmission réussie.

Le seul cas où ce mécanisme n'est pas utilisé est quand la station décide de transmettre un nouveau paquet et que le support a été libre pour un temps supérieur au DIFS.

La durée d'attente aléatoire (DAA) du Backoff est calculée de la manière suivante : DAA= random(0,CW)*SlotTime, où random(0,CW) est une variable aléatoire uniforme comprise entre 0 et CW-1 où CW est la taille de la fenêtre de contention, CW = [CWmin CWmax].

Lors de la première tentative de transmission, CW = CWmin; et à la fois suivante (en cas de collision), après chaque échec de transmission, la fenêtre de contention (CW) est doublée selon la formule $CW = (CW_{min} \times 2^{1})-1$ (dont les valeurs autorisées par la norme ne sont que des puissances de 2 moins 1) jusqu'à ce qu'elle atteigne une valeur maximale prédéfinie CWmax. La borne supérieure de la fenêtre est ré-initialisée à CWmin sitôt qu'un paquet a été transmis correctement (ou lorsque les timers de ré-émission expirent). Un exemple d'évolution de la fenêtre de contention est donné sur la figure 17.

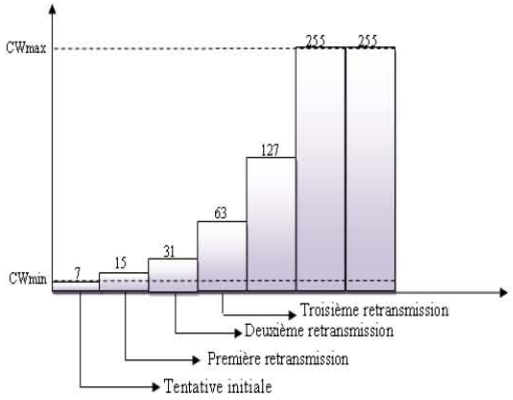


Figure 1. 17 : Un exemple de backoff exponentiel

c) Le mécanisme RTS/CTS

Le CSMA/CA basique offre un excellent mécanisme pour éviter les collisions mais ne règle pas tous les problèmes des transmissions radios. Afin de pallier celui de la station cachée (Figure 1. 12) où deux émetteurs qui ne peuvent pas du tout s'entendre (en général à cause d'un obstacle) veulent atteindre un même récepteur, la norme IEEE 802.11 ajoute le mécanisme d'échange de paquets RTS/CTS entre l'émetteur et le récepteur avant le début de la transmission des paquets de données. Le principe de l'échange RTS/CTS appelé mécanisme de VCS (Virtual Carrier Sense) est illustre par l'exemple de la Figure 1.18 [2].

Lorsque cette fonction est utilisée, une station émettrice transmet un RTS et attend en réponse un CTS. Toutes les stations du réseau recevant soit le RTS, soit le CTS, déclencheront pour une durée fixée leur indicateur NAV pour retarder toutes transmissions prévues (Figure 1. 18). Dans chaque nœud, le NAV indique pour combien de temps le canal est utilisé par quelqu'un d'autre, indépendamment de ce qui est physiquement perçu sur le canal (on parle aussi de détection de porteuse " logique ") qui leur permet alors de se bloquer et s'empêcher d'émettre pendant toute cette période (durée précisée dans le paquet RTS).

La station émettrice peut alors transmettre et recevoir son accusé de réception sans aucun risque de collision. Le mécanisme RTS/CTS est particulièrement performant si la taille des paquets de données est grande. Or, si ces derniers sont petits alors le mécanisme RTS/CTS résultera d'une surcharge du réseau. Pour savoir si un échange RTS/CTS doit avoir lieu, le standard définit un seuil nommé RTS_Threshold, si la taille de la trame est plus grande que ce seuil, alors un échange RTS/CTS doit être effectué avant l'envoi de la trame.

Dans la Figure 1. 18 on présente les mises à jour du NAV au niveau d'un mobile alors qu'une trame est échangée entre deux autres mobiles. Lorsque le nœud non concerné par l'échange reçoit le RTS, il sait grâce aux informations contenues dans ce dernier pour combien de temps il ne devra pas accéder lui-même au canal [17][21].

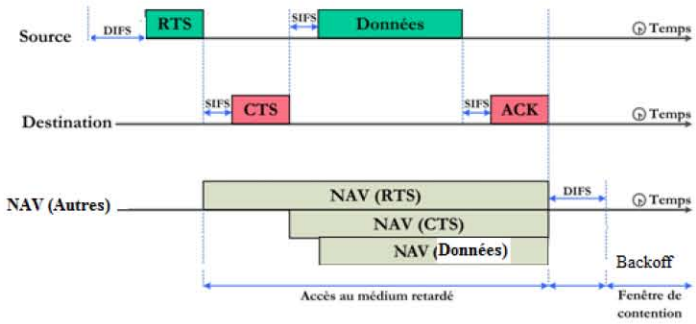


Figure 1. 18 : Le mécanisme d'accès DCF (avec RTS/CTS) [17]

d) Le mécanisme EIFS

Dans la configuration présentée sur la Figure 1. 19 (à gauche), le nœud nommé « autre » arrive à détecter la porteuse du nœud émetteur sans pour autant comprendre ses messages (le signal est trop faible pour être décodé, mais suffisamment fort pour être reconnu comme tel). Afin d'éviter une collision au niveau de l'émetteur au moment de la réception d'un CTS ou d'un acquittement, la norme IEEE 802.11 impose l'utilisation d'une intertrame EIFS (Extended Inter Frame Spacing).

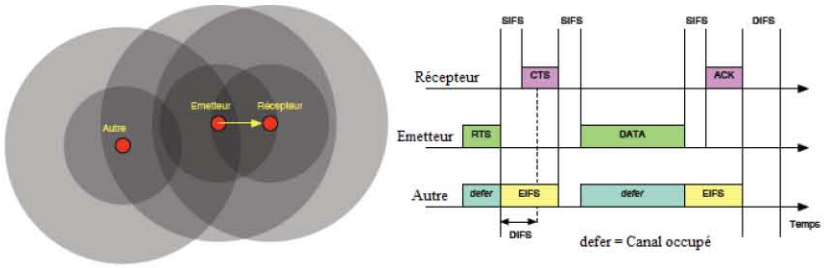


Figure 1. 19 : Le mécanisme EIFS [13]

La Figure 1. 19 (à droite) détaille ce qui se passe : Tout d'abord le nœud émetteur diffuse un paquet RTS qui sera correctement reçu par le récepteur et non décodable par le nœud « autre » car il est très loin. Le nœud récepteur répondra alors par la diffusion d'un CTS. On note que le nœud « autre » n'a pas pu décoder le paquet RTS du nœud émetteur et par conséquent il n'a pas pu mettre à jour son NAV. Le mécanisme de détection de porteuse l'empêche d'émettre durant l'envoie du RTS et pendant une durée DIFS consécutive, le DIFS étant plus court que SIFS+CTS. Si jamais le nœud « autre » avait terminé de décrémenter son Backoff trop vite, il aurait pu émettre pendant l'envoie du CTS ne s'écoule, causant une collision au niveau de l'émetteur. Pour protéger les paquets CTS et ACK, la norme IEEE 802.11 impose qu'un nœud qui reçoit un signal non décodable attende une durée EIFS avant d'accéder au canal.

La durée de l'EIFS doit être suffisamment longue pour que la réception du CTS ou de l'ACK se déroule dans de bonnes conditions et se fasse sans collisions [2][13][16][22].

I.8.2.1.3 Fragmentation et Réassemblage

Dans les environnements radio, plus la taille d'une trame est importante, plus elle a de chance d'être corrompue. La fragmentation d'une trame en plusieurs trames de taille inférieure accroît la fiabilité de la transmission (les petites trames ont une probabilité plus importante que les grandes trames d'être émises sans erreurs). En utilisant cette propriété, la fragmentation a été ajoutée au mécanisme de la couche MAC 802.11 (Figure 1. 20). Les grandes trames sont ainsi divisées en de petits fragments lorsque leur taille dépasse un certain seuil. Chaque fragment est émis et acquitté séparément. Une fois que la station a accès au medium

elle peut émettre plusieurs séquences Data/ACK séparées par SIFS, ce qui lui assure de ne pas être interrompue par une autre station avant de compléter sa transmission [23].

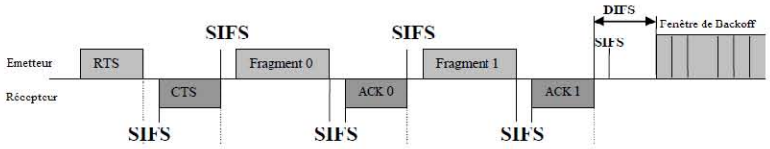


Figure 1. 20 : Transmission des fragments d'un MPDU (MAC Protocol Data Unit) séparés par SIFS [24]

I.8.2.2. Point Coordination Function (PCF)

Dans la DCF, on avait recours à une durée aléatoire avant l'émission de chaque paquet. Le temps passé à attendre représente autant de débit effectif perdu. Pour remédier à ceci et obtenir un meilleur taux d'utilisation du canal, la 802.11 propose en option un mécanisme d'accès centralisé. Ce mode optionnel et complémentaire au mode d'accès décentralisé, ne fonctionne qu'avec infrastructure en centralisant l'accès au canal au niveau d'une station de base qui contrôle le support et autorise ou non les stations à émettre en définissant un PC (*Point Coordination)*. Le PC régit le partage en utilisant une structure temporelle appelée SuperTrame qui se décompose en deux périodes de temps avec ou sans contention (Figure 1. 21).

La CP ***(Contention Period)*** : Correspond à une période de temps avec contention durant laquelle la méthode d'accès est DCF.

La CFP ***(Contention Free Period*** : Est une période de temps sans contention durant laquelle la station de base impose l'ordre des transmissions à chacun des terminaux qui lui sont rattachés.

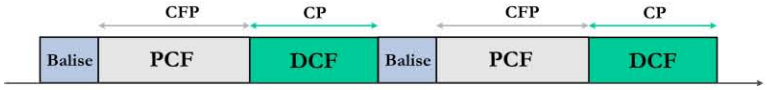


Figure 1. 21: L'alternance des modes PCF et DCF [15]

Cette décomposition permettra la cohabitation des deux modes d'accès afin de permettre aux stations n'implémentant pas la PCF d'accéder au canal. Le point d'accès se charge de la génération des périodes d'accès sans contention avec une période prédéfinie "CFP_MAX_duration" (durée maximale possible de la période CFP). Le temps est divisé en supertrame contenant chacune une période sans contention ou Contention Free Period (CFP) suivi d'une période avec contention ou Contention Period (CP) (Figure 1. 21).

Le début d'une supertrame (la CFP) est marqué par l'envoi par le PC d'un paquet Beacon (Balise). Le paquet Beacon doit être envoyé à une allure régulière, une temporisation PIFS est employée pour l'envoi du Beacon.

Le point d'accès utilise une intertrame PIFS *(PCF Inter Frame Spacing)* entre deux envois de la supertrame. L'intertrame PIFS est d'une durée intermédiaire entre SIFS et DIFS. L'incertitude sur l'instant effectif de début d'une Super Trame est au plus d'une durée correspondant à l'enchainement RTS-CTS-Trame de données-ACK. La balise qui marque le début de la Super Trame peut être retardée impliquant une réduction de la durée de l'intervalle de temps affecté au mode PCF comme illustré sur la Figure 1. 22 [2][13][16][22].

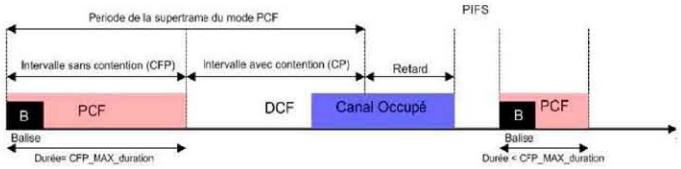


Figure 1. 22 : Le mode PCF [2]

I.9. Conclusion

Dans ce chapitre, après avoir exposé les différentes normes de réseaux locaux sans fil (HiperLAN1 et 2, Bluetooth, Wifi, etc…) ; une présentation est faite sur la norme 802.11 et ses différentes variantes (a, b,….g, ….n). Ensuite un petit rappel sur la couche physique et les techniques utilisées comme FHSS, DSSS, IR et OFDM a été donné. Le dernier point a été consacré à la couche MAC, la DCF, la PCF ainsi que les mécanismes de RTS/CTS et EIFS.

À cause de la mobilité, la topologie du réseau évolue au cours du temps ce qui va poser des problèmes de routage dans les réseaux ad hoc. Ce dernier a été largement étudié ces dernières années et plusieurs classes ont vu le jour. Tout ceci avec plus de détails fera l'objet du prochain chapitre.

Chapitre 2 : Routage dans les réseaux ad hoc

II.1 Introduction

La problématique du routage dans les réseaux sans fil est identique à celle du routage dans les réseaux traditionnels (filaires). Il s'agit d'aiguiller les paquets vers leur destination en fonction d'informations (topologie du réseau) que chaque station à en sa possession.

Pourtant, le routage dans les réseaux sans fil présente toutefois des particularités comme la fragilité de la transmission radio s'appuyant généralement sur la diffusion (joindre en une seule transmission tous les voisins), la limitation de la bande passante disponible, et le caractère unidirectionnel des liaisons radio.

Le groupe de travail MANET (*Mobile Ad hoc Networks*) fût crée en 1997 par l'IETF (*Internet Engineering Task Force*) pour définir une spécification de protocole de routage pour les réseaux sans fil ad hoc. Il est le seul organisme de normalisation à ce jour qui se penche sur ce problème.

Dans la suite de ce chapitre nous commençant par une définition du routage, les difficultés rencontrées et les caractéristiques de ces protocoles dans le contexte ad hoc. Ensuite, nous abordons les différentes techniques de classification du routage: hiérarchique, plat, à la source, saut par saut, vecteur de distance et à état de liens. Suivie par la description des grandes familles de protocoles (proactives, réactives, hybrides, hiérarchiques et géographiques).

Pour chaque classe, deux protocoles sont étudiés comme représentants de la famille. Et nous terminons par une discussion sur le problème de routage et les protocoles étudiés.

II.2 L'acheminement de l'information dans les réseaux ad hoc

En tant que système de communication, un réseau ad hoc doit acheminer l'information d'un nœud source vers un nœud destination. Deux types d'acheminement sont possibles : l'envoi direct et le routage.

II.2.1 L'envoi direct

N'aura lieu que si l'environnement physique le permet. L'émetteur doit pouvoir envoyer ses données directement d'un nœud à un autre quelque soit la destination. Les nœuds mobiles sont suffisamment proches les uns des autres ce qui permet aux nœuds d'avoir des liens étroits et directs entre eux et aucun autre intermédiaire ne peut s'interposer dans cette relation directe privilégiée.

II.2.2 Le routage

Dans un contexte ad hoc de nature multi-sauts, les communications se font entre paires (source-destination ou émetteur-récepteur). Dans le cas où le nœud destinataire se trouve dans la portée du nœud émetteur (envoi direct) nous n'aurons pas besoin de routage proprement dit, malheureusement ce n'est pas toujours le cas, en effet, lorsque deux nœuds trop éloignés (i.e. ne sont pas dans le rayon de portée d'une communication directe) souhaitent communiquer, ils ont besoin de nœuds relais (i.e. routeurs) qui participent à l'acheminement des paquets de données vers leurs destinations [25]. Pour cela l'information doit transiter (i.e. router) de nœud en nœud (i.e. de proche en proche) jusqu'à arriver à sa destination [13].

Le but principal du routage est l'établissement de routes qui soient correctes et efficaces entre une paire quelconque de nœuds, ce qui assure l'échange des données d'une manière continue. Vu les limitations des réseaux ad hoc, la construction des routes doit être faite avec un minimum de contrôle et de bande passante.

Dans les réseaux ad hoc, les protocoles de routage sont souvent soumis à des contraintes liées aux problèmes de changements fréquents de la topologie et d'énergie, ainsi qu'à la nature du canal de communication.

Un protocole de routage, pour qu'il soit opérationnel, doit prendre en compte les trois cas suivants :

- **Dissémination de l'information de routage** : Elle permet de connaître suffisamment d'éléments sur la topologie pour choisir un chemin vers le nœud de destination. Selon la quantité d'informations échangées, la vue faites par les nœuds sur la topologie du réseau est plus ou moins précise. Le protocole de routage est censé d'optimiser l'envoi de ces informations pour mieux gérer la bande passante.
- **Sélection de chemin**: Une fois les informations de routage obtenues, le protocole de routage peut sélectionner une route suivant un certain nombre de critères (i.e. le nombre de sauts pour atteindre la

destination) parmi les routes disponibles. Les routes choisies doivent être dépourvues de boucles (temporaires et permanentes [26]) afin d'éviter à un paquet de données de tourner indéfiniment.

- **Maintenance des routes**: dans un réseau mobile, la topologie du réseau ne cesse d'évoluer avec le temps. Donc, les routes sont amenées à changer avec le déplacement des nœuds. Le protocole de routage doit donc tenir compte de ces changements et permet restituer les routes précédemment coupées [25].

II.2.2.1 Définition du routage

Le routage est une méthode à travers laquelle on fait transiter une information donnée depuis un certain nœud émetteur vers un nœud destinataire bien précis. Cette fonction est assurée par les nœuds eux même qui coopèrent à la transmission de cette information. Le routage ne se résume pas seulement à trouver un chemin entre les deux nœuds du réseau, mais encore à trouver un acheminement optimal pour router les paquets de données.

Donc, l'objectif du routage est de déterminer une route (i.e. un ensemble de liens à parcourir), respectant certaines contraintes, pour établir une connexion entre deux entités communicantes (source & destination).

Le routage est réalisé au niveau de la couche réseau [2]. Les nœuds, dans un réseau ad hoc, peuvent être à la fois émetteurs, récepteurs et routeurs disposant dans ce dernier cas d'une table de routage contenant les informations de correspondance entre les nœuds faisant partie d'un chemin donné.

La table de routage est construite d'une manière dynamique sans aucune configuration initiale par échange des messages entre routeurs selon la stratégie des algorithmes de routage à vecteur de distance (Distance Vector) ou à états de lien (Link State) (voir paragraphe 2.4.3) [27].

Grâce à ce routage, la portée radio d'un nœud peut être virtuellement étendue en utilisant ses voisins comme relais de l'information. Par exemple sur la Figure 2. 1, le nœud A peut envoyer des messages au nœud G bien que celui-ci soit hors de portée radio (cercle en pointillé) du nœud A. Pour cela il utilise la route BCFG.

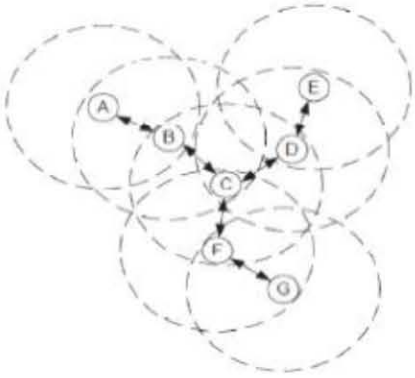


Figure 2. 1: Le principe du routage dans les réseaux ad hoc [4]

II.2.2.2 Problématique

Les réseaux ad hoc sont souvent peu stables avec des caractéristiques spécifiques ajoutant une complexité et des contraintes supplémentaires qui doivent être prises en compte lors de la conception des algorithmes et des protocoles de routage. Notant que les problématiques posées dans un contexte des réseaux ad hoc sont différentes et plus complexes que celles rencontrées dans le monde filaire qui sont entre autre :

1- En raison de la mobilité physique des nœuds, la topologie d'un réseau ad hoc est susceptible de fréquents changements (il faut assurer l'interopérabilité des différents réseaux et garantir sa stabilité pour éviter que les transmissions ne soient interrompues, sans que le trafic nécessaire à la connaissance de cette topologie ne mobilise toute la bande passante [30]).

2- **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 (CPU, mémoire), de logiciel, de taille (petit, grand) et de mobilité (lent, rapide). Dans ce cas, une adaptation dynamique des protocoles s'avère nécessaire pour supporter de telles situations.

3- **La contrainte d'énergie** : Les équipements mobiles disposent de batteries limitées, 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 [8].

4- **La densité** dans les réseaux ad hoc semble être une question préoccupante quand le nombre de nœuds mobiles augmente: le fait que la taille d'un réseau ad hoc peut être énorme, souligne que la gestion de routage de l'environnement doit être complètement différente des approches de routage classiques. La plupart des protocoles développés pour ad hoc sont optimisés pour des petit nombre, le problème qui se pose est l'adaptation de la méthode d'acheminement utilisée avec le grand nombre d'unités existant dans un environnement caractérisé par de modestes capacités de calcul et de sauvegarde.

5- Des problèmes liés au support de communication hertzien et sa qualité variable dans l'espace et dans le temps. En effet l'utilisation des liaisons radios introduit des différences clairs et de nouvelles problématiques par rapport aux communications filaires telles que la limitation physique ou réglementaire de la capacité disponible pour l'accès radio, la qualité fluctuante des liens radios (influence des obstacles, du mouvement, des interférences, …). De même, le médium radio est peu fiable en termes de perte d'information et de sécurité.

6- les liens radio peuvent être asymétriques, l'information passe dans un sens mais pas dans l'autre (à cause des irrégularités des ondes électromagnétiques) [4].

7- Un problème lié a la localisation des positions des différents nœuds du réseau [33].

8- Dans les réseaux ad hoc basés sur un protocole d'accès avec détection de porteuse (comme le 802.11), les caractéristiques de l'accès au support génèrent des problèmes supplémentaires : le problème de la station cachée et celui de la station exposée. Il faut bien trouver un protocole de routage qui soit en mesure de prendre ça en considération [11].

Assurer la fonction de routage dans de telles conditions devient une tâche complexe. Il faut bien choisir un protocole de routage qui sera capable de remédier à ces problèmes.

II.2.2.3 Caractéristiques des algorithmes de routage

Un algorithme doit:

- Optimiser les ressources du réseau et éviter les boucles de routage
- Empêcher la concentration du trafic autour de certains nœuds ou liens.
- Offrir un support pour pouvoir effectuer des communications multipoints fiables.
- Assurer un routage optimal et permet de prendre en compte différentes métriques de coûts (bande passante, nombre de liens, ressources du réseau, délais de bout en bout,…etc.).
- Assurer une maintenance efficace de routes avec le moindre coût possible.
- pouvoir s'adapter aux changements de topologie rapidement, en proposant des routes de longueur acceptable, même en cas de forte mobilité des nœuds [6][28].

Les algorithmes de routage doivent être :

- Correctes, simples et optimaux (offrent les meilleurs chemins),
- Robustes (les réseaux sont en place pour longtemps),
- Stables (une communication continue indépendamment des coupures),
- Adaptatifs ou non (opèrent dans toutes les situations) [29].

II.3 Modes de communication dans un réseau ad hoc

Les principaux modes de communication dans les réseaux mobiles sont :

a) La communication point à point ou unicast, pour laquelle il y a une seule source et une seule destination,

b) La communication multipoint ou multicast, qui permet d'envoyer un message d'une source à plusieurs destinataires,

c) La diffusion ou Broadcast, qui envoie un message d'une source à tous les nœuds du réseau.

Ces trois modes de communication sont schématisés par la Figure 2. 2.

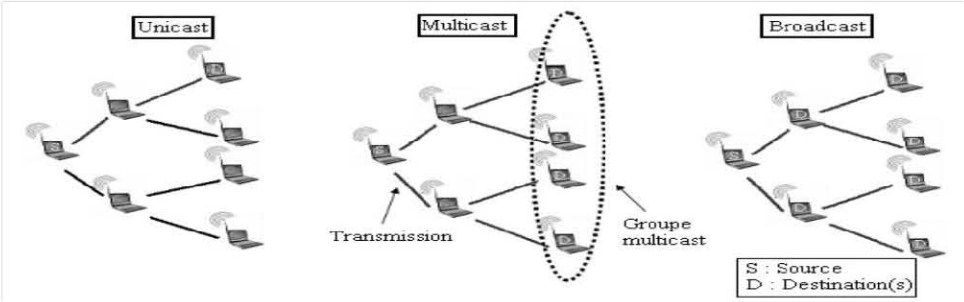


Figure 2. 2: Modes de communication dans les réseaux mobiles [11]

II.4 Différentes classifications des protocoles de routage

Les protocoles de routage pour les réseaux ad hoc peuvent être classés suivant plusieurs critères.

II.4.1 Routage hiérarchique ou plat

Le premier critère utilisé pour classifier les protocoles de routage dans les réseaux ad hoc concerne le type de vision qu'ils ont du réseau et les rôles qu'ils accordent aux différents mobiles.

- **Les protocoles de routage "à plat"** considèrent que tous les nœuds sont égaux (Figure 2. 3(a)). La décision d'un nœud de router des paquets dépendra de sa position et pourra être remise en cause dans le temps. L'AODV (Ad hoc On Demand Distance Vector) est un exemple utilisant cette technique.

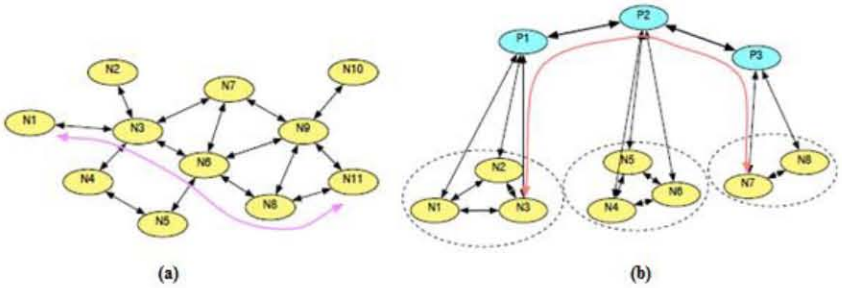


Figure 2. 3 : Routage « à plat » (a) & routage hiérarchique (b) [13]

- **Les protocoles de routage hiérarchique** : fonctionnent en confiant aux mobiles des rôles qui varient de l'un à l'autre. Certains nœuds sont élus et assument des fonctions particulières qui conduisent à une vision en plusieurs niveaux de la topologie du réseau. Par exemple, un mobile pourra servir de passerelle pour un certain nombre de nœuds qui se seront attachés à lui. Le routage en sera simplifié, puisqu'il se fera de passerelle à passerelle, jusqu'à celle directement attachée au destinataire. Un exemple est donné sur la Figure 2. 3(b), où le nœud N3 passe par les passerelles P1, P2 et P3 pour atteindre N7. Dans ce type de protocole, les passerelles supportent la majeure partie de la charge du routage (les mobiles qui s'y rattachent savent que si le destinataire n'est pas dans leur voisinage direct, il suffit d'envoyer à la passerelle qui se chargera du reste). Ce type de routage présente certains avantages. Un exemple de protocole utilisant cette stratégie est l'OLSR (Optimized Link State Routing) [13].

II.4.2 Le routage à la source et le routage saut par saut

- **Le routage à la source** : ou « Source Routing » consiste à déterminer complètement les routes à partir des nœuds sources. Il fonctionne en incluant dans chaque paquet routé l'intégralité du chemin que devra suivre le paquet pour atteindre sa destination. L'entête de paquet va donc contenir la liste des différents nœuds relayeurs vers la destination. Le protocole le plus connu se basant sur cette classe est : DSR (Dynamic Source Routing) [31].
- **Le routage saut par saut** : ou "Hop by Hop" consiste à donner uniquement à un paquet l'adresse du prochain nœud vers la destination. L'AODV fait partie des protocoles qui utilisent cette technique.

II.4.3 Etat de liens ou vecteur de distance

Une Autre classification, héritée du monde filaire, est possible pour les protocoles de routage : les protocoles 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.

Les algorithmes de routage basés sur ces deux méthodes, utilisent la même technique qui est la technique des plus courts chemins, et permettent à un hôte donné, de trouver le prochain hôte pour atteindre la destination en utilisant le trajet le plus court existant dans le réseau [27].

- **Les protocoles à état de lien** : cherchent à maintenir dans chaque nœud une carte plus ou moins complète du réseau où figurent les nœuds et les liens les reliant. A partir de cette carte il est possible de construire les tables de routage. Cette famille de protocoles se base sur les informations rassemblées sur l'état des liens dans le réseau. Ces informations sont disséminées dans le réseau périodiquement ce qui permet ainsi aux nœuds de construire une carte complète du réseau. Un nœud qui reçoit les informations concernant l'état des liens, met à jour sa vision de la topologie du réseau et applique un algorithme de calcul des chemins optimaux afin de choisir le nœud suivant pour une destination donnée.

  Chaque nœud commence par établir la liste de ses voisins et le coût de la communication avec chacun d'eux (Figure 2. 4(a)). Il diffuse ensuite cette liste partout dans le réseau grâce à un mécanisme appelé *inondation*. Un des avantages de ce type de protocole est leur capacité à pouvoir facilement trouver des routes alternatives lorsqu'un lien est rompu. Il est même possible d'utiliser simultanément plusieurs routes vers une même destination, augmentant ainsi la répartition de la charge et la tolérance aux pannes dans le réseau. En contre partie, si le réseau est étendu, la quantité d'informations à stocker et à diffuser peut devenir considérable. Les principaux protocoles de routage qui appartiennent à cette classe sont les suivants : TORA (Temporally Ordered Routing Algorithm routing protocol) [32], OLSR et TBRPF (Topology Broadcast based on Reverse Path Forwarding) [9][13].
- **Les protocoles à vecteur de distance**. Plutôt que de maintenir une carte complète du réseau (ce qui peut s'avérer extrêmement lourd), ces protocoles ne conservent que la liste des nœuds du réseau et l'identité du voisin par lequel passer pour atteindre la destination par le chemin le plus court. A chaque destination possible sont donc associés le saut suivant (next-hop) et une distance généralement en nombre de sauts. La démarche adoptée (Figure 2. 4(b)) consiste alors à diffuser à ses voisins (et non plus au réseau entier) des informations concernant les chemins choisis pour atteindre chacun des autres nœuds connus (et non plus seulement ses propres voisins). Si un voisin envoie un paquet de contrôle dans lequel il indique être plus près d'une destination que le saut suivant que l'on utilisait jusqu'alors, alors il le remplace dans la table de routage. Un des inconvénients de cette technique et qu'il est du coup plus difficile de conserver plusieurs routes alternatives au cas où celle qui est privilégiée serait rompue (on ne dispose que du saut suivant, et on ne sait pas si la suite de la nouvelle route est indépendante de celle qui a été rompue). Les protocoles de routage basés sur le vecteur de distance les plus connus pour les réseaux ad hoc sont : DSR, DSDV (Dynamic destination-Sequenced Distance-Vector) et AODV [13].

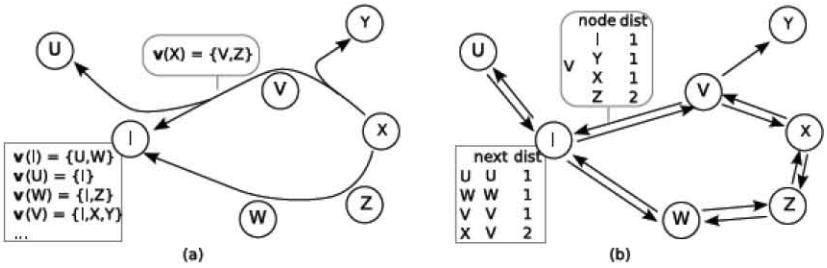


Figure 2. 4: Algorithmes d'état de lien (a) et de vecteur de distance (b) [9]

II.4.4 Protocoles uniformes et non-uniformes

Certains protocoles de routage n'utilisent pas tous les nœuds d'un réseau pour faire transiter les messages, au contraire ils en sélectionnent certains, en fonction du voisinage ou pour former des cellules. Ces protocoles sont dits non-uniformes. Ceux qui utilisent tous les nœuds du réseau capables de router sont appelés protocoles uniformes.

II.5 La classification du groupe MANET

C'est la classification qui nous intéresse et qu'on maintient pour la suite de ce chapitre. Le principal but de toute stratégie de routage est de mettre en œuvre une bonne gestion d'acheminement qui soit robuste et efficace. D'une manière générale, les protocoles de routage peuvent être répertoriés suivant la façon dont ils disséminent l'information de contrôle (le mode de mise à jour de l'information de routage) que l'on peut classiquement regrouper dans les trois premières grandes familles de protocoles [34][35]: les protocoles

proactifs ou "Table Driven" avec un comportement identique à ceux des réseaux filaires (échange périodique des informations sur la topologie par chaque nœud du réseau), les protocoles réactifs ou "On Demand" (échange des informations de routage uniquement lors de la création d'une route) et les protocoles hybrides (un mélange entre les deux types précédents, proactif pour de faibles distances et réactif pour distances supérieures). Comme ils peuvent également être classés selon le critère de hiérarchie entre nœuds et on parle dans ce cas des protocoles hiérarchiques. Le dernier critère de classification est l'utilisation d'informations de localisation (position géographique) nés avec l'apparition de systèmes de positionnement comme GPS (Global Positioning System) appelés protocoles géographiques (Figure 2. 5) [4][25][36][37][38].

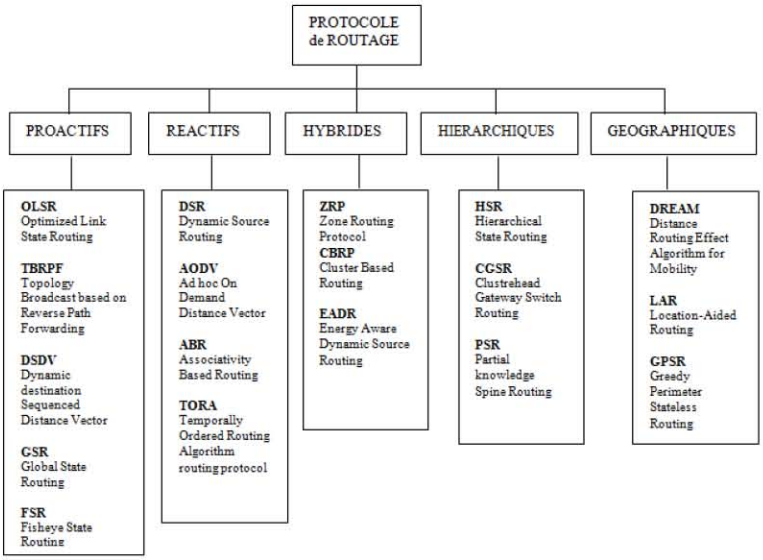


Figure 2. 5: Classification et exemples des protocoles de routage dans les réseaux ad hoc

II.5.1 Les protocoles de routage proactifs

Les protocoles de routage proactifs reprennent le principe du routage des réseaux filaires. Ici les procédures de création et de maintenance des routes, durant la transmission des paquets de données, sont contrôlées périodiquement. En cas d'absence de trafic de données dans le réseau, cette maintenance reste toujours active. Dans ce type de protocole, Chaque nœud maintient une table de routage contenant les informations nécessaires (par exemple le prochain nœud sur le chemin…) pour atteindre un autre nœud du réseau. En consultant sa table de routage, un nœud peut à tout instant transmettre un paquet de données vers un autre nœud du réseau. Des mises à jour périodiques de l'état de la topologie gardent effectives les routes présentes dans la table de routage.

Le principal avantage de ces protocoles est leur ***réactivité***. En effet, à tout moment chaque élément du réseau connaît un moyen d'atteindre les autres membres du réseau.

En revanche, il faut être capable d'actualiser les tables de routage (maintenance) en permanence pour tenir compte de la mobilité des nœuds, cela entraîne la diffusion de nombreux messages de contrôle (trafic additionnel nécessaire au maintient de l'état des routes) pour le rafraîchissement des informations concernant la topologie. Si le rafraîchissement est trop élevé, comparé à l'évolution de la topologie, le nombre d'informations de routage émises sur le réseau est trop important, consommant inutilement de la bande passante. Dans le cas contraire s'il est trop faible, les tables de routage ne sont pas suffisamment mises à jour, rendant les informations qu'elles contiennent obsolètes. Pour un fonctionnement optimal de ce type de protocoles, un compromis entre l'échange des informations de routage et la prise en compte de l'évolution de la topologie doit être trouvé [25]. Vu que les nœuds participent pleinement à ces mises à jour permanentes, une énergie plus importante est consommée inutilement

Deux principales méthodes sont utilisées dans cette classe de protocoles proactifs : la méthode ***Link state*** et la méthode ***Distance Vector***. Les protocoles proactifs fonctionnent vis-à-vis de la couche réseau comme des protocoles de routage classiques.

Les différents protocoles proactifs se différencient principalement par le mode de mise à jour des tables de routage [4]. Parmi les protocoles de routages proactifs les plus connus on citera le DSDV, OLSR, TBRPF, GSR (Global State Routing), FSR (Fisheye State Routing), WRP (Wireless Routing Protocol) …

Deux des protocoles les plus cités en littérature en l'occurrence OLSR et DSDV seront présentés [39] :

II.5.1.1 Le protocole OLSR

Le protocole OLSR a été standardisé en 2003 [40]. Son fonctionnement est basé sur l'algorithme à état de liens [25] où de nombreux changements ont été apportés pour le rendre exploitable dans un réseau ad hoc [4]. Il est considéré comme une optimisation du protocole LSR (Link State Routing) pour les réseaux mobiles ad hoc. LSR fonctionne sur le principe d'une inondation globale du réseau par les messages de contrôle où chaque nœud signale périodiquement son état à ses voisins; qui à leur tour propagent cette information à tous le réseau. Cette technique d'inondation consomme la bande passante disponible qui est généralement limitée dans les environnements sans fil ce qui cause une dégradation de performance surtout des réseaux de grande dimension.

Pour une meilleure gestion de la bande passante et pour réduire le nombre de paquets nécessaires à l'échange de la topologie (un paquet est émis uniquement à son voisinage immédiat) [25]. L'innovation du protocole OLSR réside dans sa façon d'économiser les ressources radio lors des diffusions grâce à l'utilisation de la notion du concept des multipoints relais (MPR: Multi Point Relay) [41] a qui on délègue la retransmission de l'information dans le réseau [8].

Des messages de contrôle périodiques doivent être utilisés pour le maintien des tables de routage et de voisinage. Dans le protocole OLSR, les deux principaux messages utilisés sont les paquets "Hello" et les paquets TC (Topology Control). Périodiquement, chaque nœud diffuse localement un paquet Hello contenant des informations sur son voisinage et l'état des liens. Ce type de paquet comprend la totalité de la base de liens (ensemble des nœuds lui ayant transmis un paquet Hello) connue par l'émetteur du paquet. Ceci permet à chaque nœud de prendre connaissance de son voisinage à un et deux sauts. Une fois les voisins découverts, les nœuds peuvent échanger les informations sur leur voisinage pour former la topologie du réseau. Cette fonction est attribuée à des nœuds particuliers appelés relais multipoints (MPRs), les seuls autorisés à transmettre les informations de routage [25] (voir Figure 2. 6).

L'ensemble MPR est alors construit dans chaque nœud de façon à contenir un sous-ensemble de voisins à un saut qui couvre tous les voisins à deux sauts. Afin de construire les tables nécessaires au routage des paquets, chaque nœud génère périodiquement un paquet TC contenant la liste de ses voisins l'ayant choisi comme MPR. Le message TC est diffusé dans l'ensemble du réseau. Seuls les voisins MPR rediffusent un paquet TC reçu pour éviter l'inondation. A la réception d'un message TC, la table de topologie peut être construite. Chaque nœud peut calculer la table de routage qui permet d'acheminer les paquets vers n'importe quelle destination dans le réseau [5][8].

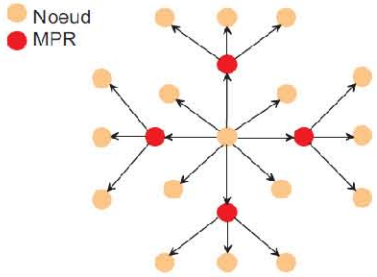


Figure 2. 6 : Le principe des nœuds MPR [**4**]

La table de routage est calculée à partir de la table de topologie et celle des voisins à chaque fois qu'au moins une de ces tables est modifiée. Un algorithme de plus court chemin est utilisé pour transmettre les données à destination, en se servant des routes disponibles dans la table de routage. Chaque entrée de route contient, entre autres, les informations suivantes : les adresses du nœud destination et le nœud suivant sur la route, la distance en nombre de sauts vers la destination.
La table de routage est modifiée lorsqu'un changement est détecté au niveau des liens, des voisins à un et deux sauts, et de la topologie.

La Figure 2. 7 présente le fonctionnement du protocole OLSR où le nœud 8, après avoir effectué une détection de voisins et une diffusion de topologie, a mis à jour les quatre tables qui y figurent [39].

Le protocole OLSR est performant dans les réseaux denses car les MPR permettent de limiter l'inondation du réseau. De plus il est très réactif, chaque nœud sait à tout moment comment atteindre les autres. Par contre, en termes de consommation énergétique, il sollicite beaucoup les nœuds car ils doivent émettre en permanence des messages et c'est en émission que les supports radio consomment le plus [4].

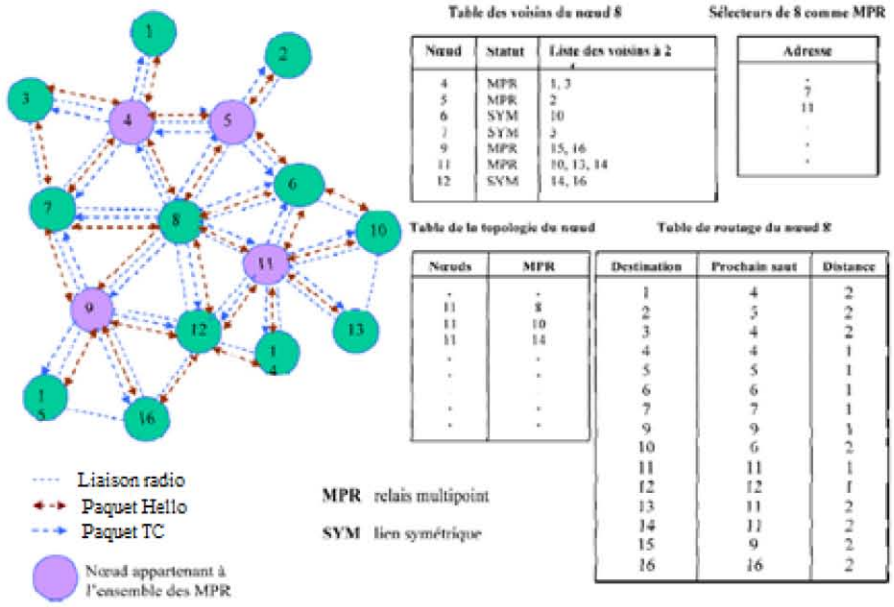


Figure 2. 7 : Fonctionnement du protocole OLSR [39]

II.5.1.2 Le protocole DSDV

DSDV est un protocole basé sur l'algorithme distribué de Bellman-Ford, en rajoutant quelques améliorations. DSDV élimine le problème de comptage à l'infini (counting to infinity) et celui de boucles de routage (routing loop) [15]. Il est basé sur l'algorithme du vecteur de distance utilisant comme métrique le nombre de sauts. Le protocole à vecteur de distance permet de limiter l'échange des messages de contrôle de la topologie uniquement aux voisins d'un nœud. Ce point est extrêmement important pour préserver la bande passante disponible sur le réseau [25].

Dans ce protocole chaque nœud maintient une table de routage qui contient toutes les destinations possibles, le nombre de sauts pour atteindre la destination, le numéro de séquences (SN) qui correspond à un nœud destination, permettant de distinguer les nouvelles routes des anciennes, ce qui évite la formation de boucles. Si le nœud reçoit un paquet de mise à jour possédant un numéro de séquence inférieur au dernier traité par le nœud, ce paquet est ignoré [5].

La Tableau 2. 1 représente la table de routage du nœud N4 du réseau ad hoc de la Figure 2. 8 [4].

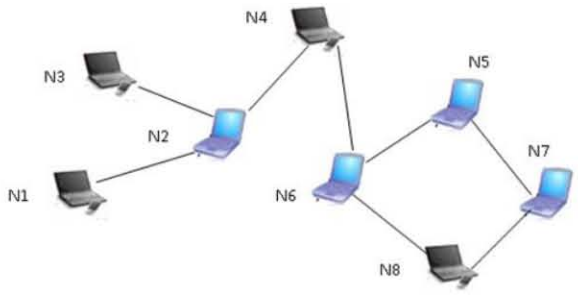


Figure 2. 8 : Un exemple de réseau utilisant le protocole DSDV d'après [42]

Destination	Saut suivant	Métrique	Numéro de Séquence
N4	N4	0	SN-4
N1	N2	2	SN-1
N2	N2	1	SN-2
N3	N2	2	SN-3
N5	N6	2	SN-5
N6	N6	1	SN-6
N7	N6	3	SN-7
N8	N6	2	SN-8

Tableau 2. 1: Un exemple de table de routage d'un nœud de réseau DSDV d'après [42]

Les tables de routage sont mises à jour périodiquement mais également lors d'événements particuliers (la découverte d'une route invalide, déplacement de nœuds, apparition d'un nouveau voisin...) par diffusion à travers le réseau afin de maintenir la consistance des informations. La diffusion a pour avantage de rendre l'information disponible à l'ensemble des voisins mais elle génère un trafic important qu'il faut limiter [4].

En général, un paquet de mise à jour contient le nouveau numéro de séquence incrémenté de l'émetteur (un numéro pour dater l'information) et l'ensemble de sa table de routage (l'adresse de la destination, le nombre de nœuds pour atteindre cette destination et le numéro de séquence des données reçus à la destination).

A partir de 02 numéros de séquence, il est possible de déterminer quelle information est la plus récente [25].

Cette mise à jour peut se faire de deux façons selon la stabilité du réseau :

Mise à jour complète: si le réseau subit des changements rapides qui n'est rien autre que la mise à jour périodique, c'est-à-dire que le nœud transmet la totalité de sa table de routage vers ses voisins ce qui nécessite l'envoi d'un volume important des paquets de contrôle.

Mise à jour incrémentale: si le réseau est relativement stable; cette mise à jour n'est faite qu'en cas d'événements (rupture de lien, mobilité de nœuds ...etc.), et dans ce cas il n'y a que l'entrée concernant le nœud en question dans la table de routage qui change (paquets plus petits, ne contenant que les informations qui ont subit un changement depuis la dernière mise à jour ce qui réduit le nombre de paquets de contrôle). Cette mise à jour est aussi dite mise à jour partielle [6].

A la réception de ces informations, les voisins mettent à jour leur table de routage en suivant un schéma bien précis. Toute entrée de la table de routage est mise à jour, seulement, si l'information reçue est plus récente (celle avec la plus grande valeur du numéro de séquence), ou si elle a le même âge (a le même numéro de séquence) mais possède la meilleure métrique (moins de sauts). A terme, le protocole DSDV fournit pour chaque destination, la route qui possède le plus faible nombre de nœuds.

Sur des réseaux de grande dimension, la mise à jour des tables en cas de mobilité des nœuds peut être lente car elle est initiée par la destination. Un autre problème de ce protocole est qu'il utilise une mise à jour basée sur les événements, ce qui engendre un contrôle excessif dans les communications.

Pour être un protocole de routage complet, le protocole DSDV doit maintenir l'état des chemins. Pour cela, les nœuds détectent les ruptures de lien. Chaque nœud émet, périodiquement, ses informations de routage à l'ensemble de ses voisins. Si pendant un certain temps, un nœud ne reçoit plus les informations de routage d'un nœud voisin c'est que ce dernier ne fait plus partie de son voisinage. Un lien coupé affecte l'ensemble des routes utilisant ce lien. Un nœud, décelant une coupure, diffuse un paquet contenant l'ensemble des destinations ne pouvant plus être atteint à travers ce lien. Tout nœud, recevant un tel paquet, le propage immédiatement pour faire connaître au plus vite le changement de topologie.

Un des problèmes de cet algorithme est qu'il réagit trop lentement aux mauvaises nouvelles. La destination doit prendre connaissance d'une coupure pour transmettre une mise à jour de la topologie [25].

II.5.2 Protocoles de routages réactifs

Les protocoles réactifs n'essayent de déterminer les routes que lorsqu'elles sont demandées par les nœuds. Lorsqu'un nœud demande une route vers une destination, il initie une découverte de route. Ce processus est complété une fois qu'une route est trouvée ou si toutes les routes possibles ont été examinées. Une fois que la route est établie, elle est maintenue jusqu'à ce qu'une procédure de maintenance de route détecte l'inaccessibilité de la destination à partir de ce chemin ou que la route ne soit plus désirée. AODV, DSR, TORA sont quelques protocoles de routage de cette famille [35][39][43][44].

II.5.2.1 Le protocole DSR

DSR [45] est un des premiers protocoles de routage qui a été proposé pour les réseaux sans fil ad hoc par le groupe MANET. Il a été standardisé en 2007 [46]. Le protocole DSR est un protocole de routage réactif unicast, simple et efficace, dédié aux réseaux ad hoc mobile multi sauts [47]. Son fonctionnement est très proche du protocole AODV à la grande différence qu'il fournit dans les paquets de données l'ensemble des nœuds permettant d'atteindre une destination (routage par la source). Cet ajout dans les paquets de données accroît le surcoût et consomme un peu plus de bande passante. A contrario, ces informations lui permettent de gérer l'asymétrie des liens présents dans le réseau. En effet, un paquet de données peut prendre une route différente de son acquittement. Le fonctionnement basique de DSR s'avère assez simple à mettre en œuvre. Il met en place uniquement deux phases [25]: la découverte de route et la maintenance de route.

Découverte de route (Route Discovery) : Lorsqu'un nœud cherche à émettre un paquet qui provient de la couche supérieure vers une destination pour laquelle il n'a pas de route en cache, le nœud initie une découverte de route vers la destination (cible) et met le paquet dans un tampon. Ce dernier sera automatiquement vidé après un délai sans réponse.

Un message ROUTE REQUEST (RREQ) est envoyé en diffusion à l'aide d'un mécanisme d'inondation. Chaque paquet RREQ contient les informations nécessaires au bon fonctionnement de la découverte de route, à savoir : l'adresse du nœud initiateur, l'adresse de la cible, un numéro d'identification unique de la requête, ainsi qu'une liste de tous les nœuds parcourus par le message. Cette liste est évidemment différente pour chaque instance du message [47].

Lorsqu'un nœud intermédiaire reçoit un paquet RREQ, il vérifie tout d'abord s'il a déjà reçu le message. Pour cela, il utilise les champs adresse source, adresse destination et identifiant qui permettent d'identifier de manière unique un paquet RREQ. Si un tel message a déjà été reçu, il est supprimé. Dans le cas où la requête lui est destinée, il l'acquitte en envoyant un paquet (RREP) au nœud initiateur confirmant le chemin « source-destination », sinon il ajoute sa propre adresse au paquet RREQ et le propage par diffusion à d'autres nœuds, avec le même numéro d'identification (le nœud s'ajoute à la liste des nœuds parcourus) [25][39].

Cette réponse contient la liste des nœuds parcourus par le message ROUTE REQUEST reçu. Si la cible possède une route vers le nœud initiateur, alors cette route est utilisée pour l'envoi de réponse (le chemin « destination- source »). Dans le cas contraire, deux solutions sont possibles :

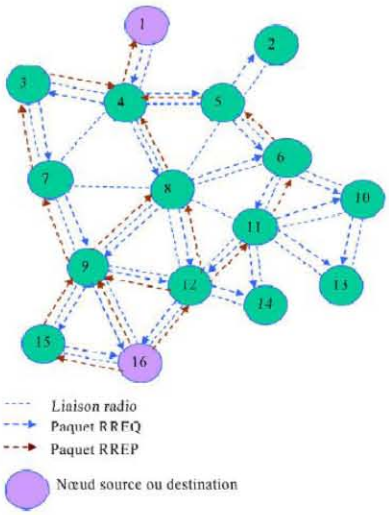
Le mécanisme de découverte de route peut être utilisé pour obtenir une route de la cible au nœud initiateur de la première requête. Si cette approche est utilisée ; la réponse ROUTE REPLY (RREP) doit être incluse « sur le dos » du nouveau paquet RREQ (piggyback) afin d'éviter toute boucle ; la liste des nœuds parcourus est inversée et utilisée comme route pour la réponse ROUTE REPLY. Cette approche est requise si le support du réseau est 802.11 afin de vérifier que toute la route fonctionne de manière bidirectionnelle (Le protocole DSR prend en compte les liens unidirectionnels).

Lorsque le nœud initiateur reçoit une réponse ROUTE REPLY, la route fournie est mise en cache afin de pouvoir la réutilisée ultérieurement (ajouter dans sa table de routage). Les paquets mis en tampon pour la cible sont finalement émis.

Différentes optimisations sont possibles. La plus importante est probablement la possibilité pour un nœud recevant un message ROUTE REQUEST pour une cible pour laquelle il possède une route en cache d'envoyer directement au nœud initiateur une réponse ROUTE REPLY contenant la route en cache concaténée à la liste des nœuds parcourus par le message ROUTE REQUEST.

Une seconde optimisation consiste à mettre en cache toute route qui est apprise de manière inopinée afin d'éviter une éventuelle découverte de route pour la destination. Ainsi, tout nœud intermédiaire acheminant une réponse ROUTE REPLY peut disposer gratuitement d'une route vers la cible ayant émis cette réponse, mais aussi vers les nœuds entre lui-même et la cible [47].

La Figure 2. 9 : **Processus de découverte de routes du protocole DSR [39]**Figure 2. 9 illustre le processus de découverte de route entre le nœud « 1 » et le nœud 16. Le tableau à droite indique l'ensemble des routes pour une destination quelconque et en particulier le nœud 16.



Cache de route du nœud 1

Destination	Routes
2	4-5
3	4
	4-8-9-7
	4-8-13-9-7
.	.
.	.
.	.
.	.
.	.
.	.
.	.
.	.
.	.
.	.
.	.
16	4 3 7 9
	4-3-7-9-12
	4-3-7-9-15
	4-8-9
	4-8-9-15
	4-8-12
	4-5-6-11-12

Figure 2. 9 : Processus de découverte de routes du protocole DSR [39]

Maintenance de route : L'opération de maintenance consiste dans un premier temps à déterminer si un lien est rompu. Cette opération peut être réalisée par la sous-couche MAC. Si au bout d'un certain nombre d'émissions aucun acquittement n'est reçu, le lien peut être considéré comme coupé. Un nœud détectant la rupture prévient l'ensemble des sources avec un paquet d'erreur (ROUTE ERROR). A la réception d'un tel paquet, les sources déterminent une nouvelle route si aucune autre n'est connue [25].

Lors de l'acheminement d'un paquet d'une source à une destination, la retransmission du paquet pour chaque saut est acquittée par le nœud intermédiaire recevant le paquet. Cet acquittement peut avoir plusieurs formes, actives ou passives. Si la couche MAC fournit un acquittement pour la réception des trames, cet acquittement est automatiquement utilisé comme acquittement du paquet ;

- Une approche plus directe consiste à demander explicitement un acquittement par l'envoi d'un simple message. Dans un tel cas, il est possible de considérer l'acquittement comme valide pour un temps limité et de ne pas requérir d'acquittement pour les prochains paquets émis dans cet intervalle de temps.
- Si un paquet n'est pas acquitté lors d'un saut, alors toute route dans la table de routage (le cache) passant par le nœud suivant est invalidée et un message ROUTE ERROR est émis à destination des nœuds utilisant le nœud suivant dans une de leurs routes afin de les prévenir que celles-ci ne sont plus valides.

Une opération appelée "sauvetage des paquets" peut aussi être effectuée dans le cas où le nœud détectant un lien coupé dispose d'une route alternative vers la destination. L'opération consiste à utiliser cette route en remplacement de la route spécifiée par l'émetteur. Le message ROUTE ERROR est toujours transmis [47].

A réception d'un tel paquet, les sources déterminent une nouvelle route si aucune autre n'est connue. La Figure 2. 10 illustre le processus de maintenance de route lorsque le lien entre les nœuds 9 et 16 est brisé après déplacement du nœud 9. Donc, les routes qui passent par le nœud 9 vont êtres supprimés [25] [39].

Plusieurs propositions pour optimiser le protocole DSR ont été faites et évaluées par les auteurs du protocole [48]. Ces propositions sont :

- La récupération : qui consiste, pour un nœud intermédiaire à utiliser une autre route de son propre cache lorsqu'un lien est brisé sur la route de la source d'un paquet de données ;
- La réparation gratuite de routes : correspond au processus dans lequel un nœud source qui reçoit un paquet RERR le diffuse aux autres nœuds. Cela permet aux autres nœuds de supprimer les routes invalides de leurs caches ;
- L'écoute de proximité : permet à un nœud d'acheminer plus rapidement vers la destination, un paquet qui ne lui est pas adressé. Pour le faire, il transmet un message de type RREP à la source avec cette nouvelle route. Toujours par écoute de paquets, un nœud pourrait ajouter toutes les informations de contrôle utiles à sa table de routage ; et
- La non propagation de paquets RREQ : concerne l'exécution d'une découverte de route s'arrêtant au niveau des voisins à un saut d'abord, avant de recourir à une diffusion dans tout le réseau. Cette proposition permet de réduire les paquets de contrôle dans les cas où les caches des voisins du nœud ont une route pour la destination prévue ou lorsque le nœud destination est un voisin [39].

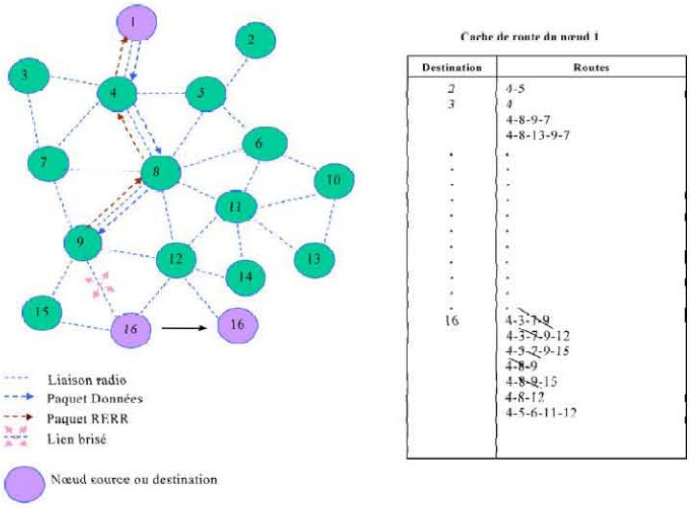


Figure 2. 10 : Processus de maintenance de routes du protocole DSR [39].

En conclusion, DSR est un protocole qui a l'avantage d'être relativement simple, fournissant de bons résultats. Pas de problème de surcharge sur le réseau et une meilleure gestion de l'énergie consommée.

Mais ce dernier présente des inconvénients: un délai additif avant toute communication et risques d'informations obsolètes dans les tables de routage des différents nœuds [47].

II.5.2.2 Le protocole AODV

L'AODV est basé sur l'algorithme de routage DSDV (Destination Sequenced Distance Vector Routing Algorithm). Les objectifs de l'algorithme AODV peuvent être résumés comme suit :

1. Eliminer le besoin d'une diffusion globale des informations de routage dans tout le réseau ad hoc;
2. Minimiser le temps d'attente quand de nouvelles routes sont demandées.

Hypothèses architecturales

L'algorithme AODV ne fait aucune hypothèse spécifique sur la couche physique autre que les suivantes :

- Les nœuds voisins peuvent écouter leurs diffusions respectives : les nœuds voisins sont les nœuds séparés par une distance inférieure a la portée du signal de chaque nœud;
- Les liens entre les nœuds voisins sont symétriques : si le nœud A peut entendre la transmission du nœud B, ceci implique que le nœud B peut entendre la transmission du nœud A.

Principe de fonctionnement

Pour que le routage fonctionne efficacement, chaque nœud dans le réseau doit se conformer à un ensemble de règles qui collectivement, constituent l'algorithme AODV. Les règles de base sont :

- Tous les nœuds acceptent de faire router les paquets de données et les informations de routage même s'ils ne sont pas directement impliqués dans le paquet transmis.
- Un nœud source souhaitant communiquer avec un nœud destinataire doit d'abord consulter sa table de routage. S'il ne trouve pas localement toutes les informations sur la route à suivre; il diffusera un message de demande de la route (RREQ : route request message) aux nœuds voisins.
- Comme le message RREQ se propage dans le réseau, chaque nœud recevant le message doit mettre à jour sa table de routage en créant une entrée dans la table pour le nœud source. Cette opération est appelée "reverse path setup" schématisée dans la Figure 2. 11.
- Si un nœud reçoit un message RREQ pour une destination autre que lui même, il consultera sa table de routage. S'il trouve une route vers la destination demandée, il envoi en unicast un paquet (route reply paquet ou RREP) au nœud source via le "reverse path". Le paquet RREP doit avoir le même numéro de séquence (Seq-no) que le paquet RREQ.
- Un nœud recevant un message RREQ pour lui même, doit envoyer en unicast un message RREP au nœud source avec le même numéro de séquence « Seq-no » que celui du message RREQ original.

- Comme le paquet RREP se propage vers la source via le chemin “reverse path", chaque nœud sur le chemin recevant le paquet doit mettre à jour sa table de routage en créant une entrée pour la destination, ainsi est formée la route du nœud source au nœud destination comme le montre la Figure 2. 11.

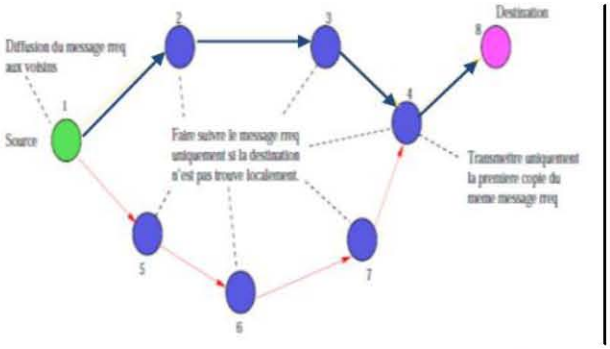


Figure 2. 11 : Propagation d’un message RREQ [49]

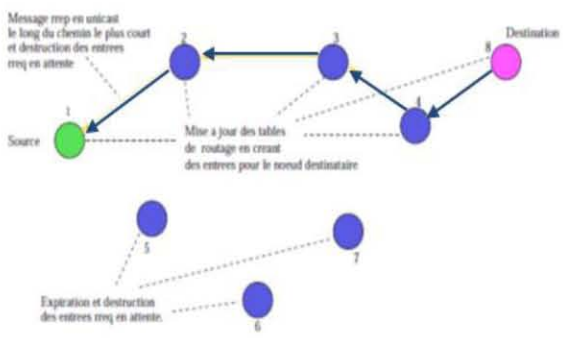


Figure 2. 12: Propagation d’un message RREP [49]

- Les nœuds ayant reçus un paquet RREQ mais n’ayant pas envoyés un message RREP au nœud source vont expirer et supprimer l’entrée réservée au nœud source dans leurs tables de routage pour le chemin “reverse path", après un certain temps.

  L’hypothèse faite dans ce cas est que soit il n’existe pas de chemin du nœud source au nœud destination, ou que ce nœud n’est pas dans le plus court chemin entre la source et la destination.
- Un nœud recevant plusieurs paquets RREQ et qui ne peut pas envoyer de message RREP au nœud source, doit faire suivre uniquement la première copie du même message RREQ et ignorer les autres. Les paquets RREQ similaires ont le même nœud source et le même numéro Seq-no. Ces paquets similaires peuvent se produire à cause des chemins multiples du nœud source au nœud qui les reçoit.
- L’hypothèse ici est que la première copie du message prend toujours le plus court chemin, i.e. le plus petit nombre de sauts (hops) et de ce fait elle doit être préférée.

  Un nœud se trouvant avec plusieurs routes vers une certaine destination, soit a travers plusieurs chemins “route replies" ou a travers des messages Hello périodiques, doit toujours choisir une route contenant le plus petit nombre de sauts a moins que la réponse la plus récente ait un nouveau numéro de séquence. Des durées de vie sont associées aux informations de routage. Les routes étant non utilisées expirent après une certaine période de temps et sont supprimées de la table de routage.
- Chaque nœud doit périodiquement diffuser un message Hello à des nœuds voisins.
- Un nœud recevant un message Hello, met à jour en conséquence sa liste de voisins et sa table de routage. Cela assure la mise à jour des routes même quand les nœuds sont en mouvement.
- Des durées de vie sont également associées aux listes de voisins des nœuds.

Quand un nœud ne reçoit pas un message Hello de l’un de ses voisins, il suppose que ce voisin est hors portée et de ce fait, il le supprime de sa liste de voisins avec les informations de routage le concernant [49].

II.5.3 Protocoles Hybrides

Une troisième catégorie appelée les protocoles hybrides permet de combiner les deux concepts : celui des protocoles proactifs et celui des protocoles réactifs. Généralement, le réseau est divisé en deux zones et le principe est d’utiliser une approche proactive pour avoir des informations sur les voisins les plus proches, qui se trouvent au maximum à deux sauts du nœud mobile. Une approche réactive est utilisée au-delà de cette zone prédéfinie afin de chercher des routes.

L’avantage de cette troisième catégorie est le fait qu’elle s’adapte bien aux réseaux de grandes tailles. Cependant, cette approche a comme inconvénient de cumuler les points faibles des protocoles réactifs et ceux des protocoles proactifs, tels que les messages de contrôle périodique et le coût d’établissement d’une nouvelle route. Il existe plusieurs protocoles connus appartenant à cette catégorie de protocoles hybride, citons CBRP (Cluster Based Routing) [39] et ZRP (Zone Routing Protocol) [50].

II.5.3.1 Le protocole ZRP

ZRP [51] est le plus populaire de la classe hybride. Ce protocole découpe la topologie du réseau en zones de routage. La zone de routage pour chaque nœud est un sous-ensemble du réseau à l’intérieur duquel tous les nœuds peuvent être atteints dans un rayon maximal de « r » sauts [44].

Une première zone est celle dans le voisinage de chaque nœud (voisins se trouvant à une distance inférieure ou égale au rayon « r » de la zone), elle est appelée Intrazone dans laquelle les paquets seront routés en utilisant une approche proactive (Figure 2. 13).

Une seconde zone est la zone extérieure à un nœud, appelée Interzone, c'est-à-dire l'ensemble des nœuds qui se trouvent à un nombre de sauts supérieur à « r » dans laquelle, ZRP s'appuie sur une technique réactive. Les nœuds qui sont exactement à la distance « r » sont appelés nœuds périphériques [25].

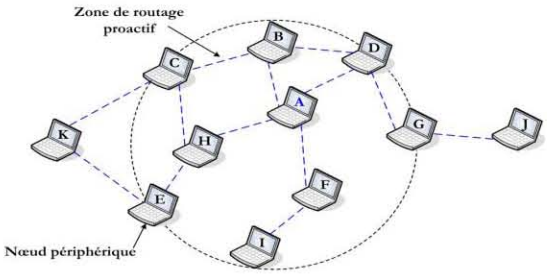


Figure 2. 13 : Zone de routage du nœud A définie par ZRP [15]

L'IntraZone est construite à partir de la découverte des voisins qui se fait en utilisant, soit directement les protocoles MAC, soit le protocole de détection de voisins NDP (Neighbor Discovery Protocol). Le NDP repose sur la transmission de messages Hello par chaque nœud. Ce dernier est libre de choisir les nœuds selon divers critères tels que la force du signal ou la fréquence des messages [39].

Pour déterminer un chemin pour joindre la destination, deux protocoles de routage vont être employés suivant la zone dans laquelle se trouve la destination. Ainsi, si la destination se situe dans l'Intrazone, le protocole de routage proactif IntrAzone Routing Protocol est utilisé. Dans le cas contraire (destination en dehors de cette zone), le protocole de routage réactif IntErzone Routing Protocol est utilisé [25].

Une fois que l'information de routage locale est collectée, le nœud diffuse des messages de découverte périodiquement pour maintenir son voisinage à jour grâce au protocole proactif IARP (IntrAzone Routing Protocol) [52] fondé sur un protocole à état de liens [39]. A l'aide des informations diffusées, les nœuds construisent la topologie et déterminent les routes vers les nœuds jusqu'à une distance « r ».

Pour limiter la propagation des paquets de contrôle sur la totalité du réseau, la source initialise le champ TTL avec à la valeur de « r » (i.e. le nombre de saut maximum auquel se limite l'Intrazone).

Chaque fois qu'un nœud reçoit un tel paquet, il met à jour sa table de routage puis décrémente de 1 le champ TTL. Dés que la valeur de ce champ devient nulle, le paquet est supprimé, sinon il est propagé [25].

La communication entre les différentes zones est assurée par le protocole réactif IERP (Inter-zone Routing Protocol) [53] qui permet la découverte de routes pour les nœuds situés en dehors de la zone de routage [39].

Lorsque la source ne connaît pas de chemin vers la destination (celle-ci ne se trouve pas dans l'Intrazone), il utilise le protocole IERP (IntErzone Routing Protocol) responsable uniquement des communications entre les différentes zones selon une approche réactive pour déterminer un chemin jusqu'à cette destination.

La source détermine un ensemble de nœuds périphériques à son Intrazone qu'elle utilisera pour déterminer un chemin jusqu'à la destination, tout en réduisant le délai et le surcoût pris par la recherche. Lors de la réception de la requête de demande de création de route, les nœuds périphériques ajoutent leur identifiant dans l'entête de la requête. Ensuite, deux procédures sont appliquées selon que ces nœuds connaissent une route vers la destination ou non (Figure 2. 14):

- La destination est dans l'Intrazone d'un nœud frontière : une réponse est envoyée à la destination en prenant le chemin inverse contenu dans l'entête de la requête.
- La destination est en dehors de l'Intrazone d'un nœud périphérique: la requête est propagée à l'ensemble de ses nœuds périphériques et l'opération se répète jusqu'à déterminer un chemin [25].

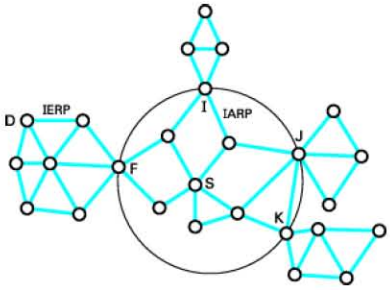


Figure 2. 14 : Principe de fonctionnement du protocole ZRP tiré de [54]

Prenons l'exemple de la Figure 2.15 ou le nœud 11 a des paquets de données à transmettre au nœud 1, il commence par vérifier si ce dernier est situé dans la zone de routage (IntrAzone) , si c'est le cas , il délivre alors les paquets de données, sinon il diffuse les paquets requêtes (i.e. RREQ) aux nœuds périphériques situés à la frontière de la zone de routage (nœuds 4, 5, 6, 7, 9, 10, 13, 14 et 15). Si un nœud périphérique possède une route vers le nœud destination dans sa table de routage, il renvoie un paquet de réponse (i.e. RREP) au nœud source (nœud 11). Autrement, le nœud source rediffuse le paquet RREQ aux nœuds périphériques jusqu'à ce que le nœud destination soit localisé.

Dans la Figure 2. 15, les nœuds 4 et 5 qui ont le nœud 1 dans leur table de routage retournent des paquets RREP au nœud source (nœud 11) [39].

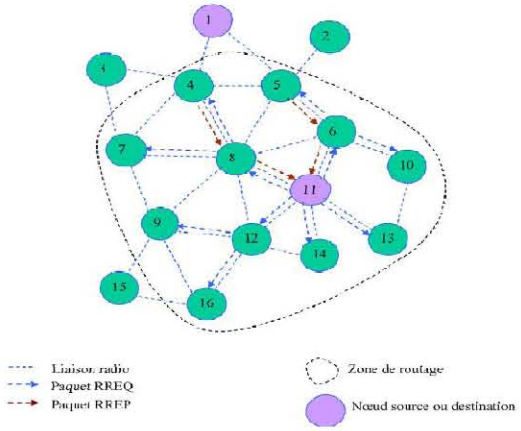


Figure 2. 15 : Recherche de chemin du protocole ZRP [39]

II.5.3.2 Le protocole CBRP

Dans le protocole réactif CBRP (Cluster Based Routing Protocol) [55], l'ensemble des nœuds du réseau est décomposé en groupes (clusters). Un cluster est défini par un ensemble de nœuds et possède un nœud nommé nœud chef ou Cluster Head (CH). Un nœud peut être élu chef de groupe (CH), ou passerelle (gateway) entre groupes (Figure 2. 16) suivant sa situation dans le réseau (i.e. sa visibilité des autres nœuds). Le CH possède généralement des ressources spéciales par rapport aux autres nœuds du réseau [4].

Méthode de formation des clusters : Le principe de formation des clusters (groupes) est le suivant :

1. Un nœud p sans statut (i.e. ni membre, ni CH), active un temporisateur et diffuse un message "Hello".
2. Lorsqu'un CH reçoit ce message, il envoie immédiatement une réponse à l'émetteur.
3. Lors de la réception de réponse, le nœud p rejoint le cluster correspondant et devient "membre".
4. Si le nœud p ne reçoit pas une réponse après un certain timeout :
   a. S'il possède un lien bidirectionnel vers au moins un voisin, il se considère lui-même comme un CH.
   b. Sinon p (toujours sans statut) répète la même procédure.

A cause des changements rapides de la topologie, l'attente des nœuds sans statut est très courte.

Afin de sauvegarder la répartition des nœuds dans les groupes, chaque nœud maintient une table des voisins où chaque entrée est associée à un voisin ; elle indique l'état du lien (uni ou bidirectionnel) et le statut du voisin (membre ou CH).

Le représentant de groupe (CH) maintient les informations des membres qui appartiennent à son cluster. Il possède aussi une table des clusters adjacents où chaque entrée contient les informations d'un cluster voisin : l'identificateur du cluster, l'identificateur du nœud de liaison à travers lequel le cluster peut être atteint [58].

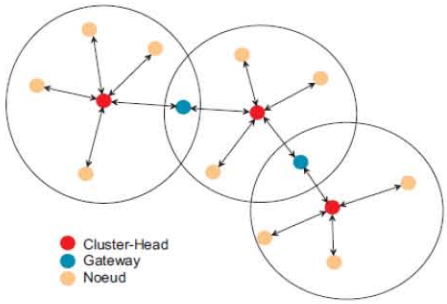


Figure 2. 16 : Les différents types de nœuds dans CBRP [**4**]

Dans ce protocole réactif, lorsqu'un nœud source veut envoyer des données à un nœud destination, il doit diffuser une requête demandant un chemin uniquement aux CHs des clusters voisins.

Chaque CH vérifie l'existence du nœud destination dans son cluster (en utilisant la table de membres). Le CH de cluster qui contient le nœud destination, y envoie directement la requête de demande de route. Dans le cas contraire, la requête est rediffusée aux CHs des clusters voisins. Les adresses des CHs sont incluses dans cette requête. Le CH ignore toute requête déjà traitée.

Quand la destination reçoit la requête, elle répond par l'envoi du chemin qui a été sauvegardé dans le paquet de la requête. Dans le cas où le nœud source ne reçoit pas de réponse après une certaine période, il envoie de nouveau une requête de demande de route.

Lors du routage des données, si un nœud détecte qu'un lien est défaillant, il envoie un message d'erreur à la source et il applique un mécanisme de réparation locale. Dans ce mécanisme, si un nœud « n » trouve qu'un nœud suivant « m » ne peut pas être atteint, il essaie de vérifier si le nœud « m » ou le nœud suivant «m» dans le chemin peut être atteint à travers un autre nœud voisin. Si l'un des deux cas est vérifié, les données sont envoyées en utilisant le chemin réparé [54].

II.5.4 Protocoles hiérarchiques

Les protocoles de routage à plat fonctionnent bien quand le réseau comprend un nombre limité de nœuds, mais deviennent ingérables lorsque le réseau devient important (i.e. un grand nombre de nœuds). Dans ce dernier cas, et pour maitriser la gestion du réseau, on doit le structurer. La structuration la plus connue est la hiérarchie. La technique de hiérarchisation sert à partitionner le réseau en sous ensembles afin de faciliter sa gestion surtout le routage, qui se réalise à plusieurs niveaux [25][56].

Les notions de partitionnement et de groupes sont très répandues dans les réseaux mobiles ad hoc. La formation de groupes améliore considérablement les performances des réseaux. Le partitionnement, quant à lui, peut être exploité dans les réseaux de grande taille afin de réaliser un routage hiérarchique, ce qui réduit le contrôle des données de routage. Le problème principal du routage hiérarchique dans les réseaux sans fil est la mobilité et la gestion de la localisation [58].

Dans ce type de protocoles, la vue du réseau devient locale; des nœuds spéciaux peuvent avoir des rôles supplémentaires. Nous distinguons deux types de groupes de nœuds : la zone et le cluster (Figure 2. 17).

Le CH est élu suivant différents critères et informations sur le réseau (le niveau de l'énergie, la connexion avec les autres nœuds, la position géographique, etc).

Une zone est définie par un ensemble de nœuds mais ne possède pas un CH. Ainsi, un cluster est une sous-classe d'une zone. La construction des groupes (zones ou clusters) s'appuie sur des informations sur le réseau, exigeant donc son instrumentation [25][56].

Une autre structure utilisée est la chaîne [57]. Le principe d'une chaîne est qu'un nœud ne peut communiquer qu'avec deux voisins. Nous trouvons aussi des structures qui combinent les groupes et les chaînes. En se basant sur une architecture hiérarchique. Plusieurs protocoles de routage pour les réseaux ad hoc de grande taille ont été proposés. Dans la suite nous en détaillerons quelques uns.

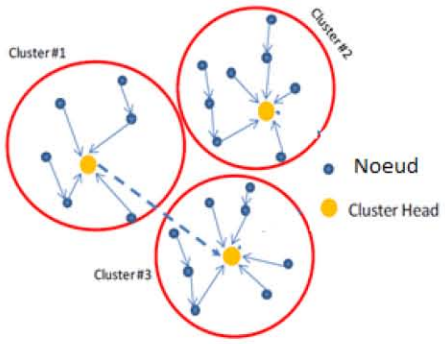


Figure 2. 17 : Architecture en cluster [56]

II.5.4.1 Le protocole HSR

L'idée proposée par le protocole HSR (Hierachical State Routing) ou "Routage à Etat Hiérarchique" [50] consiste à classer les nœuds suivant une structure hiérarchique en arbre. Il combine les notions de groupes dynamiques et de niveaux hiérarchiques avec une gestion efficace de localisation. Dans le HSR, l'image de la topologie du réseau est sauvegardée sous forme hiérarchique. Le réseau est partitionné en un ensemble de groupes, dont l'union donne le réseau entier.

Les nœuds sont tout d'abord assemblés en groupes (clusters), puis un nœud représentant est élu pour chaque groupe. Les représentants des groupes dans un niveau *n*, deviennent des membres dans le niveau *n + 1*. Ces nouveaux membres, s'organisent en un ensemble de groupes de la même manière du niveau bas, et ainsi de suite pour le reste des niveaux [58].

Une méthode simple consiste à utiliser pour chaque groupe l'identifiant de son représentant. Chaque nœud peut donc être désigné par le *k*-uplet des identifiants des différents groupes auquel il appartient. Ce *k*-uplet est appelée HID (Hierarchical ID), en commençant par le niveau le plus haut jusqu'au plus bas (c'est-à-dire l'identifiant propre du nœud).

A noter que le découpage en groupes n'implique pas nécessairement des intersections nulles entre ceux-ci (ce qui conduit donc à l'existence de HID multiples pour certains nœuds) [25].

Plusieurs algorithmes de partitionnement sont proposés dans la littérature pour la création dynamique des groupes et l'élection des représentants de groupes. Le but principal du partitionnement du HSR est l'utilisation efficace des médiums de communication et la réduction du contrôle de routage effectué par la couche réseau (i.e. la sauvegarde des tables de routage, le traitement et la transmission des données).

La Figure 2. 18 illustre l'application de ce mécanisme de partitionnement dans un réseau de 13 unités mobiles. Le réseau est décomposé en 4 groupes, qui sont : G0-1, G0-2, G0-3, et G0-4. Ces groupes forment le niveau le plus bas de la hiérarchie (niveau 0). A partir de ce niveau, les niveaux qui suivent (niveaux 1 et 2), sont formés. Cela est fait en prenant l'ensemble des représentants de groupes et le décomposer en groupes de la même manière que précédemment [58].

Dans la décomposition en groupe, on peut avoir 3 types de nœuds (Figure 2. 18):

- un nœud représentant du groupe, appelé aussi tête de groupe (par exemple les nœuds 1, 2, et 3)
- un nœud de liaison, qui relie deux groupes (exemple, les nœuds 7 et 9)
- un nœud interne qui n'a aucun rôle spécial (exemple, les nœuds 4, 11 et 12).

Un nœud de liaison 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 (suivant les groupes).

En générale, cela ne pose aucun problème, car le nœud peut être atteint à travers l'une de ces adresses hiérarchiques qui 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, par exemple: <1, 1, 7> est une adresse du nœud de liaison d'ID 9.

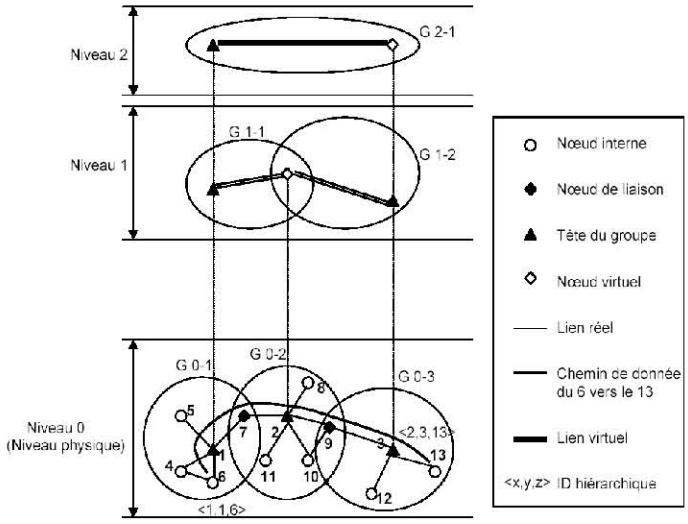


Figure 2. 18: Le partitionnement du réseau en groupes [58]

L'avantage de l'adressage hiérarchique permet à chaque nœud de mettre à jour dynamiquement et localement son HID lors de la réception des données de mise à jour du routage provenant des nœuds de niveau supérieurs et qu'il est suffisant pour la délivrance des paquets de données à une destination, indépendamment de la localisation de la source, et cela en utilisant la table HSR.

Prenons comme exemple le nœud 6 (Figure 2. 18) comme source, et le nœud 13 comme destination. Les adresses de ces nœuds sont respectivement : HID(6) = <1, 1, 6> et HID(13) = <2, 3, 13>. Pour acheminer une information du nœud 6 vers le nœud 13, le nœud 6 envoie l'information au nœud supérieur, qui le suit hiérarchiquement (i.e. le nœud d'ID 1). Le nœud 1 délivre l'information au nœud 3 qui suit le nœud destination dans l'ordre hiérarchique. Un "lien virtuel" existe entre les nœuds 1 et 3, matérialisé par le chemin (1, 7, 2, 9, 3). Par conséquent, l'information suivra ce chemin pour atteindre la destination.

Dans la dernière étape, le nœud 3 délivre l'information au nœud 13 en suivant le chemin hiérarchique qui le relie à la destination (dans notre cas, ce chemin se réduit en un seul saut) [58].

Pour que le protocole fonctionne, chaque nœud membre de niveau *n* doit connaitre une route vers son représentant de niveau *n*. Chaque représentant doit connaitre les routes vers les membres de son niveau [25].

En plus de la décomposition (partitionnement) en groupes basé sur les relations géographiques entre les différents nœuds, le protocole HSR utilise aussi un partitionnement logique. Ce partitionnement est basé sur des relations logiques qui peuvent exister entre les nœuds du réseau (comme par exemple l'appartenance à une même société), et il joue un rôle clé dans la gestion de la localisation. En plus des adresses physiques, une adresse logique de la forme <*subnet, host*>, est associée à chaque nœud. Ces adresses ont un format similaire au format IP : elles peuvent être vues comme des adresses IP privées dans le réseau mobile.

Chaque sous-réseau (*subnet*) correspond à un groupe particulier d'utilisateurs (précédemment défini) qui possède un serveur de gestion de localisation dit LMS (Location Management Server). Chaque sous-réseau possède un nœud particulier (Home Agent) qui associe à chaque nœud local un HID. Les HID de ces mêmes "Home Agent" sont transmis aux niveaux hiérarchiques supérieurs.

Quand la couche de transport délivre au réseau un paquet contenant l'adresse IP privée, le réseau doit trouver, à partir de l'adresse IP, l'adresse hiérarchique basée sur les adresses physiques [58].

Lorsqu'un message doit être transmit d'un nœud V à W, V utilise l'adresse logique de W (la seule qu'il connaisse) pour en déduire l'adresse logique du Home Agent correspondant. Il transmet alors sa requête au représentant de son groupe qui la fait remonter vers le représentant du niveau supérieur. On remonte les niveaux jusqu'à ce qu'un représentant soit capable de localiser le "Home Agent" (dont les HID sont connus des niveaux hiérarchiques supérieurs).

Une réponse redescend alors vers V lui indiquant le HID en question. V délivre son message au Home Agent (par le même procédé de remonté puis de redescente dans les groupes hiérarchiques) qui est alors en mesure de le renvoyer à la destination finale W.

L'avantage de HSR est d'une part de permettre un découpage logique du réseau en fonction d'éventuelles relations entre utilisateurs (appartenance à un même organisme ou à une même société), d'autre part de limiter les messages de contrôle, puisque l'information est localisée en certains nœuds uniquement.

Néanmoins ce choix peut poser certains problèmes tels que la mobilité de nœuds à faible rôle hiérarchique et la disparition des représentants de haut niveau entraîne une désorganisation importante [25].

II.5.4.2 Le protocole CGSR

Le protocole CGSR (Clustrehead Gateway Swith Routing) [59] utilise principalement l'algorithme de routage DSDV, décrit précédemment et est basé sur une architecture de réseau basée sur des groupes. Le réseau est ainsi décomposé en groupe, comme illustré sur la Figure 2. 19 [60].
Dans chaque cluster un nœud, appelé ClusterHead (CH), est chargé du contrôle de cluster. Le cluster est formé des nœuds situés à la portée de communication de son CH. Un nœud qui appartient à plusieurs clusters en même temps (i.e. la portée de communication de plus d'un CH) est appelé nœud de liaison (gateway).

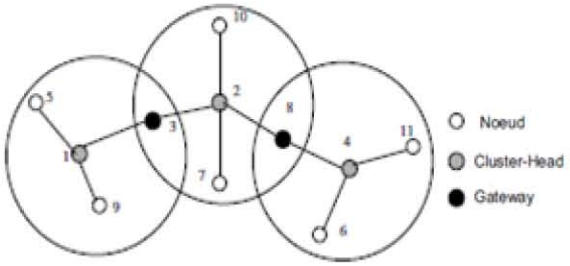


Figure 2. 19 : illustration de CGSR [60]

Le principe de formation des groupes est identique à celui de protocole CBRP (voir section II.5.3.2). La Figure 2. 20 illustre un réseau composé de trois groupes avec leurs tables de routage [58].

Afin de régler le problème des changements fréquents de topologie dans les réseaux ad hoc, le protocole CGSR utilise un algorithme appelé LCC (Least Cluster Change).

Dans cet algorithme, un changement de CHs intervient seulement dans le cas d'une fusion de deux clusters, ou dans le cas où un nœud sortirait complètement de la portée de tous les CHs (représentants du réseau).

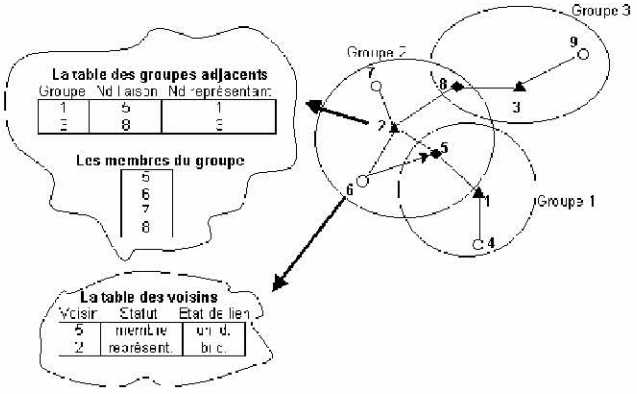


Figure 2. 20: L'organisation du réseau avec le protocole CBRP [58].

Le routage des informations dans CGSR se fait de la manière suivante :

1. Le nœud source transmet ses paquets de données à son CH (représentant de groupe).
2. Le CH envoie les paquets aux nœuds de liaison, qui relient ce CH avec le CH suivant dans le chemin qui existe vers la destination.
3. Le processus se répète, jusqu'à ce que ces paquets atteignent le CH du cluster dans lequel se trouve la destination. Ce CH transmet alors les paquets reçus vers le nœud destination.

La Figure 2. 21, donne le chemin de routage des paquets de données entre le nœud 9, et le nœud 18 [60].

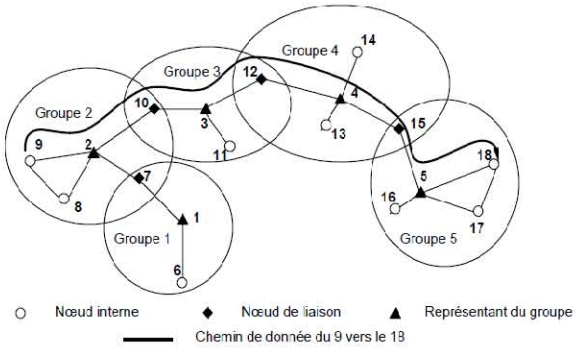


Figure 2. 21 : Un exemple d'acheminement d'information dans le CGSR [60].

Chaque nœud possède deux tables. La première est la table de membres du cluster, qui associe à chaque nœud du réseau l'identificateur d'un CH. Chaque nœud diffuse cette table d'une façon périodique et met à jour sa propre table (après la réception des données de mise à jour provenant d'un autre nœud), en utilisant l'algorithme DSDV. La deuxième est la table de routage (vecteur de distance). Elle détermine le nœud suivant à emprunter dans le cluster correspondant au cluster destinataire.

Lors de la réception d'un paquet, le nœud intermédiaire trouve le représentant de groupe le plus proche dans le chemin envisagé vers la destination (*D* par exemple), et cela en utilisant sa table des membres de groupes et sa table de routage. Par la suite le nœud consulte sa table de routage, pour trouver le nœud suivant afin d'atteindre ce représentant *D*. Les paquets seront transmis alors, au nœud trouvé [60].

Notons que cette manière de routage assure un procédé déterministe et efficace pour l'acheminement des informations, cependant un chemin choisit peut ne pas être optimal [25]. On peut remarquer dans l'exemple précédent, si on suppose que tous les coûts des liens sont égaux, le chemin (9,2,10,3,12,4,15,5,18), ne représente pas le meilleur chemin (ici (9,2,10,3,12,4,15,18) est meilleur). Cela est dû au fait, que tous les nœuds applique la même stratégie, le nœud 15 trouve (en utilisant sa table de routage) que le nœud suivant correspondant au nœud 18 est le nœud 18 lui-même. Le nœud 15, consulte sa table de membres de groupes pour connaître le représentant du groupe associé à 18, le nœud trouvé est alors celui de l'ID 5 ce qui fait que les paquets passent par 5 et ne passent pas directement vers la destination [60].

II.5.5 Routage géographique

Cette classe de protocoles de routage se différencie de ceux précédemment présentés par l'utilisation d'une donnée supplémentaire dans la recherche des routes : la position géographique des nœuds du réseau.

L'idée dans les protocoles de *routage géographique* est d'utiliser des informations géographiques externes (obtenues par GPS: Global Positioning System [61]) pour trouver des routes afin d'acheminer les paquets.

Un nœud désirant envoyer un paquet à un autre doit en premier lieu obtenir la localisation de celui-ci. Pour accéder à une telle information, le nœud considéré peut émettre par diffusion une requête de localisation et attendre la réponse d'un nœud possédant l'information souhaitée [62].

Grace à un système de localisation, les nœuds connaissent leur position et également la position de tous les autres nœuds du réseau. Avec ces hypothèses, il est facile de concevoir qu'un nœud peut facilement choisir parmi ses voisins un nœud relais pour acheminer un paquet dont il connaît la destination finale et donc aussi sa position. Il est très simple d'envisager des critères de sélection parmi ces voisins; donnons quelques exemples parmi les heuristiques les plus classiques cités dans la littérature [54]:

- Une heuristique particulièrement simple est celle qui choisit le voisin qui permet de se rapprocher le plus de la position de la destination finale.
- Une autre heuristique qui permet de minimiser le nombre de sauts dite l'heuristique de la progression maximale est celle qui choisit le voisin qui permet de progresser le plus en direction de la destination finale (schématisée sur la Figure 2. 22).
- La dernière heuristique très simple est celle qui choisit le voisin le plus proche en direction de la destination finale. Elle permet de minimiser l'énergie pour relayer le paquet vers sa destination finale.

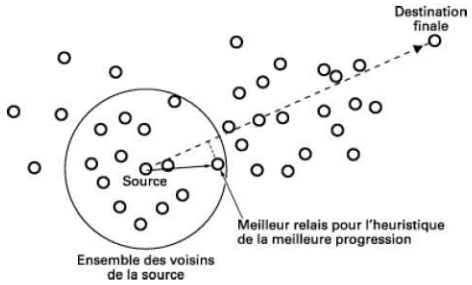


Figure 2. 22: Routage géographique avec l'heuristique de la meilleure progression vers la destination

Parmi les algorithmes de cette classe on trouve: DREAM (Distance Routing Effect Algorithm for Mobility), LAR (Location-Aided Routing) [4].

II.5.5.1 Le protocole LAR

Le protocole LAR (Location-Aided routing) [63] appelé *"Routage Aidé par la Localisation"* est un protocole de routage à la demande (routage réactif) basé sur l'utilisation des informations de localisations fournies par le système de positionnement global GPS (Global Positioning System) pour y découvrir des routes [64][65].

Ce protocole procède d'une manière très similaire au protocole DSR. La principale différence entre les deux protocoles, réside dans le fait que le LAR utilise les informations de localisation, fournies par GPS, dans le but de limiter l'inondation des paquets de requête de route (i.e. l'utilisation d'une estimation de la position afin d'accroitre l'efficacité de la procédure de découverte de route). Afin d'assurer cela, deux approches peuvent être utilisées [60][66]:

Dans la première approche, le nœud source définie une région circulaire dans laquelle la destination peut être localisée dite *"expected zone"* (Figure 2.23) dont la position et la taille, sont estimées en se basant sur :

- la dernière 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 (« *request zone* ») qui correspond à la zone d'inondation partielle.

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 (Figure 2. 23).

Dans la deuxième approche, le nœud source calcule la distance qui le sépare de la destination, et l'inclut dans le paquet de requête de route. Ce dernier est envoyé par la suite aux nœuds voisins. Quand un nœud reçoit le paquet de requête, il calcule la distance qui le sépare de la destination et la compare avec la distance contenue dans le paquet reçu. Dans le cas où la distance calculée est inférieure ou égale à la distance reçue, le nœud envoie le paquet reçu. Lors de l'envoi, le nœud met à jour le champ de distance avec sa propre distance qui le sépare du nœud destination [58][60].

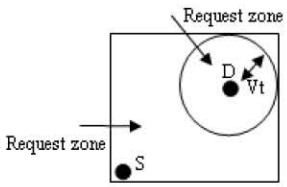


Figure 2. 23: Concept de request zone et exepted zone dans le protocole LAR [60]

Dans les deux méthodes, si aucune réponse de route n'est reçue après une certaine période (timeout), le nœud source rediffuse une nouvelle requête de route en utilisant une diffusion pure (sans limitation) [58].

II.5.5.2 Le protocole DREAM

Le protocole de routage DREAM (Distance Routing Effect Algorithm for Mobility) est conceptuellement proche des protocoles de routage proactifs. Il intègre les données dans les paquets de recherche de route [67].

Chaque nœud du réseau dispose d'une table des positions qui peut être assimilée aux tables de routage des nœuds utilisées par un protocole proactif [4]. Ces tables ne contiennent pas le prochain nœud ou le chemin

pour joindre une destination mais les informations de localisation de chaque nœud. Lorsqu'un nœud veut transmettre un paquet de données, il utilise les informations de localisation concernant la destination dans sa table et envoie le paquet uniquement dans sa direction [25].

Les mises à jour effectuées sur cette table concernent la position des autres mobiles dans le réseau, ainsi que leur vitesse et l'instant de mise à jour de ces informations. La fréquence de mise à jour des informations (échange des informations de localisation entre les nœuds) pour un mobile dépend de deux paramètres :

- *La distance* : c'est-à-dire son éloignement par rapport au nœud mettant à jour sa table. Par conséquent, chaque nœud a besoin de mettre à jour sa localisation moins souvent vis-à-vis des nœuds éloignés [25]. En effet, plus des nœuds sont distants, moins leur déplacement relatif est important, mais la route est moins stable car il y a beaucoup de sauts entre eux.
- *La mobilité* : la nature du déplacement effectué (lent ou rapide). Tout nœud qui se déplace doit mettre à jour sa position. Plus un nœud est mobile, plus les informations le concernant doivent être actualisées.

A base de ces deux paramètres, le volume du trafic de contrôle se trouve considérablement réduit par rapport à un protocole réactif pur et les fréquences d'émission des paquets de mise à jour sont variables et dépendent de la localisation relative d'un nœud et sa mobilité [4].

Chaque nœud émet périodiquement les informations de position le concernant (ses coordonnées, sa vitesse…). Pour réguler la distance de propagation de ses messages de contrôle, chaque nœud marque le paquet par une certaine distance. Lorsqu'un nœud le reçoit, il calcule la distance que le paquet a voyagé. Si elle est plus grande que celle marquée dans le paquet, il est supprimé, sinon il est propagé.

Ces paquets sont transmis d'autant plus souvent que la distance marquée est faible. La fréquence à laquelle les nœuds émettent les paquets de contrôle est fonction de la vitesse de déplacement du nœud lui-même. Plus il se déplace rapidement, plus il transmet souvent les informations sur sa position.

Lorsqu'un nœud a besoin d'envoyer un paquet à un autre nœud, il détermine l'ensemble des voisins permettant d'atteindre la destination avec une probabilité p. Lorsqu'un nœud reçoit un paquet, il vérifie s'il en est le destinataire. Si c'est le cas, il regarde si le paquet est un acquittement ou un paquet de données. Chaque nœud destination acquitte un paquet de données pour permettre à la source de savoir qu'il est correctement transmis. En cas de réception d'un même paquet plusieurs fois, il est acquitté à chaque réception (cas de perte d'acquittement). S'il n'est pas la destination, il propage ce paquet [25].

Dans le cas où la source envoie les données en spécifiant les nœuds suivants (en se basant sur les localisations), un *timer* associé à la réception des acquittements est activé. Si aucun acquittement n'est reçu avant l'expiration du timeout, les données seront retransmises en utilisant une diffusion ordinaire [60].

Dans la Figure 2. 24, un nœud A pour déterminer la route à suivre pour atteindre un nœud J utilise sa table de position. Dans cette table, il possède la localisation de J à l'instant t_1 ainsi que sa vitesse v_J, ce qui permet au nœud A de 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 ($t_2 \rightarrow t_1$). Le nœud A détermine ensuite un cône dont il est le sommet et donc la zone probable de présence de J est la base. Le nœud A envoie ensuite les informations à transmettre à tous les nœuds inclus dans le cône qui retransmettent jusqu'à atteindre J. A chaque retransmission, la route est mise à jour dans le paquet afin que J mémorise le chemin de retour (en sens inverse) pour l'utiliser pour retourner les acquittements de trame. Si plusieurs chemins sont possibles, J choisit celui comportant le moins de sauts.

Le protocole DREAM utilise souvent 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 l'inondation (le flooding) sur l'ensemble du réseau [4].

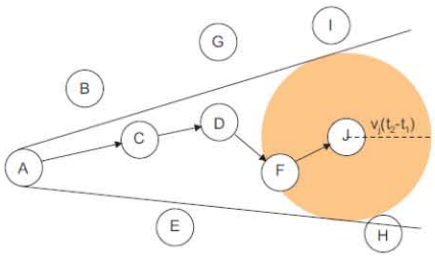


Figure 2. 24: Le principe de recherche de route dans le protocole DREAM [**4**]

II.6 Conclusion

Les protocoles de routage best-effort discutés peuvent être classés selon les critères de : la dissémination de l'information de routage, la hiérarchisation du réseau, l'utilisation d'éléments de localisation, ou par combinaison de ces critères. Le trafic additionnel, engendré par les informations de routage, consomme énormément de la bande passante du réseau et peut être source de plusieurs congestions et collisions. Cette diminution de la bande passante du réseau restreint le nombre de flux pouvant transiter sur le réseau [25].

Les protocoles proactifs (cf. § II.5.1) essaient de réduire le nombre d'informations de routage échangées sur le réseau en utilisant l'optimisation de relais multipoints comme le cas du protocole OLSR.

Les protocoles réactifs (cf. § II.5.2), quant à eux, réduisent les informations nécessaires au routage en diminuant l'impact sur la bande passante et ne gérer qu'un trafic de contrôle qu'en cas de besoin de routes. Pour cela, les nœuds maintiennent plusieurs routes dans leur table de routage, comme AODV par exemple.

La réduction des informations de routage des protocoles proactifs et réactifs bénéficie, également, aux protocoles hybrides (cf. § II.5.3) puisqu'ils combinent ces deux types de protocoles pour déterminer une route (réactifs pour les réseaux moins dense et proactifs dans le cas contraire), c'est le cas de ZRP.

Les protocoles de routage peuvent utiliser une hiérarchisation du réseau (cf. § II.5.4) pour réduire le nombre d'informations de routage. Cette hiérarchisation, utilisée dans CBRP et HSR, réduit le nombre de nœuds concernés par ces informations et réduisent la bande passante consommée en réduisant la taille des tables de routage transmises par les nœuds.

Les protocoles, utilisant des informations de localisation (cf. § II.5.5), sont également un moyen de réduire la bande passante consommée par la découverte des routes. Bien souvent, les protocoles de localisation sont des protocoles réactifs et opèrent donc par flux. Ces protocoles diminuent le nombre d'informations de routage en connaissant la position de la destination. Ils peuvent ainsi réduire le nombre de nœuds du réseau transmettant les paquets de routage comme par exemple les protocoles LAR et DREAM. Une réduction de l'espace de recherche peut entraîner l'échec de détection d'une route. Ainsi, il est toujours nécessaire de combiner cette méthode de recherche avec un autre protocole de routage [25].

Les protocoles OLSR, AODV et DSR sont les seuls protocoles standardisés dans les réseaux MANETs. De fait, ces protocoles sont implémentés en priorité dans les équipements supportant les réseaux MANETs.

L'objectif de cette thèse est la proposition d'une architecture pour le support de la qualité de service pour les MANETs. Malgré que le terme QdS soit très largement employé dans la littérature, aucun consensus sur une définition bien claire n'est fait pas toujours le point. C'est pour cette raison, le prochain chapitre tente d'apporter des éclaircissements sur ce point en présentant un état de l'art très détaillé sur les différentes solutions réalisées ayant trait aux réseaux ad hoc que ce soit pour la couche réseau ou MAC.

Chapitre 3 : Qualité de Service dans les réseaux

III.1. Introduction

Le terme ***Qualité de Service*** (QdS) recouvre différentes significations selon les communautés. Il est donc difficile d'en donner une définition rigoureuse et satisfaisante. Elle est composée de deux mots qui sont eux mêmes mal définis et très ambigus ! Le terme ***qualité*** désigne l'ensemble des caractéristiques d'un produit ou d'un service qui confèrent à celui-ci la possibilité de satisfaire aux exigences énoncées. Il est utilisé pour décrire un processus de livraison de données d'une manière fiable ou meilleure que la normale (contexte réseau de communication). La notion de ***service*** quand à elle peut recouvrir des niveaux d'abstraction plus ou moins élevés [68].

La qualité de service est un concept visant à fournir à l'utilisateur un service vérifiant certains critères qualitatifs. Cet aspect qualitatif est représenté par des paramètres quantitatifs exprimés par l'utilisateur. Ces critères sont spécifiques à l'application de l'utilisateur.

Le terme qualité de service (QdS) est largement utilisé aujourd'hui, non seulement dans le monde des télécommunications où il trouve son origine, mais de plus en plus dans le contexte des services large bande, hertziens et multimédias basés IP [69].

Dans la première partie de ce chapitre, on va définir les différents concepts liés à la qualité de service, puis on a présenté les différents modèles et travaux concernant la QdS, et dans la dernière partie, on a étudié un certain nombre de travaux récents traitant la QdS pour mieux situer notre contribution.

III.2. Définition de la qualité de service

La qualité de service a été définie selon une recommandation du CCITT (Comité Consultatif International Téléphonique et Télégraphique) comme "l'effet général de la performance d'un service qui détermine le degré de satisfaction d'un utilisateur du service" [70]. Il s'agit d'une définition subjective qui donne une perception de la qualité de service du point de vue utilisateur.

Cette définition est la plus largement acceptée puisqu'elle ne référence aucune métrique comme la bande passante, le délai, etc…, ou un mécanisme comme le contrôle d'admission, le protocole de signalisation, etc.

D'un point de vue technique, la qualité de service peut être définie comme la capacité de garantir un certain niveau d'assurance, de telle sorte que la fluidité des trafics et/ou des services requis soit au mieux satisfaite pour une application, un hôte ou même un routeur. Cette qualité de service peut également correspondre dans un réseau, à un ensemble de mécanismes permettant de partager équitablement selon les besoins requis des applications, les différentes ressources offertes par le réseau, de manière à donner, autant que possible, à chaque application (à chaque utilisateur) la qualité dont elle a besoin [15].

La RFC 2386 [71] caractérise la QdS 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, variance de délai (gigue), bande passante, pertes de paquets et de disponibilité.

La qualité de service est basée en général sur un certain nombre de paramètres, de natures différentes et qui ont pour but de préciser les besoins des utilisateurs envers les fournisseurs de service. Par exemple, la téléphonie sur IP a pour but de pouvoir converser en temps réel sans entre-coupures engendrées par des délais supplémentaires, ce qu'on peut qualifier de **facteur du délai** ; le téléchargement d'une application volumineuse nécessite une assez large bande passante pour récupérer les fichiers de l'application le plus vite possible. Dans ce cas nous parlons de **facteur du débit** ; la plupart des applications exigent des garanties en termes de réception de l'intégralité des paquets. Elles sont sensibles au **facteur de pertes de paquets**.

La plupart des algorithmes de qualité de service du monde filaire reposent sur la connaissance d'informations précises sur l'état du réseau, considèrent que les pertes sont faibles, que la bande passante disponible est large ou encore que la topologie du réseau est stable. Plusieurs travaux ont été réalisés mais il est encore trop tôt pour que l'un d'entre eux s'impose comme étant la solution de qualité de service pour les réseaux MANETs.

Dans le cas des réseaux ad hoc, le support de la qualité de service doit prendre en compte un certain nombre de contraintes (mobilité des nœuds, énergie limitée, lien imprévisible, médium radio partagé, sécurité, et maintenance des routes). Ceci a conduit au développement de solutions garantissant la qualité de service (réduction de la congestion, les délais et la perte de paquets d'informations). Cependant ces solutions ont visé plusieurs niveaux du réseau, en proposant des fonctions d'accès au canal radio avec QdS, des protocoles de routage avec QdS, une combinaison de couches (solution cross layer), des modèles de QdS, etc [15].

La notion de qualité de service est, comme nous l'avons précédemment explicitée, un aspect multidimensionnel basé sur des critères plus ou moins complexes qu'il faut garantir. Les principaux aspects connus de la qualité de service sont le délai, la gigue, le débit, la bande passante et la disponibilité (souvent exprimée en termes de taux d'erreurs). En générale, la QdS est exprimée par les paramètres ou critères décrits dans le paragraphe suivant :

III.3. Paramètres de la QdS

III.3.1. Paramètres de délai (delay)

- **Le délai** : C'est le temps écoulé entre l'envoi d'un paquet par la source et sa réception par le destinataire. C'est une des caractéristiques principales de la QdS. Ce délai tient compte du délai de propagation le long du chemin et le temps de transmission qui est fonction du débit binaire et de la taille des paquets émis, le temps de traitement, le délai induit par la mise en file d'attente des paquets dans les routeurs et le délai introduit par le buffer de compensation de la gigue pour assurer la synchronisation. La plupart des applications et surtout les applications temps réel sont très sensibles aux valeurs élevées de délais et exigent un délai limité pour un fonctionnement correcte [72][73][74].
- **La Gigue** (jitter): C'est la variation de délai d'acheminement des données de bout en bout (transferts variables dans les nœuds de réseau). La gigue est un critère particulièrement important pour les applications de transmission Audio/vidéo. A cause de la gigue introduite par le réseau, les flux audio/vidéo capturés au même instant peuvent ne pas arriver simultanément chez le récepteur. Au niveau du récepteur, les applications temps-réel doivent bufférriser les données pour enlever la gigue ajoutée par le réseau et retrouver les relations temporelles originales. L'IETF a aussi défini formellement la notion de gigue instantanée comme étant la variation instantanée de délai (Instantaneous Packet Delay Variation : IPDV). C'est la différence de délai de transmission entre deux paquets *k* et *k+1* consécutifs. Cette gigue instantanée reflète l'évolution de l'état de congestion du lien. Si elle est stable, la charge du lien est constante. Si par contre elle augmente, elle indique la dérivation vers un état de congestion [68].
- **La latence**, représente le retard entre l'émission et la réception d'un paquet.

III.3.2.Paramètres de débit (throughput)

- **Le débit** binaire ou par abus de langage, la bande passante, entre deux entités communicantes est le nombre de bits que le réseau est capable d'accepter ou de délivrer par unité de temps. C'est le taux de transfert maximum pouvant être maintenu entre ces deux entités (un émetteur (E) et un récepteur (R)). Ce facteur est influencé non seulement par les capacités physiques des liens, mais aussi par les autres flux partageant ces liens. Le débit utile dépend du niveau auquel on se place dans la hiérarchie protocolaire. Par exemple, la bande passante d'un lien réseau, représente la capacité en bits par seconde que ce lien peut transporter, dans laquelle les données n'incluent pas les bits nécessaires pour les entêtes de niveau 2. Au niveau application, on considère la capacité du lien (throughput) qui correspond au volume effectif de données transmis. La capacité utile du lien (goodput) est égale au nombre total de bits issus de l'application et correctement transmis par unité de temps [68][72][73][74].

III.3.2. Paramètres de fiabilité

- **Le taux de perte de paquet** (Packet Loss Ratio) : correspondant au taux de non délivrance de paquets (le rapport du nombre de paquets non livrés sur le nombre total de paquets transmis). Les pertes dans un réseau sont causées par la congestion, l'instabilité du routage, les défaillances de liens physiques et l'incertitude des liaisons sans fil. La perte de paquets est la conséquence d'une congestion. Elle peut se produire soit par dépassement de capacité des files dans les routeurs qui se trouvent souvent obligés d'éliminer des paquets pour faire face à des situations de congestion ou soit par violation de délai borné entre les bouts (E/R). En effet, certaines applications ne tolèrent pas des valeurs élevées de ce facteur. Cependant, dans les réseaux sans fil, le taux d'erreur n'est pas négligeable [68][72][73][74].

- **Dé-séquencement** : il s'agit d'une modification de l'ordre d'arrivée des paquets.

En effet, le but de le la QdS est d'optimiser les ressources du réseau, de garantir un degré de performances aux applications et d'offrir aux utilisateurs des débits importants et des temps de réponse rapides. Selon le type de service utilisé, la qualité de service pourra résider dans :

- Le débit (ex: la diffusion vidéo)
- Le délai (ex: la téléphonie IP, sensible à la gigue)
- La disponibilité (ex: l'accès à un service partagé)
- Le taux de pertes de paquets

Après avoir définit la QdS, on essaye de faire un bilan non exhaustif sur les travaux de QdS dans les réseaux ad hoc. Vu que la majorité des travaux sont basés sur l'un ou la combinaison des deux premiers modèles du monde filaire (IntServ et DiffServ), le paragraphe suivant leurs sera consacré.

III.4 Modèles Classiques (IntServ & DiffServ)

III.4.1. Modèle IntServ

III.4.1.1 Description

Intserv (Integrated Services) [76] fut le premier modèle de QdS orienté flux pour le réseau Internet. Il repose sur une approche de réservation de ressources pour d'offrir aux applications multimédia et temps-réel des garanties en termes de bande passante et de délai dans les files d'attente des différents nœuds du réseau et un mécanisme de contrôle d'admission pour vérifier si les nœuds disposent des ressources suffisantes pour répondre aux besoins des applications. Dans Intserv chaque application, avant de commencer le transfert de ses données, elle transmet ses conditions (exigences en QdS) à tous les nœuds traversés jusqu'à la destination par un protocole dédié et le transfert ne débutera que si l'ensemble des nœuds sont capables d'honorer ces conditions (Les clients obtiennent donc soit exactement ce qu'ils ont demandé soit leur requête échoue). IntServ aborde la QdS en reprenant le concept du "Best effort" et en lui associant le support du trafic en temps réel. L'utilisation d'IntServ à grande échelle pose de sérieux problèmes [77][78].

Le groupe IETF IntServ a défini plusieurs classes de services. Les trois principales sont : le service garanti [80] (i.e. Guaranteed Service ou GS conçu pour répondre aux besoins des applications temps-réel non adaptatives caractérisées par des garanties strictes de délai et de bande passante), le service à charge contrôlée [79] (i.e. Controlled Load ou CL qui vise à répondre aux besoins des applications temps-réel adaptatives ; il effectue une différenciation entre les trafics à base de priorités) et le service meilleur effort ou best effort (i.e. aucunes garanties de QdS ; cas de la messagerie électronique et le transport de données) [73].

III.4.1.2. Architecture d'IntServ

Dans l'architecture IntServ, avant que la transmission des paquets ne commence, une phase de réservation de ressources est effectuée au niveau de chacun des nœuds du chemin par l'application ayant besoin d'un certain niveau de QdS. La réservation de ressource est effectuée par le protocole RSVP [81]. Elle comporte : la phase d'établissement, de rafraîchissement, de données et de libération.

Les deux composants nécessaires dans le modèle IntServ/RSVP sont :

- ***Le plan de contrôle***, permettant de réserver les ressources nécessaires. Pour le faire, le routeur dispose d'un agent RSVP chargé de l'initialisation et du maintien des réservations, d'un contrôle d'admission vérifiant si les ressources sont suffisantes pour répondre aux besoins de l'application et d'un contrôle de règles déterminant si la demande de réservation est légitime par rapport aux règles fixées par l'administrateur du réseau (i.e. si la requête est issue d'un utilisateur autorisé) ;
- ***Le plan de données***, transmettant les paquets en fonction de l'état des réservations. A leurs arrivées, les paquets appartenant aux flux réservés sont sélectionnés et insérés dans les files d'attente appropriées par le module de classification. L'ordonnanceur alloue les ressources aux flux en fonction des informations contenues dans la table des réservations. Le module de gestion du trafic assure la conformité des flux par rapport à leurs spécifications, et si ce n'est pas le cas, ils seront marqués ou éliminés [82].

Chaque routeur IntServ est constitué des éléments suivants: le classificateur (classifier), l'ordonnanceur (scheduler), le contrôleur d'admission (admission controller) et un démon du protocole RSVP [73].

III.4.1.3. Le protocole RSVP

RSVP (ReSerVation Protocol) fut développé par l'IETF en 1993 [83] puis amélioré pour le rendre plus flexible avec la capacité de gérer des sessions entre un grand nombre d'utilisateurs.

Une session[1] dans RSVP comporte les éléments suivants : adresse de destination, le type de protocole de la couche transport et le numéro de port de la destination [80][84].

RSVP est caractérisé par les éléments suivants :

- l'hétérogénéité du protocole : pour un même flux, on peut recevoir une QdS différente ;
- le protocole est orienté récepteur : c'est le récepteur qui choisi la QdS à fournir au flux;
- protocole dynamique parce que les réservations peuvent être renégociées à tout instant;
- les états du protocole sont de type "soft-state": rafraichissement de l'état des routeurs par réémission des messages ce qui permet à de nouveaux participants de se joindre à une session de façon dynamique [73].

RSVP se base sur l'utilisation de deux types de messages unidirectionnels (Figure 3. 1):

- le message PATH permet d'établir un état dans chaque nœud traversé dans le réseau et trace le chemin de retour qui sera suivi par le second type de message;
- le message RESV émis des récepteurs vers les émetteurs pour concrétiser la réservation [73].

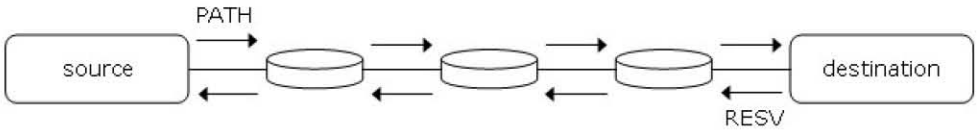


Figure 3. 1: Fonctionnement du protocole RSVP [82]

III.4.1.4 Principe de réservation

Quand la source envoie en best effort un message PATH (Figure 3.1), contenant la spécification du flux. Chaque routeur traversé sur le chemin de la source à la destination, insère ou modifie des informations relatives au chemin suivi, ainsi que les services et ressources disponibles dans le message PATH.

En fonction des informations collectées sur les possibilités réelles du réseau et de la spécification du flux, le destinataire est capable de déterminer la QdS qui pourra être assurée. Pour le faire, il retourne le message RESV sur le chemin de retour contenant le type de service désiré et le filtre (i.e. protocole de transport et le numéro de port), qui caractérisent les paquets pour lesquels la réservation doit être établie.

La demande de réservation de ressources dans chacun des routeurs recevant le message RESV est soumis à un contrôle d'admission et un contrôle des règles d'usage. En cas d'échec de réservation par manque de ressources ou par problème d'authentification, une erreur est retournée à la destination. Un rafraichissement est effectué périodiquement toutes les 30 secondes [82].

III.4.1.5 Limites du modèle

Les principaux problèmes posés par IntServ sont de deux types:

- Le problème d'extensibilité (facteur d'échelle ou déploiement à l'échelle du plein Internet) avec une réduction considérable des performances des routeurs par la gestion d'un grand nombre de flux (coût de signalisation nécessaire au maintien des états par rafraîchissement périodique et la charge additionnelle induite par le mécanisme de réservation) et
- Le problème de définition d'une architecture de bout en bout capable d'interfacer les applications avec les services IntServ.

Pour les MANETs, le modèle IntServ n'est pas adéquat due au problème de passage à l'échelle et de l'incapacité des nœuds mobiles à gérer une grande quantité d'information relative aux flux puisqu'ils disposent de ressources limitées (stockage et traitement).

La charge de contrôle additionnelle induite par les mécanismes de réservation, de signalisation et de contrôle consomme une quantité non négligeable de bande passante sur des liens déjà limités.

1 : (RSVP_session = <dest_address, protocol_transport-layer, dest_port_number>).

III.4.2 Modèle DiffServ

III.4.2.1 Description

Le modèle des services différenciés, ou DiffServ (Differentiated Services) [85] fut développé à la fin de l'année 90 pour répondre aux limites d'IntServ. Il effectue une différenciation de services en divisant le trafic des utilisateurs en un petit nombre de classes (approche basée sur l'agrégation) pour pallier le problème d'extensibilité (facteur d'échelle) de l'architecture IntServ [82].
DiffServ, utilise une technique de marquage des paquets en utilisant le champ TOS (Type Of Service) dans l'en-tête d'un paquet IP pour déterminer à quelle classe de trafic ce paquet spécifique appartient [15].

DiffServ est basé sur les principes fondamentaux suivants :

- Il remplace la notion de flux en cœur de réseau par la notion de *classe* ou les paquets d'une même classe sont traités de la même façon par tous les routeurs le long du chemin entre source et destination.
- Il n'assure pas de contrôle d'admission (aucun protocole de signalisation n'est défini), mais il s'appuie sur la notion de contrat *SLA (*Service Level Agreement) négocié au préalable entre client et fournisseur de service lors du transfert des données. Ce contrat précise les engagements de l'opérateur en ce qui concerne la QdS proposée, les métriques associées et les pénalités applicables en cas de non-respect des engagements [82].

La partie technique SLS (*Service Level Specification)* d'un SLA spécifie (à minima) la quantité maximale de trafic de chaque classe que l'utilisateur a le droit d'injecter, ainsi que le traitement (*conditionnement*) que subiront les paquets entrants.

Une définition de SLA contient les paramètres suivants: Identificateur de la source (@IP, N°port, protocole), Identificateur de destination (@IP, N°port, protocole), débit, taux de perte et dates (début et fin) du flux [78].

III.4.2.2. Les services proposés

Le groupe IETF DiffServ a définit les trois services suivants (niveaux de priorités): le service par défaut ou Best Effort (i.e. correspond à la priorité la plus faible, c'est le service utilisé actuellement dans l'Internet), le service AF (Assured Forwarding), définit dans la RFC 2597 [86] (i.e. offre des garanties de débit et de taux de pertes avec différents niveaux de priorité ou quatre classes sont définies pour ce service) et le service EF (Expedited Forwarding *ou premium service),* définit dans le RFC 2598 [87] (i.e. assure un transfert à fortes contraintes temporelles) [73][77][82].
Nous résumons dans le tableau ci-dessous (Tableau 3. 1) les différents services, les priorités associées, des exemples d'applications courantes avec la classe de services de l'architecture DiffServ [73].

Service	Priorité	Type d'application	Classe
Best Effort	Faible	E-mail	1
Assured Forwarding	Moyenne	Navigation Internet	2
Expedited Forwarding	Forte	Vidéo conférence	3

Tableau 3. 1: Services DiffServ et qualités requises pour les applications courantes

III.4.2.3. Architecture de DiffServ

Pour décharger les routeurs de ne plus sauvegarder les informations d'états (cas d'Intserv), le principe de DiffServ consiste à diviser le réseau en domaines. On distingue ainsi deux types de routeurs : les routeurs de cœur (Core Routers) à l'intérieur d'un domaine, chargés uniquement de l'acheminement des paquets selon le marquage et les routeurs de bord ou d'accès (Border or Edge Routers), chargés de la classification, du marquage et du maintien de l'état des flux[15].

Les routeurs d'accès sont connectés aux clients, tandis les routeurs de bord sont connectés entre eux.

L'architecture DiffServ (Figure 3. 2) classifie les paquets en un nombre limité de classes. Elle suppose qu'un champ d'en-tête du paquet IP appelé DS (*Differentiated Service*) porte l'indice de la Classe de Service DSCP (*Differentiated Service Code Point*). Les routeurs de cœurs se consacrent exclusivement au traitement des paquets en utilisant ce champ et ceux de bord sont chargés de conditionner le trafic entrant en indiquant explicitement sur le paquet, le service qu'il doit subir [15][77].

Pour réaliser le marquage des paquets, le groupe DiffServ propose d'utiliser le champ Type de Service (TOS) d'IPv4 ou l'octet COS (Classe Of Service) d'IPv6 [88].

Chaque ensemble de paquets défini par une classe reçoit un même traitement PHB (Per Hop Behavior) soit EF ou AF codé par le DSCP [77].

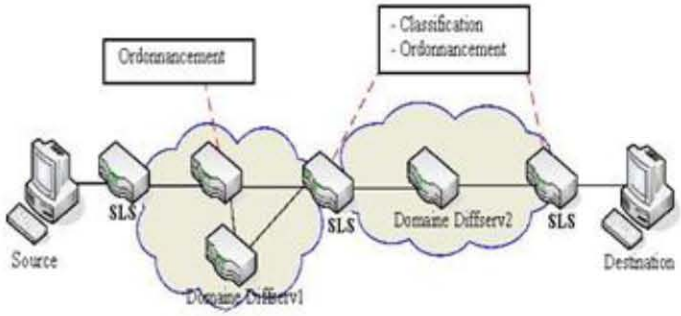


Figure 3. 2: Architecture du modèle DiffServ [15]

III.4.2.4. Limites du modèle:

Les problèmes majeurs laissés ouverts par DiffServ sont [78]:

- La disponibilité des ressources sur le chemin de données ;
- La non garantie d'une différenciation absolue, c'est-à-dire que plus une classe est grande, plus elle sera privilégiée pour le partage des ressources par rapport aux autres classes concurrentes.

Le modèle DiffServ n'est pas utilisé dans les MANETs puisque dans les réseaux ad hoc, les nœuds sont identiques et sont à tout moment soit source ou destination d'une communication ou soit un nœud de relai (routeur) pour une autre. La projection (définition) des routeurs de bord et des routeurs de cœur ne trouve pas sa place et reste très ambiguë pour les réseaux ad hoc [15].

III.4.3 Conclusion

Les deux modèles classiques (IntServ et DiffServ) proposés pour le support de la QdS dans l'internet offrent plusieurs services en plus du service de base ou Best Effort. Pour le modèle IntServ, on trouve le service contrôlé (CL) dédiés aux besoins des applications temps-réel adaptatives et le service garanti (GS) qui vise à répondre aux besoins des applications temps-réel non adaptatives exigeant des garanties strictes de délai et de bande passante. Quant au modèle DiffServ, on trouve le service AF (Assured Forwarding) offrant des garanties de débit et de taux de pertes et le service EF (Expedited Forwarding) pour applications à fortes contraintes temporelles pour un débit et un délai borné.

Pour les MANETs, le modèle IntServ n'est pas adéquat due au problème de passage à l'échelle et le volume de charge de contrôle et de signalisation que les nœuds mobiles ne sont pas en mesure de supporter et qui consomme une quantité non négligeable de bande passante ce qui pèse énormément sur la capacité des nœuds aux ressources déjà limitées.

Le modèle DiffServ classe les nœuds du réseau en nœud d'accès ou de cœur, et comme dans les MANETs, les nœuds sont identiques, ce modèle ne trouve pas sa place et reste très ambiguë pour les réseaux ad hoc.

III.5. QdS dans les MANETs (réseaux ad hoc)

Les recherches sur la QdS dans les réseaux ad hoc sont souvent classées suivant les couches réseau [89]:

- les protocoles d'accès au médium cherchent à ajouter des fonctionnalités aux couches basses du modèle OSI afin de pouvoir offrir des garanties en QdS (couche MAC)
- les protocoles de routage avec QdS recherchent les routes ayant suffisamment de ressources disponibles pour satisfaire une requête (Couche réseau)
- les protocoles de signalisation cherchent à offrir des mécanismes de réservation de ressources indépendants du protocole de routage sous jacent.
- les modèles de QdS définissent des architectures dans lesquelles des garanties peuvent être fournies.
- le codage particulier de l'information multimédia (couche application)

Dans ce qui suit, nous aborderons les modèles de QdS les plus traités dans la littérature.

III.5.1. Modèles de QdS pour les MANETs

III.5.1.1. Définition d'un modèle de QdS

Un modèle de QdS décrit un ensemble de services de bout-en-bout, qui permettent aux clients de sélectionner un nombre de garanties qui gouvernent des propriétés telles que le temps, l'ordonnancement et la fiabilité. Il spécifie l'architecture qui permettra de proposer un service meilleur que celui du modèle best effort traditionnel. Cette architecture doit prendre en considération les contraintes imposées par les réseaux ad hoc : Topologie dynamique et imprévisible, des ressources limitées en termes de capacité de la batterie, de traitement et de stockage et les contraintes de délai, de bande passante limitée et de fiabilité.

Ces travaux sont inspirés de ce qui a été réalisé pour les réseaux filaires pour intégrer les aspects tels que: le partage de charge entre les différents nœuds, l'économie de l'énergie consommée, etc. [15][90].

III.5.1.2. Les principaux modèles utilisés

III.5.1.2.1: FQMM (Flexible QdS Model for MANETs)

FQMM [91] fut le premier modèle de QdS proposé pour les MANETs en 2000; ses concepteurs tentaient d'offrir un mécanisme de QdS suffisamment proche des protocoles filaires afin de l'interfacer dans un avenir proche aux réseaux filaires de type Internet. FQMM repose sur une architecture réseau plate non hiérarchique, constituée d'une cinquantaine de nœuds mobiles, formant un domaine DiffServ. Il combine les propriétés des modèles filaires IntServ et DiffServ, en offrant une méthode d'approvisionnement hybride : par flux de l'IntServ, pour les trafics prioritaires basé sur l'hypothèse que tous les paquets dans le réseau ne demandent pas la priorité maximale, et par classe de DiffServ pour les autres trafics.

Dans le réseau, les nœuds peuvent avoir des rôles différents suivant les trafics existants : nœud d'entrée ou émetteur (ingress node), intermédiaire (core ou interior node) ou de sortie ou récepteur (egresse node). A un moment donné, le nœud peut avoir un seul rôle (i.e. une seule connexion), deux rôles différents (i.e. le cas ou il fait partie dans deux connections) ou trois rôles différents (i.e. le cas ou il fait partie dans trois connections) [15][91].

Les nœuds d'entrée permettent de marquer et classifier les paquets, qui seront ensuite relayés par les nœuds intermédiaires suivant leurs PHB [86], jusqu'à arriver au nœud destinataire. Ce modèle repose essentiellement sur la couche IP, où les fonctionnalités sont séparées en deux grands plans : le plan relayage de données et le plan contrôle de gestion (Figure 3.3) [8][75][90].

Architecture modèle : La Figure 3. 3 présente l'architecture du *modèle FQMM*.

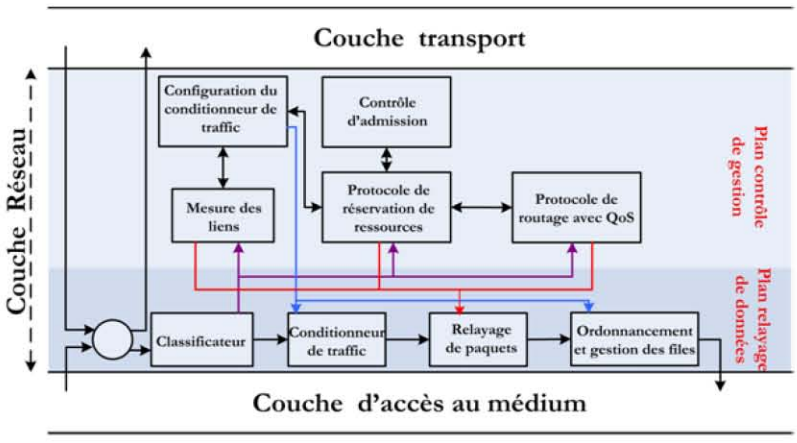


Figure 3. 3: *Le modèle FQMM* [15]

Les différents composants assurent les mêmes fonctions que ceux vues dans IntServ ou DiffServ.

Le module de conditionnement de trafic ou conditionneur est associé au nœud d'entrée source du trafic. Il dirige le trafic en fonction du profil associé à ce trafic après qu'un chemin valide est trouvé. Ce module contient les composants : de marquage, de mesure ou meter (i.e. vérifie si le flux est conforme au profil déterminé par le configurateur), de remise en forme ou shaper, de rejet (i.e. retardant et éliminent respectivement les paquets non conformes (hors profil)) et d'un configurateur de profil de trafic chargé de décider de la politique des autres composants qui changent de configuration en fonction du profil du trafic.

Ce profil est définit dans FQMM comme étant le pourcentage relatif de la capacité effective du lien pour maintenir une différenciation cohérente entre les classes en tenant compte de la mobilité.

FQMM requiert l'utilisation d'un protocole de routage capable d'offrir une certaine QdS avec suffisamment de ressources.

Le partage de la bande passante des liens et l'allocation de la mémoire tampon sont deux aspects importants dans le module de gestion des ressources. Le premier est fait par l'ordonnanceur et le second par une gestion active (AQM) de files d'attente pour contrôler la congestion du réseau [15][91].

Malgré ces avantages, FQMM présente quelques lacunes comme l'absence de tout contrôle explicite du nombre de services par flux offerts pose un problème d'extensibilité (facteur d'échelle), la difficulté de coder le PHB dans le champ DS (taille limitée) s'il contient une granularité par flux, la difficulté de faire un profil dynamiquement négocié pour le trafic, et la résolution de la plupart des problèmes liés au fonctionnement des réseaux ad hoc (volume de signalisation, consommation de l'énergie, bande passante limitée et l'interaction avec la couche MAC) est laissée à la charge du protocole de routage sous-jacent [8][15][75][90]

III.5.1.2.2. SWAN (Service differentiation in Wireless ad hoc Networks)

SWAN [92] est un modèle réseau sans état (i.e. ne garde pas d'état dans les nœuds) basé sur des algorithmes de contrôle distribués dans le but d'assurer une différenciation de services dans les réseaux ad hoc. Il n'utilise aucun message de contrôle pour garantir dynamiquement la bande passante des flux QdS [93]. Il considère le trafic TCP comme un trafic best effort et le trafic UDP comme un trafic temps réel (trafic avec QdS).

Pour accepter un nouveau trafic temps réel, un contrôle d'admission sonde la bande passante minimale disponible sur la route (valide et obtenue par un protocole de routage) et c'est à la source de prendre la décision adéquate en fonction de la bande passante obtenue.

Le modèle SWAN utilise un certain nombre de mécanismes pour la régulation du débit pour le trafic best effort et le contrôle d'admission pour le trafic temps réel. Pour réguler le trafic temps réel au moment de la congestion dans le réseau, le modèle SWAN utilise des mécanismes de contrôle basés sur la réaction.

Architecture de SWAN : Le modèle SWAN est composé de quatre éléments (Figure 3. 4) : *un contrôleur d'admission*, *un contrôleur de débit*, *un classificateur* et *un shaper* (module de lissage). Les deux contrôleurs sont associés aux différents nœuds ; le classificateur et le shaper opèrent tout les deux entre les couches IP et MAC. Le classificateur permet de différencier entre les deux types de trafic (temps réel et best effort) en obligeant le shaper de ne traiter que le trafic best effort mais pas le trafic temps réel. Le but du shaper est de retarder le trafic best effort en conformité avec le débit calculé par le contrôleur de débit pour offrir plus de chances aux trafics temps réel (au niveau paquet) [93].

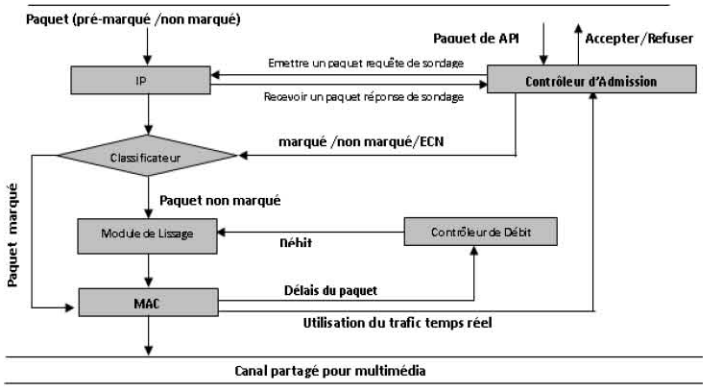


Figure 3. 4 : *Le modèle SWAN.*

Pour garantir dynamiquement la bande passante aux flux QdS sans aucun message de contrôle, SWAN met en place trois mécanismes. Un mécanisme de contrôle d'admission à la source des flux QdS et deux mécanismes de régulation dynamique de trafic, l'un pour les flux QdS et l'autre pour les flux best effort déclenchés seulement si le protocole SWAN estime que le réseau est dans un état congestion [93].

Le contrôleur d'admission décide de l'acceptation ou le rejet d'un flux selon un sondage de la bande passante disponible.

Donc le protocole SWAN permet de fournir une QdS logicielle (soft QdS) en raison du manque de réservation de ressources au niveau des nœuds intermédiaires.

Un flux prioritaire admis n'est pas sûr d'avoir des garanties pour l'entière durée de la communication, et peut à tout moment être violé par d'autres demandes de trafics. Un mécanisme de contrôle de débit des flux best effort n'est pas à lui seul suffisant pour offrir des garanties aux applications temps réel.

III.5.1.2.3. CEQMM (Complete and Efficient Quality of service Model for MANETs)

CEQMM [94] comme son nom l'indique est un modèle complet et efficace de QdS pour les MANETs. Il combine des solutions proposées dans les réseaux filaires et les projettent dans le contexte des réseaux mobiles ad hoc en tenant compte des caractéristiques de ces derniers.

L'idée de base dans ce modèle est qu'il utilise à la fois une gestion par flux d'IntServ et une différenciation de services de DiffServ. En d'autres termes, ce modèle propose un approvisionnement par flux pour les trafics de plus haute priorité et par classe aux autres trafics (autres priorités).

Pour atteindre un tel objectif et à veiller à ce que certains paquets reçoivent une transmission à une priorité plus élevée que d'autres, les composants : classificateur de priorité qui *différencie* les trafics reçus entre trafic de contrôle, trafic avec QdS et trafic best effort puis les *dirige* vers les files d'attente correspondantes selon le niveau de priorité par une gestion de la file active, et l'ordonnanceur de paquets sont intégrés dans l'architecture CEQMM (Figure 3. 5). Les algorithmes de gestion de la file *contrôlent* la taille des files en éliminant des paquets si cela est nécessaire. L'algorithme d'ordonnancement détermine quel est le prochain paquet à envoyer sur le lien.

CEQMM définit plus d'une classe pour le trafic QdS avec différents niveaux de priorité et une seule classe pour le trafic best effort avec la priorité la plus basse.

Le trafic de contrôle est mis dans une file séparée avec la plus haute priorité que le trafic de données pour disposer d'informations plus récentes sur la topologie du réseau et les conditions de QdS.

CEQMM utilise le protocole de routage QOLSR [95] pour supporter plusieurs critères relatifs aux paramètres de routage et de réagir rapidement quand des changements de topologie et/ou de conditions de QdS sont détectées.

Le contrôle des flux QdS avec la plus haute priorité est délégué au nœud source, tandis que les autres classes de QdS sont à la charge des nœuds intermédiaires (prochain saut).

Après le choix d'un chemin pour un flux QdS, le nœud source n'est en mesure de garantir les exigences de QdS de bout en bout pour les raisons de congestion et de mobilité.

Pour tous chemins satisfaisants les contraintes de flux QdS, un contrôle d'admission est nécessaire pour vérifier leurs validités et procéder à la réservation de ressources sur les nœuds intermédiaires.

Pour éviter un conflit de ressources pour deux flux QdS lancés simultanément, chaque nœud commence par une réservation *soft* et plus tard *hard* de la bande passante sur les liens pour seulement les flux QdS de plus haute priorité.

Pour empêcher que le réseau entre dans un état de congestion (déclenchée par mobilité et par dégradation des performances), CEQMM implémente des mécanismes d'évitement de congestion comme la gestion active de file d'attente et l'utilisation de l'algorithme de backoff.

CEQMM fonctionne en best effort au niveau MAC et peut inclure des nœuds hétérogènes avec des capacités radio différentes.

Architecture de CEQMM (Figure 3.5) Elle comporte cinq composantes principales : un protocole de routage avec QdS (QOLSR), un estimateur de métriques, un contrôleur d'admission et de réservation, un classificateur de trafic et un contrôleur de congestion.

Les modules QOLSR et l'estimateur de métriques coopèrent étroitement pour mesurer les contraintes de QdS localement et les propager ensuite dans le réseau. Pour chaque flux QdS, le nœud source et les nœuds intermédiaires dans le chemin calculé, effectuent un contrôle d'admission et une réservation en se basant sur la connaissance de la bande passante dans le réseau. Dans les MANETs, le réseau peut toujours développer des congestions suite à une mobilité ou changement de connectivité. Pour cette raison, un contrôle de congestion est extrêmement important pour surveiller en permanence l'utilisation de la bande passante et la perte des paquets de données [90].

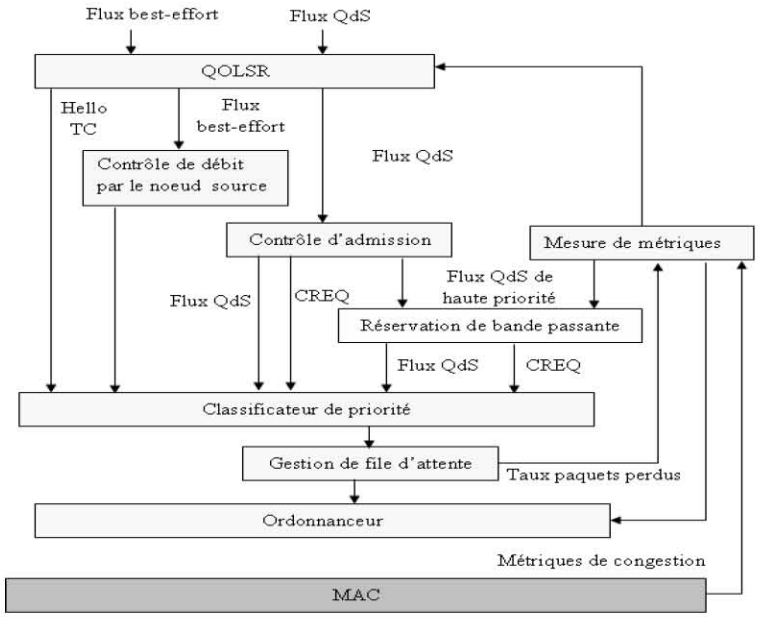


Figure 3. 5: architecture de CEQMM

NB : Les besoins en QdS (Bande Passante, délai, etc) sont des champs dans le message CREQ (Check REQuest message). Quand le nœud destination reçoit un message CREQ, il retourne un message CREP (Check REPly message) vers la source du flux QdS.

III.5.1.2.4. QPART (QdS Protocol for ad hoc Real-time Traffic)

Ce modèle se situe dans le même état d'esprit que SWAN ceci par rapport à la différentiation de services entre différents types de trafic et les garanties pour certains trafics sans garder d'états dans les nœuds. Le protocole QPART [96] se base sur une estimation passive de la bande passante et un mécanisme de régulation dynamique du débit des flux best effort. Il classifie les trafics avec QdS en trafics sensibles au délai, sensibles à la bande passante et bien sûr il gère le trafic de base best effort.

QPART n'effectue ni un routage avec contraintes ni un contrôle d'admission mais il agit seulement sur la résolution de conflits en cas de congestion en arrêtant certains flux selon leur priorité (l'âge du flux). Plus le flux est vieux, plus il est prioritaire grâce à une mise à jour périodique jusqu'à atteindre une priorité maximale. Chaque priorité correspond à un seuil d'admission qui lui-même correspond à un temps libre moyen entre deux périodes d'occupation du médium. Chaque fois qu'un nœud détecte que le temps libre moyen est inférieur à ce seuil, le flux est arrêté pendant un certain temps. Si au bout de ce temps, le temps libre moyen perçu pas le nœud est toujours inférieur à ce seuil alors le flux est définitivement rejeté, sinon il est redémarré. Les choix de ces paramètres et de ces seuils sont les points clés de ce type d'approche [15].

L'Architecture de QPART regroupe deux éléments : l'ordonnanceur de prise en compte de QdS (QdS-aware Scheduler) et le gestionnaire de QdS (QdS Manager) (Figure 3. 6). Les deux composants couvrent les deux couches (réseau et MAC). Les deux composants fonctionnent indépendamment et ne nécessitent aucun échange de messages entre nœuds voisins ou de connaissance sur la bande passante du canal [96].

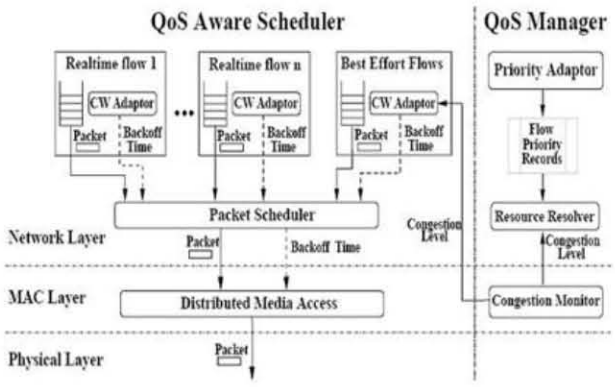


Figure 3. 6: Le modèle QPART [96]

Ordonnanceur de QdS : son rôle est de garantir aux flux temps réel admis, la QdS dont ils ont besoins et contrôle le débit des flux best effort pour allouer plus de bande passante pour les flux temps réel. Il se compose d'un protocole sous-jacent MAC et un ordonnanceur de paquets dans la couche réseau. Sa conception est basée sur la norme IEEE 802.11.

Le gestionnaire de QdS : effectue un contrôle d'admission et assure la résolution des conflits sur la base des priorités des flux temps réel et le niveau de congestion du canal. Lorsque le réseau est congestionné, il sélectionne le flux temps réel de faible priorité et le rejette. Les priorités des flux temps réel, sont attribuées de façon dynamique par le module adaptateur de priorité *(Priority Adaptor)*, elles sont mises à jour dans le module tampon des priorités de flux *(Flow Priority Record)*. Le niveau de congestion du canal est retourné par le module moniteur de congestion situé dans la couche MAC au module résolveur de ressources *(Resource Resolver)*, qui est responsable de la sélection des flux qui seront rejetés en fonction du niveau de congestion du canal et les informations de priorité des flux dans le module tampon des priorités de flux [96].

Limites de QPART :

QPART souffre des mêmes limitations que SWAN. Primo, cette approche suppose que l'ensemble des liens ont des délais similaires, chose qui n'est pas vraie dans la réalité. De plus, le traitement entrepris sur le délai ne prend aucunement en compte l'impact qu'il peut avoir sur les délais des autres flux existants dans le réseau [15]. Secundo, QPART se base sur un mécanisme d'estimation de la bande passante utilisé dans les réseaux filaires et qui est inadapté dans un contexte ad hoc. Tercio, le mécanisme de régulation dynamique du trafic best effort basé sur la variation de la fenêtre de contention ne peut garantir avec précision la bande passante des flux QdS, ni garantir une meilleure utilisation du médium pour les flux best effort [93][96].

NB : il existe d'autres modèles largement cités dans la littérature comme le modèle iMAQ (*an Integrated Mobile ad hoc QdS framework*) [97] avec son architecture cross-layer pour supporter la transmission de données multimédia dans les MANETs et le modèle 2LOoS (two Layered Quality of Service) [98] qui émule les modèles Intserv et Diffserv en utilisant une méthode de codage pour garantir les ressources réseaux.

III.5.2. Qualité de service dans la couche MAC 802.11 et 802.11e

III.5.2.1. Offre de QdS au niveau de la couche MAC

Récemment, de nombreux modèles ou schémas associés au niveau MAC ont été proposés pour les réseaux sans fil, visant à fournir une garantie de QdS pour supporter le trafic temps réel.

Cependant, ces modèles reposent en général sur un contrôle centralisé valable que pour les réseaux sans fil à un saut. Pour les réseaux sans fil multi-sauts, le besoin en un système entièrement distribué pour résoudre d'abord les problèmes des stations cachées et exposées est nécessaire. Le modèle MACA [99] (multi sauts et accès avec évitement de collisions) est proposé à base du dialogue RTS/CTS, mais ceci n'élimine pas complètement le problème des stations cachées.

MACAW (MACA Wireless) [100] a été proposé comme une extension de la MACA pour fournir une récupération plus rapide des collisions des stations cachées. La norme IEEE 802.11 spécifie les caractéristiques d'évitement de collisions de la MACA et MACAW par la fonction de contrôle distribué (DCF) à base de la CSMA/CA, ce qui résout complètement le problème des stations cachées. Toutefois, il ne fournit pas une prise en charge du trafic temps réel. Dans cette section, nous décrivons les issues de QdS proposés pour la couche MAC dans les MANETs.

III.5.2.2. Travaux de QdS au niveau de la couche MAC 802.11

On distingue deux volets, le premier concerne les travaux avant le Draft 802.11e [101] connu sous le nom de différentiation de services et le second c'est ce draft même (802.11e) et les travaux associés à ce dernier.

III.5.2.2.1 Différenciation de services pour 802.11 (avant le draft 802.11e)

Afin de concevoir des mécanismes de différenciation de services efficaces, un mécanisme de priorité entre les trames est utilisé. Pour ce faire, il faut adapter certains paramètres de la fonction DCF du protocole selon la priorité des paquets, c'est le cas de certains paquets de signalisation peuvent être plus privilégies en leur accordant un temps SIFS (Short Inter Frame Spacing) plus court que le DIFS [15].
Dans le but d'améliorer la méthode d'accès DCF du protocole 802.11, plusieurs schémas différents de différentiation de services ont été proposés [102]:

- *Variation du facteur d'incrémentation du temps de Backoff* : on associe différents facteurs d'incrémentation du temps de Backoff pour différentes priorités. Après une collision, la taille de la CW est multipliée par P, au lieu de 2 (remplacement du terme 2^{2+i} par P_j^{2+i}). Par la définition de plusieurs valeurs pour le *Pj,* il est possible alors de différencier les flux selon leurs priorités ; plus la valeur de P, est grande, plus le temps d'attente est grand avant la prochaine tentative de transmission, c'est-à-dire attribuer des valeurs de (*CW*) supérieures pour les stations les moins prioritaires et inversement, ce qui permet ainsi de donner plus de chance à une station prioritaire d'accéder au canal [103].
- *Variation de DIFS* : En associant différents DIFS à différents flux, il est possible d'établir une stricte différentiation entre flux pour l'accès au medium (chacun à sa propre valeur de DIFS qui définit son niveau de priorité pour l'accès au médium). En effet, plus la valeur de DIFS est petite, plus le flux à des chances d'accéder au canal, et vice versa [104].
- *Différentes tailles minimales de la CW*: en attribuant de courtes CW aux flux de haute priorité, cela garantit que ces flux ont plus de chance de pouvoir accéder au canal que ceux de moindre priorité [17].
- *Distributed Fair Scheduling (DFS)* (ou Fonction d'Augmentation du Backoff) [105]: Dans cette technique, la valeur du Backoff générée avant un envoi, est proportionnelle à la longueur de la trame de données et inversement proportionnelle au poids du flux. Ainsi, les stations de faible priorité (petit poids) ont moins de chance d'accéder au canal, sachant que ces dernières génèrent des temps de Backoff plus grand que ceux générés par les stations avec des grands poids [15][17].
- *Longueur maximale de paquets* : Pour augmenter la fiabilité de transmission et introduire une différenciation de service dans IEEE 802.11 est de limiter la longueur maximale de la trame (i.e. soit supprimer les paquets qui dépassent cette longueur ou de les fragmenter). Donc, chaque station a une priorité qui lui permet d'envoyer des trames ayant une taille maximale différente ce qui permet d'obtenir des débits par priorité proportionnels à la taille des paquets utilisés [17][106].
- *DCF étendue* : consiste à utiliser la CW comme un moyen d'attribution de priorités à certaines stations que d'autres. Attribuer une courte CW pour les stations qui devraient avoir une priorité plus élevée assure que dans la plupart (mais pas tous) des cas, les stations de plus haute priorité seront capables de transmettre d'avance sur ceux de faible priorité [106].
- *BlackBurst (BB)*: L'objectif principal de BB [107] est de minimiser le délai pour le trafic temps réel. BB peut être implémenté au dessus de la norme 802.11 sans aucun changement pour les procédures d'accès pour les trafics de données, et avec des changements mineurs pour les trafics temps réel [102][106].
- *AssuRed MAC Extension (ARME)* [108] propose deux classes de service : la classe associée au trafic temps réel ou Assured Rate Service (ARS) et la classe destinée au trafic best effort. ARME propose à la première classe, un débit assuré par application du mécanisme Token Bucket et le mécanisme de la procédure DCF pour les flux best effort [17].

NB : Dans l'ensemble des travaux proposés, aucun algorithme n'a été spécifié pour choisir les valeurs à attribuer à chaque niveau de priorité.

III.5.2.2.2. MACA / PR

Le protocole MACA/PR (Multiple Access Collision Avoidance with Piggyback Reservation) [109] propose de différencier la politique d'accès au médium selon la nature des flux par utilisation de la CSMA non persistante. Il propose une garantie de débit pour les trafics temps réel après une unique demande d'autorisation à transmettre par échange de RTS-CTS (pas de retransmission en cas de collision).Les flux non privilégiés bénéficient d'un service best effort et sont traités de façon standard.

Un protocole de réservation (Piggyback) est utilisé pour initialiser et gérer les réservations de bande passante

En effet, chaque nœud possède une table des réservations (RT) où il inscrit une trace des transmissions. Dès qu'il reçoit un paquet ou un acquittement temps réel, il inscrit dans sa table RT le prochain temps de transmission se trouvant dans l'entête du paquet. Cette table est utilisée pour éviter les conflits dans le cas où il y aurait plusieurs réservations. Les nœuds s'échangent périodiquement leurs table RT [15][104].

III.5.2.2.3 Limitation de QdS au niveau de la couche MAC 802.11

Malgré ce nombre important de solutions proposées au niveau de la couche MAC pour fournir une QdS pour les réseaux MANETs, chacune d'elles apporte une solution spécifique pour un problème bien particulier. La

majorité de ces travaux sont basés sur la méthode d'accès DCF du protocole IEEE 802.11, proposant différents schémas de différentiation de services (Variation de DIFS, BlackBurst, etc…) et aucunes n'est en mesure de fournir un service suffisant pour supporter un trafic avec des exigences de QdS.

III.5.2.3. Architecture du standard 802.11e

La multitude de travaux réalisés sur la technologie IEEE 802.11 l'ont rendue plus compétitive (QdS, sécurité, haut débit). Pour répondre aux challenges de garantie de QdS pour les applications temps réel, le groupe de travail 802.11e à définit deux nouveaux mécanismes d'accès au médium: Enhanced distributed Channel Access (EDCA) et Hybrid coordination function Controled Channel Access (HCCA).

Concernant la définition des paramètres de différenciation de service, le groupe de travail a pris en charge seulement la différenciation inter-classe (i.e. différencier les flux entre classes (AC : Acces Categorie) suivant les exigences de QdS) en ignorant complètement la différenciation intra-classe (i.e. différencier les flux de la même classe suivant les exigences de QdS). De plus, ce draft propose l'utilisation d'un contrôleur d'admission (Admission Control) sans spécifier un algorithme pour sa mise en œuvre.

Dans la suite, nous nous intéresserons plus particulièrement à la gestion de la QdS et ses contraintes dans les réseaux IEEE 802.11. La première partie sera consacrée à EDCA et HCCA et la seconde partie décrira les améliorations portées sur de gestion de la QdS dans la fonction EDCA comme la différenciation intra-classe basée sur le débit de l'application, la stricte différenciation inter-classe et le contrôleur d'admission responsable de l'acceptation ou le rejet de nouveau flux [17].

III.5.2.3.1. EDCA

Le mécanisme EDCA améliore la DCF originale en introduisant un système de différenciation de service au protocole CSMA/CA basé sur la priorité pour le support de la QdS. Elle définit quatre Access Catégories (AC) au niveau de chaque QdS station (QSTA). Chaque AC correspond à un niveau de priorité et pouvant être utilisée par un type de trafic (une variante améliorée du protocole DCF). Pour faciliter le choix des ACs à utiliser avec le type du trafic, le groupe 802.11e a proposé une table de correspondance entre les spécifications IEEE 802.1D (Tableau 3. 2) et les AC de IEEE 802.11e [17][110].

Priorité	Priorité des utilisateurs dans 802.1D	Access Category (AC)	Désignation (Informative)
Faible	1	AC [0]	Background
	2	AC [0]	Background
	0	AC [1]	Best Effort
	3	AC [1]	Vidéo
	4	AC [2]	Vidéo
	5	AC [2]	Vidéo
Elevée	6	AC [3]	Voix
	7	AC [3]	Voix

Tableau 3. 2: table de correspondance entre type d'application et les AC

Chaque AC utilise un ensemble de paramètres (CWmin[AC], CWmax[AC] et AIFS[AC]) pour l'accès au canal où la valeur de AIFS[AC] est égale à SIFS + AIFSN[AC] * SlotTime où AIFSN[AC] >= 2.

III.5.2.3.2. HCF ou HCCA : une fonction d'accès au médium avec QdS

Bien que EDCA améliore considérablement le mécanisme DCF, il reste néanmoins incapable de garantir une stricte protection des AC prioritaires, particulièrement si le réseau est surchargé. Pour pallier à cette limitation, le groupe 802.11e a définit un mécanisme optionnel basé sur l'interrogation HCF (Hybrid Coordination Function) désigné parfois HCCA (Hybrid CF Channel Access).

Hybrid CF propose une gestion déterministe pour l'accès au médium en rendant la PCF initiale de la 802.11 plus souple et améliorer certaines propriétés de QdS.

Elle introduit des modifications à DCF et PCF ainsi qu'un certain nombre de mécanismes et de types de trames permettant la mise en place de transferts avec QdS pendant la CP (période avec contention), durant laquelle c'est le mode EDCF qui est utilisé, et pendant la CFP (période sans contention) durant laquelle c'est le mode HCF qui gère l'accès au canal (Figure 3. 7) [19].

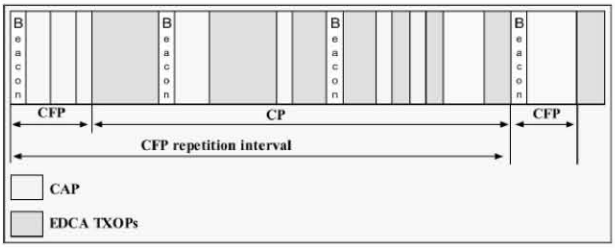


Figure 3. 7: Les périodes CAP/CFP/CP [17]

Le mécanisme HCF repose sur un système de vote afin de répondre aux contraintes de débit, de priorité, de délai, et de gigue.

Après la prise de contrôle du canal, le HC interroge les QSTA inscrites dans sa Polling list (liste de station à interroger). L'inscription dans cette liste se fait par l'envoie une réservation de QdS dans un paquet de contrôle ou management frame pour chaque flux.

III.5.2.4. Les mécanismes supplémentaires de QdS

Le standard (802.11e année 2007) présente différents mécanismes, complémentaires à HCF capable d'offrir une QdS pour cette norme. L'essentiel de ces mécanismes (contrôle d'admission et autres) fut introduit par la modification IEEE 802.11e (802.11e année 2005).

III.5.2.4.1. Le contrôle d'admission

Son rôle est la gestion et la régulation de la bande passante disponible. Il concerne l'accès par HCF avec ou sans contention (par EDCA ou par HCCA). Une QSTA souhaitant avoir des garanties de QdS (sur les délais, sur les débits où sur le taux de pertes) devra passer par le contrôle d'admission. Les algorithmes de contrôle d'admission ne sont pas définis par le standard mais leurs choix sont laissés à l'utilisateur. Cependant, le standard définit un cadre et un certain nombre de règles que ces algorithmes devront respecter [19].

III.5.2.4.2. Autres paramètres de QdS défini par le draft 802.11e

En plus à EDCA et HCCA, le draft 802.11e a introduit d'autres paramètres pour la gestion de la QdS comme:

1) **Block ACK *(blocs d'acquittements ou Block acknoledgement)* :** Cette procédure optionnelle permet d'améliorer l'utilisation du médium. En effet, elle permet à une station d'envoyer plusieurs paquets (un bloc) sans qu'ils soient acquittés individuellement. Ce bloc pourra être acquitté à la fin de l'envoi du dernier paquet du bloc ou dans un TXOP ultérieur (un seul ACK pour un ensemble de paquets). Le block d'acquittements permet de diminuer la charge introduite par le renvoi d'un ACK pour chaque paquet et l'utilisation du réseau s'en trouve ainsi améliorée [17][19].
2) ***Protocole à lien direct (Direct Link Protocol : DLP)*** : Si DLP est activé au sein d'une BSS, les stations peuvent communiquer directement sans avoir à passer nécessairement par l'AP (Acces Point). Ce mécanisme permet une meilleure utilisation du canal d'où une amélioration de la bande passante [17][19]
3) ***Sans acquittement (No ACK) :*** Pour certaine type d'application, la norme 802.11e autorise à ne pas utiliser les paquets d'acquittements (ACK). Cette possibilité est très utile pour les applications soumises à des contraintes temporelles très strictes et à de faibles contraintes de fiabilité [17][19].

III.5.2.4.3. Conclusion

Pour répondre aux challenges de garantie de QdS pour les applications temps réel, le groupe de travail 802.11e définit de nouveaux mécanismes d'accès au médium en l'occurrence : EDCA et HCCA.

EDCA est une amélioration de la procédure DCF pour fournir la QdS de niveau priorité en définissant plusieurs classes de trafic ou Access Category (AC) par contre HCCA ou plutôt HCF améliore la PCF et introduit la notion d'opportunité de transmission (TXOP) en introduisant un système de vote qui convient aux contraintes de débit, de priorité, de délai et de gigue.

D'autres mécanismes complémentaires à HCF introduit dans la norme 802.11e permettent d'offrir une QdS dont l'élément le plus important est essentiellement le contrôle d'admission ; son rôle est la gestion et la régulation de la bande passante disponible. Parallèlement à ceci plusieurs travaux se basant sur la modification de certains paramètres pour la gestion de la QdS ont été réalisés (l'acquittement par blocs, l'utilisation d'un protocole à lien direct, etc).

III.5.3. Système de signalisation pour les MANETs

III.5.3.1. Introduction

Le but des protocoles de signalisation est de fournir un moyen pour propager les informations de contrôle à travers un réseau. Les informations transmises peuvent être de différentes natures (topologiques, requêtes de recherche de routes, des rapports sur l'état du réseau et la disponibilité des ressources). Concevoir un protocole de signalisation consiste à définir les données à échanger et sous quelle manière afin de réaliser une tâche particulière.

La signalisation pour la QdS sert à réserver et libérer les ressources dans le réseau. Pour parvenir à une signalisation efficace, il est primordial, que le transfert des informations entre routeurs soit fiable [90].

La signalisation peut être *in-band* (i.e. les informations de contrôle de flux sont véhiculées dans les paquets de données) simple à mettre en œuvre, où *out-of-band* (i.e. si des paquets de contrôle spécifiques sont utilisés séparément) ce qui engendre un surcoût et consomme de la bande passante.

Le maintien des réservations peut être *"soft-state"* (i.e. les ressources réservées sont libérées si elles ne sont pas utilisées pendant un certain laps de temps) plus adaptée aux MANETs, où *"hard-state"* (i.e. les ressources ne sont libérées que lorsque cela est explicitement demandé) efficace et plus simple.

III.5.3.2 Protocole de signalisation ou modèle INSIGNIA

INSIGNIA [111] est un protocole de signalisation *in-band* spécialement conçu pour les réseaux ad hoc en 1998 pour supporter des services temps réel adaptatifs. Il établit une réservation de la bande passante orientée flux.

L'information de signalisation (messages de contrôle) est encapsulée dans les options des paquets IP (appelé option INSIGNIA) (de type *in-band)*, ce qui permet de réduire l'overhead généré par les messages de contrôle, et ce afin d'éviter de surcharger le réseau contrairement à une signalisation *out-band* explicite. De plus INSIGNIA utilise un système de réservation *soft-state* qui s'avère un bon choix dans des environnements mobiles. Il utilise des algorithmes de réservation, restauration et adaptation dédiés aux MANETs capables de répondre aux changements de topologie du réseau et aux dégradations des liens [8].

Le modèle INSIGNIA offre des garanties sur la base d'une granularité par flux aux applications adaptatives capables de modifier leur comportement en fonction de la quantité de bande passante qui leur est allouée. Ainsi, chaque application spécifie deux types de services pour la QdS [15]:

- Le niveau dégradé ou Best Effort: qui spécifie la bande passante minimale nécessaire au trafic.
- Le niveau amélioré ou Temps Réel: qui permet de spécifier le débit optimal à atteindre lorsque les ressources sont disponibles.

Lorsque le débit d'un flux ne peut plus être assuré, la destination doit avertir la source afin qu'elle prenne les mesures adéquates (i.e. diffère ses transmissions), mais si le flux transite à un débit réduit, la disponibilité de nouvelles ressources est signalée à la destination qui, encore une fois, avertit la source explicitement [15]. Dans certaines situations, des nœuds peuvent être des goulots d'étranglement pour diverses raisons et par conséquent, tous les flux qu'ils vont transmettre vont être dégradés d'un type temps réel à un type best effort.

Notons que INSIGNIA est seulement un protocole de signalisation qu'il faut l'associer à un protocole de routage (DSR ou AODV) qui se chargera de la détection des changements de topologie, de la mise à jour les tables de routage, et à un module de contrôle d'admission pour l'allocation des ressources disponibles [90].

Le modèle de QdS INSIGNIA (Figure 3. 8) doit être transparent aux protocoles de contrôle d'accès aux médias et qu'il est en mesure de fonctionner sur de multiples technologies des couches liaison et IP [111].

La demande de réservation est effectuée lors de l'envoi du premier paquet de données, et est rafraîchie par le passage des paquets de données. Le destinataire informe périodiquement la source de l'état de la route en envoyant des rapports de QdS (QdS Reporting) permettant à la source à réguler son débit d'émission.

Les différents modules d'INSIGNIA sont [8][111] : Signalisation in-band (INSIGNIA), le module transfert de paquets (packet forwarding), le protocole de routage (routing protocol), le contrôle d'admission, le module d'ordonnancement de paquets (packet scheduling), et le module MAC.

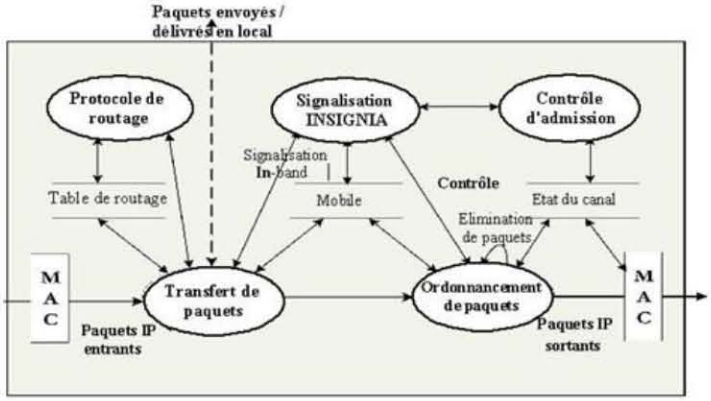


Figure 3. 8: Architecture INSIGNA [15].

Opérations du Protocole :

Les principales fonctions du module de signalisation sont : la réservation rapide (Fast reservation), Rapports QdS (QDS reporting), la gestion de réservation de ressource soft-state (soft-state resource management), restauration (restoration) et l'adaptation de flux (flow adaptation)).

Lors de son transit par les nœuds intermédiaires (cas d'une réservation rapide), le paquet de réservation fait appel aux modules de contrôle d'admission, d'allocation de ressources et établi un état de base dans chaque nœud du lien entre source-destination. Lorsque ce processus s'achève, la destination informe la source via un rapport de QdS (contenant des statistiques sur la latence, le taux de perte et le débit que la source utilisera pour réguler son débit d'émission).

En cas de changement de topologie, une nouvelle réservation est rétablie par le mécanisme de restauration. On distingue trois types de restauration: Intermédiaire (i.e. se produit quand le flux reprend selon sa réservation initiale), dégradée (i.e. se produit quand le flux se dégrade pour une période « T » avant qu'il repend selon sa réservation initiale) et la dégradation permanente (i.e. se produit quand le flux ne reprend jamais selon sa réservation initiale) [15][111].

Finalement INSIGNIA est un protocole de signalisation efficace et bien adapté aux MANETs parce qu'il allie les avantages de la signalisation *in-band* et ceux de la réservation *soft-state* ; néanmoins, il possède quelques lacunes [8][15][90]:

- ✓ D'abord le problème de passage à l'échelle (causé par la capacité des nœuds),
- ✓ Ensuite, la gestion de la bande passante n'est pas optimale,
- ✓ Enfin, INSIGNIA ne propose que deux classes de service (temps réel et best effort) et les applications multimédia adaptatives ne sont supportées et que la réservation de ressource ne peut être établie que lorsque le trafic est lancé.

III.5.3.3. Protocole BruIT (Bandwidth Reservation under InTerferences influence)

BRuIT [112] essaye d'apporter de la QdS dans les réseaux ad hoc en limitant l'impact des interférences sur les communications entre les nœuds. Il effectue une réservation de la bande passante qui prend en compte une connaissance totale des interférences en se basant sur un protocole de routage avec QdS.

Afin de résoudre le problème des interférences causées par des nœuds éloignés, il essaye d'apporter une connaissance sur le voisinage pour les nœuds des réseaux ad hoc. Ce modèle suppose que chaque mobile a une connaissance sur l'ensemble du réseau.

BRuIT est un protocole de signalisation distribué qui atteint cet objectif par un envoi périodique de messages contenant des informations sur la disponibilité de bande passante pour les transmissions. Il vise à mieux *contrôler* la bande passante dans le but d'empêcher au maximum l'apparition de congestion dans le réseau afin de *fournir* la bande passante demandée pour deux types de flux.

BRuIT considère deux types de flux : ceux qui n'auront aucune garantie sur leur débit dits best effort et ceux à qui on peut *réserver* une certaine bande passante dits privilégiés [15][90][113].

Le fonctionnement de BruIT s'appuie sur deux phases (en soft state) [112]:

1- D'abord une phase de "*découverte des voisins*" qui leur permet de s'échanger leur état de charge respectif (i.e. la valeur totale de leur bande passante déjà réservée). Cette phase permet à chaque nœud de disposer de l'état de charge de son environnement radio.

2- Ensuite vient la phase de *réservation de ressources* nécessaires au flux. En fonction de la bande passante disponible et de la charge du medium radio, un *contrôle d'admission* est effectué au niveau de chaque nœud.

Pour effectuer ces *réservations* et ce *contrôle*, BRuIT tente d'apporter régulièrement à chaque nœud suffisamment de connaissance sur la bande passante qui est utilisée dans son voisinage étendu (l'ensemble de ses voisins à un et deux sauts). A la base de cette information (connaissance), un nœud peut estimer la bande passante utilisée pour les flux privilégiés dans son voisinage étendu. À partir de là, chaque nœud peut décider de *l'admission* ou du *rejet* d'un flux privilégié qui nécessite une certaine bande passante. Ceci permet donc aux nœuds de n'accepter que les flux dont ils seront initialement en mesure d'honorer leur débit et donc d'empêcher, très souvent, l'apparition de congestion [90].

III.5.4. Routage avec QdS dans les MANETs

III.5.4.1. Introduction

Le routage dit au mieux ou best effort consiste à trouver le plus court chemin entre une source et une destination en terme de nombre de sauts. Cependant le routage avec QdS (QdS Routing) est un élément clé pour réaliser une architecture de QdS pour les MANETs. Il peut être défini comme le mécanisme par lequel les chemins associés aux flux sont déterminés à la fois par la connaissance des ressources disponibles et par les demandes en termes de QdS pour ces flux. Pour le routage avec QdS, on ajoute un certain nombre de contraintes (délai, bande passante, fiabilité, le coût de transmission, etc..) sur les routes afin de déterminer leur éligibilité (i.e. si elles sont admissibles ou non). En effet, toute route satisfaisant un certain critère quantitatif ou qualitatif peut être qualifiée de route assurant une certaine QdS [15][114][115].

Selon le type de contraintes, la recherche de routes optimales peut devenir un problème NP-complet. Les routes doivent être calculées par flux et non par destination. En effet, un flux peut avoir des besoins de QdS alors qu'un autre flux entre ces mêmes nœuds en aura d'autres.

Enfin, un protocole de routage ad hoc avec QdS doit pouvoir réagir très rapidement aux changements de topologie et aux conditions de QdS sans que les applications ne soient atteintes. Le but de ce type de protocole est donc de trouver une route dans le réseau qui puisse satisfaire de bout-en-bout les besoins en QdS demandés par une application. C'est une alliance entre un protocole de routage classique et un mécanisme de gestion des ressources.

La mobilité ou le manque d'énergie peuvent causer des ruptures dans les chemins établis, le protocole doit donc être capable de réagir très vite à ce genre d'événement en recalculant des routes valides. L'idée est donc de trouver un compromis entre le gain apporté par le routage QdS et la charge de contrôle générée. Plusieurs solutions on été proposées dans les MANETs [90]. Parmi ceux qu'on va décrire dans les sections suivantes, on cite : CEDAR, QOLSR, AQOR et TBR.

III.5.4.2. CEDAR (Core-Extraction Distributed ad hoc Routing Algorithm)

CEDAR [95][116] est un protocole de routage réactif basé sur la notion de réseau de cœur. Il semble bien faire face à la mobilité des nœuds dans les MANETs pour fournir une QdS en terme de bande passante.

Les nœuds choisis pour faire partie du réseau de cœur forment le Dominating Set (DS). Chaque nœud du réseau est soit un nœud de cœur où l'un de ses voisins et donc, il fait partie du DS.

Le rôle des nœuds de cœur est de propager efficacement les informations sur la bande passante disponible dans les liens, d'assurer le routage dans le réseau avec le minimum de nœuds et de limiter autant que possible les diffusions. La distance de propagation des informations dépend de la qualité du lien en termes de stabilité et de bande passante disponible. Le chemin entre nœuds du cœur est appelé lien virtuel [15][90].

CEDAR opère en trois phases essentielles (Figure 3. 9):

1. Core extraction (Extraction d'un cœur du réseau) [90]: Lors de cette phase, le réseau choisi dynamiquement un ensemble de nœuds qui feront parti du DS pour calculer les routes et maintenir l'état des liens du réseau où seul les nœuds de cœur participeront au calcul.

2. Link state propagation (Propagation de l'état des liens) [15][116] : Les liens stables (de grande capacité) sont considérés à forte bande passante et sont propagés dans le cœur du réseau tandis que les liens moins performants (de faible capacité) reste connus seulement au niveau local (ne sont diffusés que localement).
3. Route computation (Calcul de routes) [15][74]: basé sur la découverte et l'établissement d'un plus court chemin stable vers la destination pour garantir la bande passante demandée (entre les représentants de cœurs). Une reconstruction est initiée quand la route principale est cassée. Elle peut être locale (à l'endroit de la cassure), ou à l'initiative de la source.

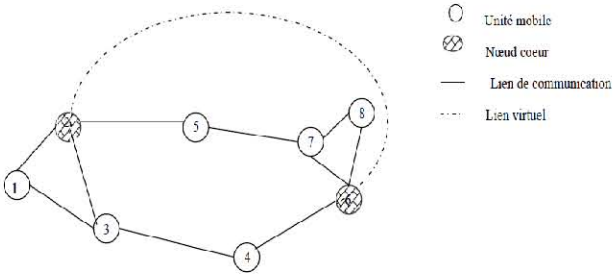


Figure 3. 9: Types de nœuds dans CEDAR [74]

CEDAR repose sur l'utilisation de protocoles d'accès au canal de type CSMA/CA et semble donc être une bonne solution pour des réseaux à faible mobilité. Mais, le fait qu'il effectue un routage à la source à la fois pour les paquets de contrôle et de données, freine ses performances de manière importante. D'autre part, CEDAR ne gère que la bande passante ce qui peut être très insuffisant pour certaines applications ayant d'autres exigences de QdS (délai ou gigue) [15][90].

III.5.4.3. QOLSR: QdS pour OLSR (Optimized Link State Routing)

Introduction :

QOLSR (QdS pour OLSR) [117] est une extension du protocole OLSR pour répondre aux exigences de QdS dans les réseaux mobiles ad hoc sans aucuns messages de contrôle additionnels.

Des champs additionnels pour la QdS sont rajoutés aux messages de contrôle (*Hello* et *TC*). De façon périodique, chaque nœud transmet des informations sur ses MPR (relais multipoints : voisins à 1-saut) et les conditions de QdS, pour leur permettre de construire les chemins via ces MPRs.

QOLSR détermine les routes avec une bande passante disponible maximale et un délai minimal. Il ne fournit aucune forme de sécurité, ni des mécanismes de contrôle d'admission et de réservation. IL applique un mécanisme de vérification régulière de conformité du flux aux exigences de QdS suite au phénomène de mobilité où par ajout de trafics supplémentaires dans le réseau [118].

Fonctionnement de QOLSR :

QOLSR est un protocole de routage proactif avec QdS qui détermine en cas de besoin des routes optimales en termes de bande passante et de délai.

Seuls les MPRs sont habilités à transmettre entre eux, l'information de QdS dans les messages TC [90]. Pour assurer un routage avec QdS, QOLSR exécute un ensemble de fonctions [117]:

- **La détection de voisinage**: chaque nœud doit détecter les nœuds voisins à 1-saut sur un lien direct et bidirectionnel ou symétrique par diffusion périodique d'un paquet Hello avec (TTL=1) contenant des informations relatives aux voisins et l'état des liens aux nœuds situés seulement à 1-saut.
- **Mesure de QdS des voisins** : chaque nœud doit évaluer les conditions de QdS (débit et délai) sur les liens directs et symétriques vers chacun de ses voisins. Ensuite, les informations de QdS sont transmises localement dans les champs de QdS réservés spécialement pour cet usage dans les messages *Hello*.
- **Sélection des MPRs**: chaque nœud du réseau sélectionne de façon indépendante son propre ensemble de MPRs qui répondent aux exigences de QdS. Pour déterminer cet ensemble qui devrait être de petite taille pour réduire le nombre de retransmissions redondantes, les informations requises sont échangées périodiquement dans les messages Hello. Cet ensemble est recalculé quand il y a un changement d'un lien symétrique (disparition ou changement des conditions de QdS) vers un voisin à 1 ou à 2-sauts.

- **Echange des informations sur les MPRs et sur les conditions de QdS** : chaque nœud dans le réseau entretient des informations topologiques (la liste de ses voisins l'ayant choisi comme MPR) et les paramètres de QdS contenues dans les messages TC qui sont diffusés à intervalles réguliers entre MPRs pour permettre à chaque nœud de construire sa table de routage.
- **Construction de la table de routage** : une entrée de la table de routage contient les informations [*nœud, source, besoins en QdS*]. Cette table est mise à jour selon les exigences d'une application ou de l'utilisateur (i.e. un changement dans les ensembles des voisins ou de topologie).

III.5.4.3. TBR: Ticket Based QdS Routing

TBR [119] est un protocole de routage distribué, qui autorise des informations d'état imprécises durant la phase de découverte de chemins. TBR a été conçu pour des réseaux avec une mobilité suffisamment faible ce qui permet d'avoir des routes avec une durée de vie plus grande devant le temps nécessaire à leur établissement ou leur restauration par une technique de réparation locale.
TBR peut supporter différentes contraintes de QdS sensible au délai et à la bande passante.

Chaque nœud maintient, en utilisant une transmission périodique de paquets de signalisation (sondes), les caractéristiques des liens vers ses voisins immédiats (délai, débit, coût du chemin traversé).

Dans TBR, le problème d'établissement de routes, les plus proches de l'optimal, de moindre coût avec des contraintes de délai est NP complet, et avec des contraintes de débit est solvable en temps polynômial.

TBR utilise l'historique et l'estimation des variations du délai, et une formule de lissage pour calculer le délai courant. La réparation de routes se fait en utilisant des reconstructions locales.

L'inconvénient de TBR est que chaque nœud doit garder les informations d'état pour chacun de ses voisins, chose qui fait défaut au réseau mobile ad hoc [115][120].

III.5.4.4. AQOR: Ad hoc QdS on-demand routing

Le protocole AQOR [121] minimise l'inondation du trafic de contrôle pour découvrir la meilleure route disponible en termes de plus petit délais de bout en bout et avec une garantie de bande passante. Un paquet de requête de route comprend les contraintes de QdS.

Si un nœud ne peut satisfaire les deux contraintes, il rediffusera la requête vers le prochain saut et passera à un statut « exploré » pendant une courte période de 2*Tmax (Tmax qui désigne la contrainte de délai). La destination retournera un paquet de réponse le long de chacune des routes suivant lesquelles, elle à reçue les requêtes. Les nœuds intermédiaires avec un statut « exploré » ne transmettent pas de réponses. Donc, la réservation de la bande passante pour chaque flux est activée uniquement dès l'arrivée du premier paquet de données à partir du nœud source. Le Délai est mesuré au cours de la phase de découverte de routes et le chemin avec le plus court délai est choisi par la source.

Aucun mécanisme de libération connexion est nécessaire ou intégré, car toutes les réservations ne sont que temporaires. Les temporisateurs (horloges) sont remis à zéro chaque fois qu'une route est utilisée. Les routes cassées ne seront détectées qu'après un temps limite supérieure. Pour réduire davantage le coût de communication au cours de la découverte de routes, AQOR peut travailler avec des protocoles de routage avec détermination d'emplacement.

Pour détecter la violation de délai, le temps de décalage estimé entre les horloges de la source et le de la destination doit être connue [115][122].

III.5. Travaux récents

III.5.1. Description

Ces dernières années, un grand nombre de travaux sur la QdS ont été proposés. Dans cette section, on va exposer un certain nombre d'entre eux.

Le travail proposé dans [123] publié en mars 2011 fait partie des protocoles de routage avec QdS à base d'une nouvelle métrique "entropy" basée sur l'énergie et la taille des tampons pour garantir une QdS maximale, augmenter la durée de vie du réseau, et assurer la probabilité de succès d'une transmission de données.

Plus l'énergie d'un nœud est grande, plus qu'il a une meilleure probabilité de servir le chemin pour une longue période de temps. Mais en pratique, il ne suffit pas qu'un nœud avec une grande quantité d'énergie

est plus adapté à la transmission. La sélection d'un nœud est déterminée en fonction de la probabilité qu'il sera en mesure de fournir un support pour tous (le maximum) les paquets avec une consommation minimale de l'énergie.

La qualité d'une transmission varie d'une application à une autre et dépend de nombreux paramètres différents (délai, débit et taux d'erreur). Par conséquent, le succès de la transmission dépend de la distribution conjointe de probabilité de ces paramètres et l'entropie ou de la variation de cette distribution au cours du temps.

Dans ce protocole, chaque nœud calcule une métrique "entropy" basée sur ses propres informations relatives à l'énergie et sur le nombre de paquets à transmettre. Les nœuds dont la valeur de l'entropie est grande (ceux répondant aux critères de QdS) seront sélectionnés et les autres seront rejetés.

Par conséquent, la probabilité avec laquelle un nœud peut transférer avec succès plusieurs paquets ne dépend pas seulement de son énergie et la distance entre ses voisins, mais aussi des autres flux sur les autres routes et de la taille de la file d'attente (tampon) du nœud à un instant donné. Les paquets de contrôle sont transmis avec une grande probabilité.

Le système proposé est plus performant et le taux moyen de consommation de l'énergie dans le réseau est réduit ce qui améliore la durée de vie du réseau.

Le protocole SQR-AODV (Stable QoS-aware Reliable on-demand distance vector routing protocol) pour les réseaux ad hoc proposé dans [124] publié en juillet 2011 n'est autre qu'une amélioration du protocole AODV pour le support de la QdS. Il tente de déterminer le chemin le plus fiable entre source et destination. Il fait partie de la catégorie des protocoles de routage avec QdS avec une différentiation de services.

La durée de vie d'une route, l'énergie restante, et le nombre de sauts sont les trois paramètres utilisés pour la sélection d'un chemin avec une grande stabilité, un niveau d'énergie suffisant et une faible latence.

Pour déterminer la stabilité des chemins, SQR-AODV utilise la durée de vie d'un lien LLT (link life time) entre deux nœuds reliés à base de l'information de mobilité des nœuds obtenue par GPS (système de positionnement global). Il utilise les mêmes phases de découverte et maintien de routes comme ceux de l'AODV tout en tenant compte des paramètres cités avant.

Le protocole SQR-AODV assure une grande fiabilité avec un taux de livraison et un débit élevé des paquets. Il offre une gestion efficace de l'énergie et de l'équilibrage de charges, ce qui fait prolonger la durée de vie du réseau et rendre les communications plus fiables.

Le protocole QMBR-AOMDV [125] publié en 2011 basé sur le protocole MRB (Multipath Routing Backbone) et le protocole AOMDV [126] (AODV multi chemins) entre dans la catégorie des protocoles de routage avec QdS. C'est un protocole de routage multi chemins dont l'objectif est d'améliorer la fiabilité, le débit et l'équilibrage de charge.

Les protocoles de routage multi chemins sont plus appropriés à la gestion de l'équilibrage de charges et sont tolérant aux pannes par rapport à ceux à chemin unique qui sont moins efficace surtout pour l'économie de la bande passante et la réduction du délai.

La capacité statique des ressources (SRC), la disponibilité dynamique des ressources (DRA), la qualité du voisinage (NQ) et la qualité et la stabilité de lien (LQS) sont quatre métriques utilisées pour le support de QdS (QSMs) et pour différencier les nœuds du réseau en fonction de leurs caractéristiques et identifier ceux qui peuvent prendre part dans la construction du chemin dans le protocole MRB.

Dans ce protocole, plusieurs chemins à liens disjoints sont déterminés par le protocole AOMDV et le chemin avec la plus haute métrique est sélectionné par le protocole de routage MBR comme route principale pour la transmission des données.

La diffusion de l'information requise pour supporter la QdS est véhiculée dans les paquets Hello échangés régulièrement entre nœuds voisins et contenant les paramètres SRC, DRA, NQ et sa bande passante disponible (BW). Le protocole QMBR-AOMDV permet d'augmenter la fiabilité (un taux élevé de livraison de paquets avec des délais réduits) et de mieux gérer l'équilibrage de charges.

Le protocole T-MAC (Throughput-aimed MAC Protocol with QoS Provision for Cognitive ad hoc Networks) proposé dans [127] en juin 2010, propose un support pour la QdS orienté débit destiné aux réseaux ad hoc cognitifs. Il utilise la technique d'accès au support TDMA (Time Division Multiple Access) associé à un mécanisme de contrôle de puissance pour améliorer la QdS au niveau de la couche MAC.

Afin de résoudre certains problèmes non encore complètement résolus (i.e. la protection simultanée des paquets de données et d'acquittements, le problème des stations cachées/exposées multi canaux), ce protocole définit une super trame pour réduire la probabilité de collision des paquets de contrôle, utilise un mécanisme de contrôle de puissance pour la prise en charge des transmissions concurrentes, et utilise des trames spéciales chargées de protéger les paquets ACK.

Dans ce protocole, on trouve les utilisateurs primaires (UPs) et les réseaux ad hoc cognitifs (CAHNs[2]). Les UPs sont autorisés à utiliser les différentes bandes de fréquences pour la communication. Les CAHNs ne sont autorisés qu'à utiliser les fréquences libres (i.e. non utilisés par les PUs). Les PUs utilisent le canal de façon déterministe selon la méthode TDMA. Par contre, les CAHNs scrutent le canal au début de chaque intervalle pour déterminer quelle bande de fréquence est libre pour l'utiliser selon un mode de réservation avec les phases : demande/confirmation de réservation et la décision d'émettre.

Le protocole T-MAC met en œuvre une synchronisation stricte non seulement entre les nœuds CAHNs, mais aussi entre CAHN et PUs. En pratique, pour garantir une synchronisation précise, la technique de localisation par satellite (GPS) est utilisée malgré qu'elle introduise une charge de contrôle supplémentaire.

Ce protocole permet d'améliorer le débit grâce à la technique de gestion déterministe du canal (TDMA) pour les PUs et l'accès compétitif entre CAHNs sur les bandes de fréquences disponibles.

Le protocole SMQR (Stabilité-Based Multipath Route QoS Protocol) pour réseaux mobiles ad hoc proposé dans [128] publié en aout 2010 est un protocole de routage multi chemins avec QdS. En plus des métriques de débit et de délai, SMQR utilise une nouvelle métrique de QdS appelée stabilité de liens à base du modèle RSM (Route Stability Model) qui se base sur la mobilité des nœuds et la puissance du signal pour calculer la probabilité de défaillance des liens.

SMQR détermine au maximum trois chemins à nœuds disjoints avec QdS. Celui avec une stabilité maximale est utilisé comme chemin principal et les autres seront considérés des chemins secondaires ou de secours en utilisant les paquets de contrôle de l'AODV [13] avec des champs supplémentaires pour gérer le QdS (QRREQ, QRREP et RERR) et un paquet spécial RouteM pour maintenir les routes secondaires selon les exigences de QdS. Il utilise aussi un paquet Hello pour le maintien du voisinage à un saut et la collecte des informations sur la stabilité et la bande passante.

Un contrôle d'admission sur le débit et le délai est effectué saut par saut lors de la transmission des paquets QRREQ/QRREP pour contrôler si les nœuds répondent aux exigences de QdS. Les requêtes non satisfaites seront soit abandonnées soit autorisées à s'exécuter dans le future.

Après une découverte réussie de chemins de routage avec QdS, la source admet le trafic temps réel sur le chemin principal et deux types de maintenance de routes sont appliqués: l'une est due à la violation des exigences de QdS sur le chemin primaire et l'autre sur les chemins alternatifs de façon continue.

Le protocole SMQR se classe dans la catégorie des protocoles de routage multi chemins avec QdS pour réseaux ad hoc en appliquant une différentiation de services pour flux temps réel, flux avec des exigences en termes de délai et de débit et ceux best effort. Il minimise la charge de contrôle, fait gagner le temps de reconstruction de routes, fait accroitre la fiabilité, et tente de garder la connectivité pour une durée maximale (i.e. presque permanente) entre nœuds d'une communication active par utilisation du multi chemins et la stabilité de routes.

Le protocole CBQR (Cluster Based QoS Routing Protocol) proposé dans [129] publié en octobre 2010 fait partie des protocoles de routage avec QdS de type DiffServ pour les réseaux ad hoc. Il améliore la QdS pour les paramètres bande passante et délai en découpant le réseau en groupes ou clusters. Permettant ainsi la réduction de l'espace de stockage, la charge de contrôle, le traitement des données, et le gain d'énergie.

Dans CBQR, lorsqu'un noeud source désire transmettre des données, il vérifie l'existence du nœud destination dans son cluster. Si c'est le cas, les données sont directement transmises. Dans le cas contraire, le paquet de données est envoyé à son ClusterHead qui à son tour procède de la même manière. Si aucune correspondance n'est trouvée, il vérifie s'il existe un nœud où la bande passante nécessaire est disponible, le paquet de données est envoyé vers ce nœud. Ce processus se poursuivra jusqu'à ce que le nœud destination soit atteint ou si le nombre de nœuds visités restera inférieur à la valeur du champ TTL (Time to Live).

2 : Cognitive Ad Hoc Networks

Nous pouvons dire que la structuration du réseau en clusters est d'un apport considérable dans le routage et dans l'offre de QdS en termes de débit ou de délai implémenté dans le protocole CBQR.

Le protocole Q-PAR (QoS Based Power Aware Routing in MANETs) proposé dans [130] publié en avril 2009 fait partie des protocoles de routage avec QdS avec des exigences en termes d'énergie et de débit. Il détermine un petit nombre de chemins satisfaisant les contraintes de QdS et l'efficacité énergétique, ce qui limite le problème d'overhead.

Q-PAR, utilise un modèle simple pour le calcul de la consommation de l'énergie à différents intervalles de temps. Il opère en deux phases : l'évaluation des contraintes de QdS (i.e. bande passante et énergie) durant la phase de découverte de route à base du protocole DSR et la phase de réparation locale activée pour déterminer une route alternative satisfaisant la contrainte d'énergie en cas d'échec de liaison.

La maintenance dans Q-PAR se fait soit par rapport à l'énergie des nœuds d'une route active, soit par rapport au changement de topologie (mobilité des nœuds).

Un module de contrôle d'admission pour estimer a priori de la bande passante et s'assurer de sa disponibilité entre les liaisons sans fil est utilisé par Q-PAR pour offrir une meilleure performance au réseau.

Le protocole de routage NDMLNR (Node Disjoint Multipath Routing considering Link and Node Stability) publié dans [131] en octobre 2009 fait partie de la catégorie des protocoles de routages avec QdS de type DiffServ. Il détermine plusieurs chemins à nœuds disjoints en se basant sur la stabilité de nœuds et de liens pour permettre une connectivité maximale des liaisons.

Pour mesurer la stabilité de liens et de nœuds, deux métriques sont utilisées : le temps d'expiration de lien (LET : Link Expiration Time) qui dépend directement du facteur de mobilité (vitesse, direction) qui sont échangés entre les nœuds à intervalles réguliers par le biais du GPS et le taux d'épuisement d'énergie (EDR : Energy Drain Rate) qui dépend des émissions et réceptions de données. Leur composition permet de garder trace du niveau de stabilité du chemin.

Le degré de stabilité de liens LSD (Link Stability Degree) est le rapport entre le facteur de mobilité et le facteur énergétique. Plus la valeur de LSD est élevée, plus, la stabilité de liens est élevée et la durée de son existence est plus grande.

NDMLNR se base sur le protocole DSR [45] par ajout des extensions (LSD et bande passante : B) aux paquets de contrôles RREQ et RREP afin de déterminer les routes à liens stables et à nœuds disjoints.

Après découverte de plusieurs chemins selon les conditions prédéfinies, ce protocole sélectionne l'un d'entre eux selon la plus grande valeur de l'énergie et du nombre de sauts.

Dans le cas ou le LSD d'un lien descend au dessous d'un seuil prédéfini, le nœud informe son voisin, qui à son tour répète cette action jusqu'à ce que la source soit avertie pour qu'elle balance ses transmissions vers une route de secours.

Le protocole NDMLNR permet d'optimiser la consommation de l'énergie sur des chemins à nœuds disjoints plus stables.

Le protocole MSR (Multipath Source Routing) [132] publié en octobre 2009, faisant partie de la catégorie des protocoles de routages avec QdS de type DiffServ, est une extension du protocole DSR pour déterminer plusieurs chemins à nœuds disjoints selon les exigences de bande passante et les contraintes de fiabilité. Il opère en trois phases : la découverte, la maintenance et la répartition du trafic.

Dans la phase de maintenance, MSR veuille à ce que les conditions de QdS soient toujours assurées par échange de messages entre voisins, sinon une nouvelle découverte est relancée.

MSR réparti les paquets d'un flux sur les différentes routes en conformité avec leurs exigences de QdS. La destination doit être capable de réordonner et réassembler les paquets d'un flux et en cas d'échec, demander la retransmission des paquets perdus. Le délai des transmissions (i.e. la différence entre le temps de réception du premier et du dernier paquet) dépend du temps de reconstruction de paquets.

Le protocole MSR avec la bonne répartition des paquets d'un flux sur plusieurs chemins, réduit le délai, économise l'énergie ce qui fait prolonger la durée de vie du réseau et assure une grande fiabilité.

CLQM [133] (Cross-Layer Qosframe Work for MANETs) proposé en 2008 est un modèle cross-layer pour le support de la QdS dans les réseaux mobiles ad hoc avec des garanties strictes par classe de service en termes de délai et de débit.

Le modèle CLQM opère dans les trois couches (MAC, réseau et application). La couche MAC est responsable de l'ordonnancement et l'allocation du médium sans fil ; elle traite les paramètres de QdS de type délai et bande passante disponible. Le protocole de routage (couche réseau) sélectionne les liens appropriés selon les exigences de QdS (qualité du chemin de bout en bout) pour transmettre les paquets vers la destination. Enfin, la couche application traite les exigences de QdS de l'utilisateur qui sont spécifiés en classes de services avec des garanties prédéfinis en termes de délai, de débit et best effort.

L'objectif global de ce modèle est de sélectionner les chemins avec des ressources suffisantes pour fournir une garantie de QdS pour le trafic de haute priorité et d'adapter de façon dynamique le comportement de la couche MAC et la couche réseau afin de maintenir les garanties de QdS pour les flux déjà admis sur la base de la charge du réseau et la QdS assurée aux différentes classes de services dans le réseau. Le modèle proposé par la QoS fournit une différenciation de service par classes.

Basé sur les exigences de QdS des utilisateurs, le trafic est répertorié en quatre classes de service (classe I, II, III et IV) qui correspondent respectivement à ceux sensibles au délai, au débit et délai, au débit et best-effort.

Dans CLQM, la couche de transport ne fait aucunes actions sur les paramètres de QdS, elle ne fait que les passer à la couche réseau pour trouver un chemin de bout en bout avec la qualité souhaitée.

CLQM propose un modèle cross layer pour le support de QdS avec une différenciation des services par classe dans les réseaux ad hoc.

Le protocole proposé dans [134] publié en mai 2008 fait partie des travaux sur la QdS au niveau de la couche MAC. Il est basé sur la combinaison de la technique CDMA (Code Division Multiple Access) capable de supporter plusieurs transmissions simultanées et le schéma de passage à jeton (token) pour l'accès au médium dans les réseaux sans fil ad hoc.

Il offre au réseau une grande stabilité, un support de QdS pour les applications de multimédia et de temps réel avec une utilisation rationnelle des ressources.

La nouveauté dans ce protocole est qu'il combine les avantages de la garantie d'accès du mécanisme de passage à jeton et la prise en charge des transmissions de paquets multiples au sein du réseau avec des exigences de QdS pour les différentes classes de trafic hétérogènes.

Il opère dans un réseau distribué et décentralisé avec une topologie composée de n nœuds (stations) avec m codes (SM) pour CDMA ($m < n$) et différentes classes de trafic, où chaque station est attribuée une classe de trafic spécifique et une file d'attente. La circulation du jeton utilisé pour la distribution des m codes CDMA dans le réseau se fait selon une topologie en anneau virtuel.

Les autorisations générées (permis) pour la transmission dépendent de la bande passante disponible au niveau de chaque nœud qui peuvent transmettre après capture d'un jeton et la disponibilité d'un code SM, un nombre de paquets égale au nombre de permis générés.

Ce protocole grâce à l'indépendance entre le temps de rotation du jeton et la charge du réseau, conduit à une distribution efficace des codes entre tous les nœuds ce qui permet d'assurer des garanties d'accès pour chaque nœud par le schéma à jetons et une disponibilité des transmissions multiples.

III.5.2. Discussions

Basé sur l'étude détaillée de ce nombre de travaux, on constate que :

1. La majorité d'entre eux assurent une différentiation de services que ce soit par flux ou par classes.
2. Pour la catégorie des protocoles de routages avec QdS, chaque proposition tente d'assurer une connectivité avec une durée de vie maximale en se basant sur la notion de multi chemins, de la stabilité des liens et de l'énergie des nœuds. Les transmissions de données se font en:
   a. Parallèle pour des différents flux chacun selon ses exigences de QdS, où pour les fragments d'un seul flux sur différents chemins ce qui permet de gérer efficacement la bande passante et réduire le délai,
   b. Utilisant un chemin pour le transfert et les autres en tant que routes de secours qu'on utilisera en cas d'échec de liaison.
   c. Choisissant les nœuds stables pour les utiliser dans les chemins à nœuds disjoints pour minimiser la consommation de l'énergie.

3. Pour la catégorie des protocoles de la couche MAC, une combinaison des méthodes d'accès déterministes (CDMA, TDMA et passage à jeton) et aléatoires permettent de réduire les collisions, d'augmenter le débit et de diminuer le délai.
4. Pour les solutions cross-layer, le concours d'un ensemble de mécanismes répartis sur les différents nivaux (application, réseau, liaison et MAC) permet à mieux supporter la QdS que ce soit pour les besoins sensibles au délai, au débit où à une combinaison des deux.

III.6. Conclusion

Dans ce chapitre, après avoir décrit les paramètres de QdS, nous avons présenté les architectures proposées par les groupes de travail IntServ et DiffServ et qui se sont montrés inadaptés aux réseaux ad hoc mais servent de support pour la majorité des travaux dans ce contexte. Ensuite, nous avons exposé différents modèles de QdS comme FQMM qui combine les propriétés des modèles filaires IntServ et DiffServ, en offrant une méthode d'approvisionnement hybride: par flux (trafics prioritaires) et par classes (autres flux). Le modèle CEQMM qui combine les solutions proposées dans les réseaux filaires et les adapte au contexte des réseaux ad hoc (modèle complet et efficace de QdS). La modèle SWAN qui se base sur des algorithmes de contrôle distribués, et le modèle QPART qui rejoint le même état d'esprit que SWAN, ceci par rapport à la différentiation de services entre différents types de trafic (sensible au délai, au débit et best effort).

Dans la catégorie de routage avec QdS, on trouve le protocole CEDAR basé sur la notion de réseau de cœur. Il fournit une QdS en terme de bande passante quant à BRuIT, il vise à mieux contrôler la bande passante en limitant l'impact des interférences sur les communications entre les nœuds. Il considère deux types de flux (best-effort et privilégié). Dans le même état d'esprit, on trouve TBR un protocole de routage distribué conçu pour des réseaux à faible mobilité qui peut supporter les contraintes de QdS sensibles soit au délai soit à la bande passante. On trouve aussi dans cette catégorie, le protocole QOLSR qui est une extension du protocole OLSR pour supporter les exigences de la QdS sans aucun message de contrôle additionnel. Il détermine les routes avec une bande passante maximale et un délai minimum sans fournir des mécanismes de contrôle d'admission et de réservation.

Dans la catégorie des protocoles de signalisation, on trouve INSIGNIA avec son mécanisme de réservation soft. Il offre des garanties pour flux best effort et temps réel.

Concernant la couche MAC, plusieurs travaux ont été réalisés pour supporter la QdS reposant sur l'amélioration de la DCF du protocole 802.11 (différents facteurs d'incrémentation du temps de backoff, variation de DIFS, différentes tailles minimales de la CW, longueur maximale de paquets et BlackBurst).

Pour répondre aux besoins sans cesse des nouvelles applications, la norme 802.11e a défini de nouveaux mécanismes d'accès au médium EDCA (améliore la procédure DCF pour fournir la QdS de niveau priorité) et HCCA (améliore la PCF et introduit la notion d'opportunité de transmission reposant sur un système de vote pour répondre aux contraintes de débit, de priorité, de délai et de gigue).

La dernière section a été consacrée à la description de quelques travaux récents (2008 - 2011) sur la QdS pour mieux situer notre proposition.

Les différentes approches décrites ci-dessus permettent d'offrir des mécanismes de QdS afin d'effectuer une différenciation de services entre les trafics, et une allocation de ressources tout en assurant une utilisation optimale et équitable.

Le contexte de nos travaux est l'amélioration de la QdS au niveau routage et dans la couche MAC. Notre choix s'est porté sur le protocole AODV pour plusieurs raisons (sa nature réactif, le plus cité dans la littérature, et il est l'un des protocoles en cours de standardisation).

Partie : Contribution

Chapitre 4 : Description et simulation du protocole AODV

IV.1. Le Protocole de routage AODV

Le protocole AODV (ad hoc On-demand Distance Vector Routing Protocol), est un protocole de routage réactif multi-sauts [135]. Il est basé sur le principe des protocoles de routage vecteur à distance. Il est conçu pour les réseaux ad hoc (MANETs) pour une population de dix à mille nœuds opérant sous différentes types de mobilité (faible, moyenne et relativement élevée).

L'AODV représente essentiellement une amélioration de l'algorithme proactif DSDV [42], en réduisant le trafic de contrôle, et cela en créant les routes lors du besoin, contrairement au DSDV, qui maintient la totalité des routes dans un but d'améliorer l'extensibilité (facteur d'échelle) et la performance. Ce protocole permet aux nœuds mobiles d'obtenir rapidement des routes pour de nouvelles destinations sans maintenir les routes pour lesquelles il n'existe pas de communication régulière active. Seuls les nœuds actifs (i.e. font partie du chemin utilisé) qui maintiennent les informations de routage. Il permet de répondre rapidement aux problèmes de rupture (coupure) de liens et au changement de la topologie.

L'AODV utilise le principe des numéros de séquence qui le fait distinguer des autres algorithmes et cela pour maintenir la consistance des informations de routage afin d'utiliser les routes les plus fraîches.

Ce protocole est opérationnel seulement dans un environnement où les liens sont symétriques. Il met en ouvre différentes opérations pour réaliser et maintenir le routage : gestion de la connectivité locale, phase de découverte et maintenance de routes, en plus du routage *nœud-par-nœud* [25].

L'AODV maintient les chemins d'une façon distribuée en gardant une table de routage, au niveau de chaque nœud faisant partie du chemin trouvé.

A cause de la mobilité des nœuds dans les réseaux ad hoc, les routes changent fréquemment et peuvent devenir invalides. Pour cela l'AODV, utilise un mécanisme d'expiration de routes associé à chaque utilisation d'une entrée dans la table de routage qui permet le nettoyage de cette dernière.

L'AODV a la particularité de :

- Diffuser les paquets de découverte uniquement lorsque cela est nécessaire.
- Faire la distinction entre la gestion de la connectivité locale et la maintenance de la topologie générale.
- Diffuser les informations lors de changements dans la connectivité locale aux nœuds voisins qui en ont besoin.

Comme il assure :

- une utilisation efficace de la bande passante (en minimisant la quantité d'information de contrôle sur le réseau).
- une réactivité aux changements de topologie.
- La prévention contre les boucles dans le réseau.

Si une nouvelle route est nécessaire, ou qu'une route disparaît, la mise à jour de ces tables s'effectue par l'échange de trois types de messages (paquets) entre les nœuds :

- RREQ: Route Request, un message de demande de route (initié lors de la découverte de route).
- RREP : Route Reply, un message de réponse à RREQ (utilisé pour finaliser la construction d'une route).
- RERR : Route Error, un message pour notifier au réseau la rupture d'un lien dans une route active (i.e. signale la perte d'une route).

On trouve, deux variantes pour le paquet RREP :

- Route Reply Message Acknowlegment : pour acquitter la bonne réception de données.
- Le message Hello : utilisé dans la phase de maintenance de routes actives (rafraichissement des liens) qui n'est autre qu'un RREP avec un TTL (Time To Live) égale à un.

Ces messages sont transportés dans des datagrammes UDP via le port 654 avec un en-tête IP traditionnelle. Le protocole n'interagit qu'avec les tables de routage des nœuds formant le réseau.

Le processus de découverte de routes se déclenche dès qu'un nœud source désire communiquer avec un nœud avec lequel il n'a aucune information de routage. Pour cela, il diffuse une requête RREQ (Figure 4. 1) via le réseau.

La réaction des voisins se fait par une réponse RREP (Figure 4. 2) au nœud source, si le nœud répondeur est concerné par la requête, sinon, la requête est transmise aux voisins, avec une sauvegarde d'une trace de l'information pour la construction du chemin de retour qui sera emprunté par le paquet RREP.

Le nœud source, après réception du paquet RREP, peut commencer l'émission des paquets de données.

La table de routage (Figure 4. 5) est crée pour maintenir temporairement les routes en sens inverse dites routes de retour pour les nœuds ayant générés un paquet RREQ.

Un lien est considéré actif tant que les paquets de données transitent périodiquement de la source (*Src*) à la destination (*Dst*). En absence d'émission de paquets de données pendant un temps défini, le lien expirera, et par conséquent il sera effacé des tables de routages des nœuds intermédiaires.

En cas de rupture de lien, le nœud qui se trouve à l'extrémité du lien avise les nœuds utilisant ce lien, qu'il n'est plus possible d'atteindre la destination en utilisant le message RERR [136].

IV.2. Format des paquets utilisés dans l'AODV

IV.2.1. Paquet Route Request (RREQ)

Ce paquet est schématisé par la figure suivante (Figure 4. 1) :

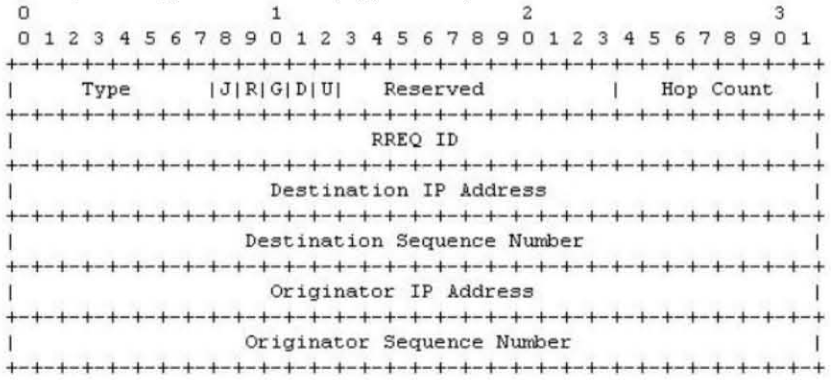


Figure 4. 1: Format général d'un paquet RREQ

Le paquet RREQ contient les champs suivants (Tableau 4.1).

Champs	Descriptions
Type	de valeur 1 pour indiquer que le paquet est un RREQ.
J (Join Flag)	Indicateur (drapeau) de liaison, utilisé en Multicast.
R (Repair Flag)	Indicateur de réparation, utilisé en Multicast.
G (Gratuitous RREP Flag)	Indicateur RREP gratuit qui doit être transmis en Unicast dans le nœud spécifié dans le champ « Destination IP Adress».
D (Destination only flag)	Indique que seule la destination peut répondre à cette requête RREQ.
U (Unknown sequence Number)	Indique que le numéro de séquence destination (NSeq_dst) est inconnu.
Reserved	Mis à zéro (0) à l'envoi et il est ignoré à la réception.
Hop Count	Indique le nombre de sauts (nœuds) traversés par la requête RREQ depuis la source jusqu'au nœud traitant la demande de route.
RREQ_ID	Un Numéro de Séquence (*NSeq*) identifiant le paquet RREQ généré par le nœud source (utilisé pour distinguer entre les RREQ déjà traités et les nouveaux RREQ non traités).
Destination IP Adress	L'adresse IP de la destination avec laquelle on désire établir un chemin.
Destination Sequence Number	Le dernier *NSeq* reçu par le nœud source avant cet instant de n'importe qu'elle chemin construit précédemment vers cette destination.
Originator IP Adress	L'adresse IP du nœud source ayant généré la requête RREQ.
Originator Sequence Number	Le numéro de séquence source (*NSeq_src*) (du nœud initiateur de la requête RREQ) figurant à l'entrée de ce nœud dans la table de routage du nœud traitant la requête.

Tableau 4. 1: description des champs du paquet RREQ

IV.2.2 Paquet Route Reply (RREP)

Ce paquet est schématisé par la Figure 4. 2 :

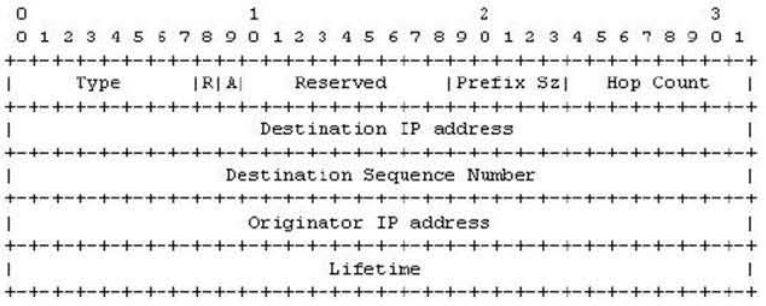


Figure 4. 2: Format général d'un paquet Route Reply (RREP)

Le paquet RREP est initié par le nœud destination ou par un nœud intermédiaire comme réponse à une requête RREQ. Le paquet RREP contient un chemin vers la destination sollicitée. Le tableau 4.2, décrit d'une façon plus détaillée les différents champs du paquet RREP :

Champs	**Descriptions**
Type	De valeur 2 pour indiquer que le paquet est un RREP.
R (Repair Flag)	Indicateur utilisé en Multicast.
A (Acknowledgment required)	Indicateur : s'il est activé, il indique que la réponse RREP nécessite un acquittement.
Reserved	Mis à zéro (0) à l'envoi et il est ignoré à la réception.
Prefix Size	S'il est non nul, les 5 bits de ce champ précise que le prochain nœud indiqué peut être utilisé pour tous les nœuds avec le même préfixe de routage (tel que défini par ce champ) comme destination demandée.
Hop Count	Indique le nombre de sauts (nœuds) de l'adresse IP source jusqu'à l'adresse IP destination.
Destination IP Adresse	L'adresse IP de la destination qu'on désire atteindre.
Destination Sequence Number	Le *NSeq_dst* associé au chemin.
Originator IP Adresse	L'adresse IP du nœud source ayant généré la requête RREQ.
Life Time	Le temps en millisecondes pour qui les nœuds ayant reçus le RREP considèrent le chemin comme valide (durée de vie).

Tableau 4. 2: description des champs du paquet RREP

Le 'A' bit est utilisé lorsque le lien sur lequel le paquet RREP est envoyé peut être peu fiable ou unidirectionnel. Lorsque le paquet RREP contient le bit 'A', le récepteur du paquet RREP devrait retourner un message RREP-ACK (Voir la section: Opération sur des liens unidirectionnels).

IV.2.3 Paquet Route Error (RERR)

Ce paquet est schématisé par la figure suivante (Figure 4. 3) :

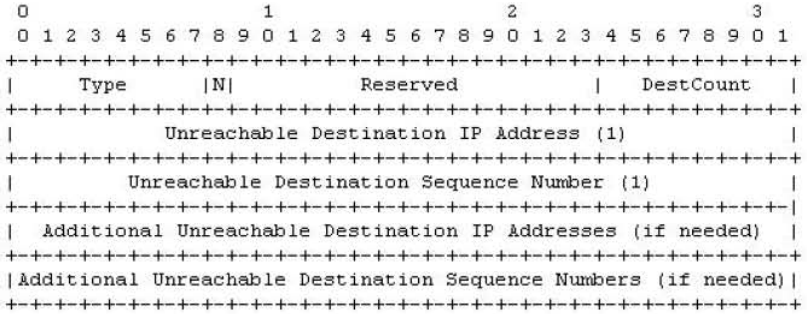


Figure 4. 3: Format général d'un paquet Route Error (RERR)

Ce paquet est émis chaque fois qu'une rupture de liens rend une ou plusieurs destinations injoignables par les nœuds prédécesseurs. Le paquet RERR contient les champs suivants (Tableau 4.3) :

Champs	Descriptions
Type	De valeur 3 pour indiquer que le paquet est un RERR.
N (No delete Flag)	Indicateur de non suppression : il est utilisé pour avertir les nœuds en amont d'un nœud qui a exécuté une réparation locale pour ne pas supprimer le chemin.
Reserved	Mis à zéro (0) à l'envoi et il est ignoré à la réception.
DestCount	Indique le nombre de destinations inaccessibles, il est inclus dans le paquet et doit être au moins égale à 1.
Unreachable Destination IP Adress (1)	L'adresse IP de la destination devenue inaccessible suite à une rupture de lien.
Unreachable Destination Sequence Number (1)	Le *NSeq* dans l'entrée de la table de routage pour la destination citée dans le champ Unreachable Destination IP Adress.

Tableau 4. 3: description des champs du paquet RRER

IV.2.4 Paquet Reply Aknowledgment (RREP-ACK)

Ce paquet est schématisé par la figure suivante (Figure 4. 4) :

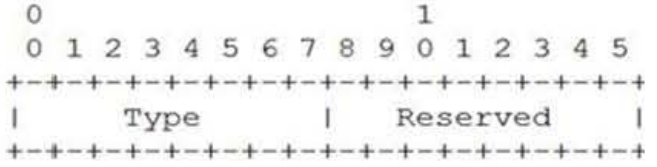


Figure 4. 4: Format general d'un paquet Route Reply Acknowledgment (RREP-ACK)

Le paquet accusé de réception (RREP-ACK) doit être envoyé en réponse à un message RREP quand le champ « A » est définit. Cette requête est invoquée si des problèmes sur les liens unidirectionnels empêchent la terminaison du cycle de découverte de routes. Il contient les champs suivants (Tableau 4.4) :

Champs	Descriptions
Type	De valeur 4 pour indiquer que le paquet est un RREP-ACK.
Reserved	Mis à zéro (0) à l'envoi et il est ignoré à la réception.

Tableau 4. 4: description des champs du paquet RREP-ACK

IV.3. Opération du protocole AODV

Cette section décrit les scénarios dans lesquels les nœuds génèrent les paquets de Route Request (RREQ), Route Reply (RREP) et Route Error (RERR) pour la communication en Unicast vers une destination, et comment les informations du paquet sont traitées. Dans le but de traiter les paquets correctement, certaines informations doivent être maintenues dans la table de routage de chaque nœud. C'est le cas pour les numéros de séquence destination et la liste des nœuds précurseurs (voisins).

IV.3.1 Gestion des numéros de séquence

L'AODV utilise le principe des numéros de séquence (*NSeq*) à fin de maintenir la consistance des informations de routage surtout en cas de mobilité des nœuds ou les routes maintenues par certains nœuds peuvent devenir invalides. Les numéros de séquence permettent d'utiliser les routes les plus nouvelles ou autrement dit les plus fraîches (fresh routes).

Pour chaque route, l'entrée dans la table de routage associée à chaque nœud, doit inclure les dernières informations valables concernant le *NSeq* du nœud destination pour lequel l'entrée de table routage et maintenue. Ce numéro est appelé « *NSeq_dst* ». Il est mis à jour toutes les fois qu'une nouvelle information provenant d'un message RREQ, RREP ou RERR est reçue par la destination.

Les numéros de séquence destination sont créés par le nœud destination et sont inclus (ajoutés) dans tous les messages retournés aux nœuds sources (demandeurs de routes). Ils permettent d'éviter le problème de création des boucles infinies et des transmissions inutiles de messages de contrôle sur le réseau.

La comparaison entre le *NSeq* associé au nœud et celui inclus dans le paquet de contrôle permet de détecter s'il s'agit d'une retransmission (i.e. le nœud a déjà été averti auparavant) ou si c'est une nouvelle information (i.e. reçue pour la première fois)

Un nœud destination modifie son propre *NSeq* dans deux situations :

- Juste avant qu'un nœud source lance une nouvelle découverte de route, il doit incrémenter son propre *NSeq*. Ceci permet de prévenir les éventuels conflits avec des routes établies précédemment en sens inverse vers le nœud source expéditeur d'une requête de découverte de route (RREQ).
- Juste avant qu'un nœud destination retourne une réponse RREP à une requête de découverte de route (RREQ), le *NSeq* doit être remplacé par le maximum entre son *NSeq* actuel et celui contenu dans le paquet RREQ.

La seule autre situation dans laquelle un nœud peut modifier son *NSeq dst* dans une de ses entrées de la table de routage est dans le cas d'un lien cassé (perdu) ou la durée de vie a expiré pour le prochain nœud vers la destination. Ce prochain nœud est déterminé par consultation de la table de routage du nœud. Dans ce cas là, pour chaque destination utilisant ce prochain nœud, les nœuds incrémentent le *NSeq* et marquent la route comme invalide.

Chaque fois qu'une nouvelle information de routage (i.e. contenant un *NSeq* au minimum égale au *NSeq* enregistré) pour une destination affectée est reçue par un nœud qui a marqué l'entrée de la table de routage comme invalide, le nœud devrait mettre à jour les informations associées à sa table de routage en fonction de l'information contenue dans la mise à jour.

Un nœud ne change son propre *NSeq* dans l'entrée de la table de routage pour la destination que si :

- il est lui-même le nœud destination et offre une nouvelle route pour l'atteindre; ou
- il reçoit un paquet de contrôle (RREQ, RREP, RERR) contenant de nouvelles informations sur le *NSeq* du nœud destination; ou
- le chemin vers une destination n'est plus valide ou est il cassé.

Quand un nœud reçoit un paquet de contrôle d'un voisin, ou crée ou met à jour une route pour une destination particulière, il examine sa table de routage de la disponibilité d'une entrée pour cette destination. Au cas où il n'y aurait aucune entrée pour cette destination, une nouvelle entrée est créée.

Lors du routage d'un paquet RREP, seulement les nœuds faisant partie du chemin peuvent mettre à jour leurs numéros de séquence pour la destination donnée si et seulement si :

- Le *NSeq* dans la table de routage est valide, ou
- Le *NSeq* dans le paquet RREP est supérieur à celui rangé dans la table de routage, ou
- Les numéros de séquence sont égaux, mais la route est marquée comme inactive, ou
- Les numéros de séquence sont les mêmes, mais le nombre de sauts (*hop count*) est inférieur pour le paquet RREP.

Les nœuds émetteurs du paquet RREQ doivent incrémenter leurs propres numéros de séquence avant la transmission du message RREQ, tandis que les nœuds destinations le font quand le *NSeq* dans le message RREQ est égal à leurs numéros rangés dans les entrées de table de routage.

IV.3.2 Gestion de la Table de routage et les listes de précurseurs

Le protocole AODV utilise une table de routage au niveau de chaque nœud de transit appartenant au chemin cherché. Une entrée dans cette table (Figure 4. 5) est créée lorsqu'un nœud reçoit un message pour une destination inconnue (i.e. il n'y a pas d'entrée correspondante pour cette destination). La table de routage doit être maintenue afin de connaître à tout moment l'état des routes pour assurer la redirection des paquets dans le cas de routes en réparation ou brisées. La mise à jour est effectuée au moyen des numéros de séquence décrits dans le paragraphe précédent (IV.3.1). Le *NSeq* est déterminé a partir des informations contenues dans le paquet de contrôle, ou sinon le champ du *NSeq* valide est mis à faux.

Adresse IP destination	*Numéro séquence destination*	*Next Hop*	*Hop count*	*Lifetime*	*Liste des Précurseurs*	*Flags*

Figure 4. 5: une entrée dans la table de routage

Chaque entrée dans la table de routage comporte essentiellement :

- ***Adresse IP de la destination :*** c'est l'adresse du nœud destination avec lequel on désire établir un chemin.
- ***Numéro de séquence destination*** : qui garantit qu'aucune boucle ne peut se former. Il permet de faire la distinction entre une ancienne information de routage se propageant dans le réseau et une nouvelle due par exemple à un changement de topologie.
- Le statut de validité du *NSeq*
- L'état de la route et autres avertissements (valide, invalide, à réparer, en cours de réparation)
- ***Next Hop*** : Adresse IP du prochain nœud en direction de la destination.
- ***Hop count*** : Le nombre de saut nécessaire pour atteindre la destination (i.e. La distance en nombre de nœuds ou de sauts).
- ***Flags*** : Indique l'état de la route. Elle peut être valide (en cours d'utilisation), invalide (le timer a expiré et elle sera bientôt supprimée) ou en cours de réparation (la route a été brisée, une réparation est en cours a fin de la rétablir).
- ***Liste des précurseurs*** (voisins actifs) : Cette liste est constituée des adresses IP des nœuds précédents qui utilisent le nœud courant pour atteindre la destination pendant un temps donné. Ils font office de "Previous Hops" et permettent, lors d'une cassure, d'informer les nœuds de sa réparation.
- ***Lifetime*** (Durée de vie d'une route) : c'est le temps d'expiration de l'entrée de la table de routage ou le temps au bout duquel l'entrée est invalidée (i.e. indique l'instant de suppression de la route). A chaque utilisation d'une entrée, son temps d'expiration est remis à jour (temps courant + active route time).
- Un tampon de requête afin qu'une seule réponse soit envoyée par requête.

A chaque fois qu'une route est utilisée pour transmettre des données, le compteur du temps de validité est réinitialisé.

Si une nouvelle route est nécessaire, ou qu'une route disparaît, la mise à jour de ces tables s'effectue par l'échange des paquets de contrôle (RREQ, RREP et RERR) [137].

Une entrée de la table de routage est mise à jour lorsque un nœud reçoit un message contenant un nouveau *NSeq*:

- Plus élevé que le *NSeq_dst* actuellement dans la table de routage, ou
- Egale au *NSeq_dst* dans la table de routage mais la route est marquée comme invalide, ou
- Egale au *NSeq_dst* dans la table de routage (TR) mais le nombre de nœuds intermédiaires (*HopCount*) déterminé à partir du paquet de contrôle est plus petit que celui actuellement est dans la TR, ou
- Le *NSeq* dans la table de routage est inconnu.

Si l'entrée de la table de routage vers une destination est créée ou mis à jour, alors les actions suivantes se produisent :

- La route est marquée comme valide,
- L'adresse du prochain nœud dans l'entrée de table de routage pour la destination est celle du nœud à partir duquel le paquet de contrôle a été reçu,
- Le nombre de nœud (*Hop Count*) est initialisé au nombre de nœuds associé à cette destination.
- La taille de préfixe est placée à la valeur du paquet de contrôle ou zéro.
- Le champ de durée de vie est initialisé au temps actuel plus la valeur *ACTIVE_ROUTE_TIMEOUT*.
- Le *NSeq_dst* est initialisé au *NSeq* contenu dans le paquet de contrôle ou à zéro pour un *NSeq* inconnu.

Chaque fois qu'une route est utilisée pour transmettre un paquet de données, le champ « durée de vie » de la source, la destination et le prochain nœud sur le chemin vers la destination est mis à jour pour qu'il ne soit pas inférieure au temps actuel plus *ACTIVE_ROUTE_TIMEOUT*. Puisque la route entre source et destination doit être symétrique, le champ durée de vie de l'entrée de table de routage pour la destination du prochain nœud sur le chemin de retour vers la source est aussi mis à jour pour qu'il ne soit pas inférieure au temps actuel plus *ACTIVE_ROUTE_TIMEOUT*.

La durée de vie d'un route active est mise à jour chaque fois que la route est utilisée indépendamment de la destination.

Noter que le champ de la durée de vie dans la table de routage joue un rôle bivalent. Pour une route active c'est le temps d'expiration, et pour une route invalide c'est le temps de sa suppression. Si un paquet de données est reçu pour une route incorrecte, le champ durée de vie est mis à jour au temps actuel plus *DELETE_PERIOD* [135].

Pour chaque entrée de table de routage pour une route valide maintenue par un nœud, ce dernier maintient aussi une liste de précurseurs qui peuvent être des nœuds véhiculant des paquets sur cette route. Ces précurseurs recevront un avertissement à partir du nœud en cas de détection de la perte d'un lien vers le prochain nœud. La liste des précurseurs dans une entrée de table de routage contient les nœuds voisins pour qui une réponse de route (RREP) a été créée ou transmise.

Les nœuds maintiennent à jour la liste de précurseurs de deux manières. Soit un nœud reçoit des données d'un de ses voisins et garde les informations de connexions. Soit si un nœud ne reçoit pas de paquet d'un autre, il peut envoyer un message HELLO à ses voisins pour le considérer toujours comme un nœud actif.

IV.3.3 Découverte de routes avec l'AODV

IV.3.3.1. Génération de Requête de Route (paquet RREQ)

En cas de besoin d'une route vers une destination et que cette dernière n'est pas disponible dans sa table de routage (Figure 4. 5), le nœud diffuse un paquet (requête) RREQ (Figure 4. 6 (a)). Cela peut se produire si la destination n'est pas connue au préalable pour ce nœud, ou si le chemin précédemment valide pour la destination a expiré ou il est marqué comme invalide ou défaillant (i.e. la métrique qui lui est associée est infinie).

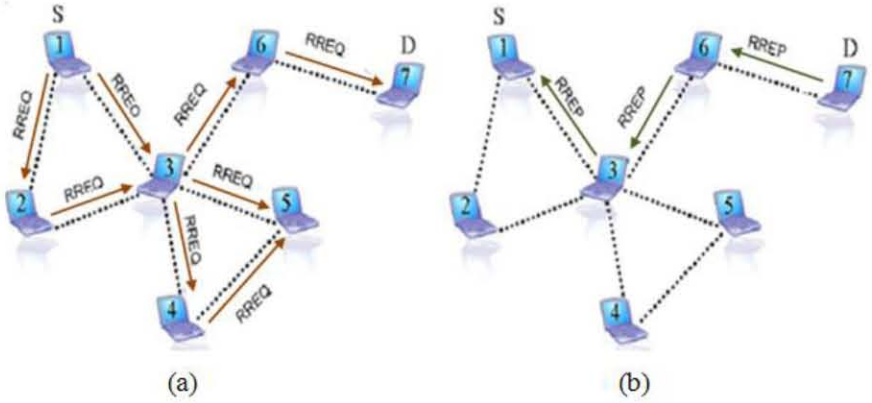


Figure 4. 6: Découverte de route (a) diffusion de RREQ et (b) réponse RREP

Le champ *NSeq_dst* du paquet RREQ, contient le dernier *NSeq* connu pour cette destination. Cette valeur est recopiée à partir du champ numéro séquence destination de la table de routage. Si le *NSeq* n'est pas connu, la valeur nulle sera prise par défaut. Le *NSeq_src* du paquet RREQ contient la valeur du *NSeq* du nœud source.

Avant de diffuser le paquet RREQ, le nœud source incrémente le NSeq de la source, initialise le champ *Hop Count* par zéro et sauvegarde en mémoire tampon l'identifiant du paquet (RREQ_ID) et l'adresse IP source (@ source) pour une durée *PATH_DISCOVERY_TIME*. De cette façon, lorsque le nœud reçoit de nouveau le même paquet de ses voisins (i.e. le cas où un voisin le lui renverrait), il va l'ignorer (i.e. pour ne pas traiter ou retransmettre le paquet une autre fois).

Chaque nœud maintient un seul RREQ_ID. La paire <@source, RREQ_ID > identifie de manière unique un paquet RREQ. Le champ RREQ_ID est incrémenté lors de chaque transmission du paquet RREQ.

Quand un nœud intermédiaire (de transit) retransmet la requête RREQ à un voisin, il sauvegarde l'identificateur du nœud à partir duquel la première copie de la requête est reçue. Cette information est utilisée pour construire le chemin de retour (Figure 4. 6 (b)), qui sera traversé par le paquet *réponse de route* (RREP) de manière unicast. Puisque le paquet *réponse de route* va être envoyé à la source, les nœuds appartenant au chemin de retour vont modifier leurs tables de routage suivant le chemin contenu dans le paquet RREP (temps d'expiration, *NSeq* et prochain saut).

Un nœud source compte souvent avoir des transmissions bidirectionnelles avec un nœud destination. Dans de tels cas, il est impératif que le nœud source ait une route vers la destination et que cette dernière ait également une route vers la source.

Pour que cela doit se produire de la façon la plus efficace que possible, n'importe quelle génération d'un paquet RREP par un nœud intermédiaire (voir section: Génération d'une réponse (RREP)) pour une transmission vers le nœud source doit être accompagnée par une certaine action qui informe la destination au sujet d'une route de retour vers le nœud source.

Le nœud source choisit ce mode de fonctionnement pour les nœuds intermédiaires en positionnant le champ 'G'. Voir la section : Génération d'une réponse RREP gratuit (gratuitous) pour plus de détails sur les mesures prises par un nœud intermédiaire en réponse à un paquet RREQ avec le champ 'G'.
Un nœud génère au plus *RREQ_RATELIMIT* paquets RREQ par seconde. Après la diffusion d'un RREQ, le nœud se met en attente d'un RREP (ou d'autres messages de contrôle avec l'information à jour sur la route vers la destination appropriée). Si aucune route n'est reçue dans une durée de *NET_TRAVERSAL_TIME* millisecondes, le nœud peut essayer à nouveau l'opération de découverte de route (une autre tentative) en diffusant un autre paquet RREQ. Le nombre de tentatives est au maximum *RREQ_RETRIES* fois égale à une valeur maximale pour TTL. Chaque nouvelle tentative doit incrémenter et mettre à jour le champ *RREQ_ID*.
Pour chaque tentative, le champ TTL de l'en-tête IP est définit selon le mécanisme décrit dans la section: Contrôle de la diffusion du paquet RREQ, afin de savoir à quelle distance le paquet RREQ est diffusé pour chaque tentative.
Les paquets de données en attente de route (i.e. en attente d'un RREP en réponse à un RREQ déjà envoyé) doivent êtres mis dans une mémoire tampon selon la politique FIFO. Si au bout de *RREQ_RETRIES* (pour une valeur TTL maximale) tentatives de découverte de routes sans recevoir de réponse (paquet RREP), tous ces paquets de données pour la destination correspondante devront être supprimés de la mémoire tampon et un message « Destination inaccessible » devra être adressé à la couche application.

Pour réduire le phénomène de congestion dans le réseau du aux tentatives répétées par le nœud source durant la phase de découverte de route pour une seule destination, l'AODV doit utiliser un Backoff exponentiel binaire. La première fois qu'un nœud source diffuse une requête RREQ, il se met en attente pour une durée de *NET_TRAVERSAL_TIME* milliseconde pour recevoir un paquet RREP.

Si aucun paquet RREP n'est reçu dans ce délai, le nœud source envoie une nouvelle requête RREQ, et par conséquent, le temps d'attente pour le RREP correspondant au deuxième RREQ est de :

*2 * NET_TRAVERSAL_TIME* millisecondes (i.e. utilisation du mécanisme de Backoff exponentiel binaire par le nœud source). Si aucun RREP n'est reçu dans ce nouveau délai, une autre requête RREQ peut être envoyée pour *RREQ_RETRIES* tentatives supplémentaires après la première requête RREQ. Pour chaque nouvelle tentative, le temps d'attente pour les RREP est multiplié par 2, de sorte que le temps est conforme à un Backoff exponentiel binaire.

IV.3.3.2 Contrôle de la diffusion du paquet RREQ

Pour empêcher la diffusion inutile des paquets RREQ à l'ensemble du réseau (limiter le coût dans le réseau), le nœud source propose d'étendre la recherche progressivement. Au départ, le nœud source utilise un *TTL = TTL_START* (un nombre de sauts limité) dans l'en-tête d'IP du paquet RREQ et fixe le délai d'attente ou d'expiration (timeout) pour recevoir un paquet RREP à *RING_TRAVERSAL_TIME* millisecondes.

RING_TRAVERSAL_TIME est calculé comme décrit dans la section 4.4 (Paramètres de configuration de l'AODV). Le *TTL_VALUE* utilisé dans le calcul de *RING_TRAVERSAL_TIME* est égale à la valeur du champ TTL dans l'en-tête IP.

Si aucune réponse RREP n'est reçue pour la requête RREQ pour le délai fixé, la source diffuse une nouvelle requête RREQ avec un *TTL* incrémenté par *TTL_INCREMENT* (augmenter le nombre de sauts). Ce processus (rediffusion) se répète jusqu'à ce que le *TTL* défini dans le paquet RREQ atteigne *TTL_THRESHOLD*, au-delà duquel *TTL = NET_DIAMETER* est utilisé pour chaque tentative. Dans tous les cas, le délai d'attente pour la réception de RREP est *RING_TRAVERSAL_TIME*. Lorsqu'on désire avoir toutes les tentatives traversées l'ensemble du réseau ad hoc, il suffit d'affecter à *TTL_START* et *TTL_INCREMENT* tous les deux la même valeur que *NET_DIAMETER*.

Le champ nombre de sauts (*Hop Count*) stocké dans une entrée invalide de table de routage indique la dernière information sur le nombre de nœuds intermédiaires pour atteindre la destination (connu pour cette destination dans la table de routage). Quand une nouvelle route vers la même destination est nécessaire à une date ultérieure (par exemple, en cas de perte de route), le TTL dans l'en-tête IP de RREQ est initialement fixé au nombre de sauts (*Hop Count*) plus *TTL_INCREMENT*.

Ensuite, après chaque expiration, TTL est incrémenté par *TTL_INCREMENT* jusqu'à ce que *TTL = TTL_THRESHOLD* soit atteint. Au-delà ce *TTL = NET_DIAMETER* est utilisé.

Une fois *TTL = NET_DIAMETER*, le délai d'attente pour les paquets RREP est fixé à *NET_TRAVERSAL_TIME* comme spécifié dans la section 4.3.3.1 (Génération de Requête de Route).

Chaque fois qu'une entrée dans la table de routage expire, elle ne devrait pas être effacée avant *(current_time + DELETE_PERIOD)* (voir la section 4.3.5 : Paquet d'erreur de route, Échéance et suppression de route). Sinon, l'état soft correspondant à la route (par exemple, le dernier nombre de sauts connu) sera détruit.
N'importe qu'elle entrée dans la table de routage en attente d'une réponse RREP, ne devrait pas être effacée avant (*current_time + 2 * NET_TRAVERSAL_TIME*).

IV.3.3.3 Traitement et acheminement d'un paquet RREQ

Quand un nœud reçoit une requête RREQ, il crée d'abord ou met à jour une route vers le nœud précédent avec *NSeq* invalide (voir section 4.3.2 : Gestion de la Table de routage et les listes de précurseurs), ensuite, il vérifie s'il a reçu un paquet RREQ avec le même couple <adresse IP source, RREQ_ID> dans au moins le dernier *PATH_DISCOVERY_TIME*. Si c'est le cas, le nœud ignore ce nouveau paquet RREQ reçu.

Si les requêtes RREQ ne sont pas rejetées, les mesures prises sont :

D'abord, le nœud ajoute la valeur « 1 » à la valeur du champ nombre de sauts (*Hop count*) dans le paquet RREQ pour mémoriser ce nouveau saut. Ensuite, le nœud cherche une route de retour vers la source d'adresse IP (voir section: Gestion de la Table de routage et les listes de précurseurs). En cas de besoin, la route est créée, ou mis à jour en utilisant le *NSeq_src* de la requête RREQ dans sa table de routage. Ce chemin de retour sera nécessaire si le nœud reçoit un paquet RREP du nœud source qui a initié la requête RREQ (identifié par l'adresse IP source).

Lorsque le chemin de retour est crée ou mis à jour, les actions suivantes sont également effectuées:

1) copié le *NSeq_src* de la requête RREQ dans la l'entrée de table de routage s'il est supérieur au *NSeq_dst* de l'entrée correspondante dans la table de routage ;
2) Le champ « *NSeq valide* » est fixé à vrai ;
3) Le prochain nœud dans la table de routage devient le nœud à partir duquel la requête RREQ a été reçue (il est obtenu à partir de adresse IP source dans l'en-tête d'IP et elle n'est pas souvent égal au champ d'adresse IP source dans le paquet RREQ) ;
4) Le nombre de nœuds intermédiaires (*Hop count*) est copié à partir du champ correspondant dans le paquet RREQ.

Chaque fois qu'un paquet RREQ est reçu, la durée de vie de l'entrée du chemin de retour pour l'adresse IP source doit être le maximum entre *ExistingLifetime* et *MinimalLifetime*, où

*MinimalLifeTime = (CurrentTime + 2 * NetTraversalTime – 2 * HopCount * NodeTraversalTime)*

Le nœud courant peut utiliser le chemin de retour pour transmettre les paquets de données de la même manière que pour toutes autres routes dans la table de routage.

Si un nœud ne génère pas un paquet RREP (suivant les règles de traitement dans la section : Génération d'une réponse RREP) après la réception d'un paquet RREQ, et le champ TTL inclus dans l'entête IP du paquet entrant (i.e. qui arrive) est supérieur a un, le nœud met à jour le paquet RREQ et le diffuse.

Pour mettre à jour le paquet RREQ, le champ TTL inclus dans l'entête IP du paquet sortant est décrémenté de "1", et le champ nombre de nœuds intermédiaires de RREQ est incrémenté de "1". Le *NSeq_dst* pour la destination demandée est le maximum entre la valeur correspondante dans le paquet RREQ reçu, et la valeur *NSeq_dst* actuellement maintenu par le nœud pour la destination demandée (dans la table de routage).

Cependant, le nœud intermédiaire ne doit pas modifier sa valeur de *NSeq_dst*, même si la valeur reçue dans le paquet RREQ entrant est plus grande que la valeur actuelle (i.e. maintenu par le nœud de routage).

Autrement, si un nœud ne génère pas un RREP, alors il rejette la requête RREQ. Notez que, si les nœuds intermédiaires répondent à chaque transmission des paquets RREQ pour une destination particulière, il en résulte, que la destination ne reçoit aucunes demandes de découverte de route (RREQ).

Dans cette situation, la destination ne peut pas avoir de route pour le nœud source (le nœud qui a généré la demande RREQ). Ceci pourrait inciter la destination à lancer une découverte de route (par exemple : dans le cas ou le nœud source tente d'établir une connexion TCP).

Pour que le nœud destination détermine des routes vers le nœud source, ce dernier doit initialiser l'indicateur G (Gratuitous RREP Flag) dans le paquet RREQ à la valeur vrai si pour n'importe quelle raison, la destination est dans le besoin d'avoir une route vers le nœud source.

Si en réponse à un paquet RREQ avec l'indicateur 'G' initialisé, le nœud intermédiaire retourne un paquet RREP, il faut aussi transmettre en mode unicast un paquet gratuitous RREP (RREP Gratuit) pour la destination (voir section: Génération d'une réponse RREP gratuit).

IV.3.3.4 Génération d'une réponse (RREP)

Un nœud génère un RREP si:

(i) il est lui-même la destination, ou

(ii) il a un chemin actif vers la destination ; le *NSeq_dst* dans l'entrée de table de routage est valide et supérieur ou égal au *NSeq_dst* dans la requête RREQ, et l'indicateur "D" n'est pas défini.

Quand un nœud génère un paquet RREP, il copie l'adresse IP destination et le *NSeq_src* du paquet RREQ dans les champs correspondants dans le paquet RREP. Le traitement est légèrement différent, selon que le nœud est lui-même la destination demandée (voir la section : Génération d'une réponse (RREP) par la destination), ou c'est un nœud intermédiaire avec une route assez fraiche vers la destination (voir section: Génération d'une réponse (RREP) par un nœud intermédiaire).

Une fois le paquet RREP est créé, il est routé en mode unicast au prochain nœud vers la source de la requête RREQ, comme indiqué dans l'entrée de table de routage pour cette source.

Comme le RREP est transmis sur le chemin de retour vers le nœud source ayant initié la requête RREQ, le champ nombre de sauts « *Hop Count*» inclus dans la réponse est incrémenté de un à chaque saut (i.e. chaque fois qu'un nœud a reçu la réponse). Ainsi, lorsque le paquet RREP atteint le nœud source, le champ nombre de sauts représente la distance, en nombre de sauts (nœuds) entre la source et la destination.

IV.3.3.4.1 Génération d'une réponse (RREP) par la destination

Si le nœud ayant généré le paquet RREP est la destination elle-même, il doit incrémenter son propre *NSeq* de un si le *NSeq* dans le paquet RREQ est égal à cette valeur incrémentée. Sinon, la destination ne change pas son *NSeq* avant de générer le paquet RREP.

Le nœud destination insère son *NSeq* (peut-être nouvellement incrémenté) dans le champ *NSeq_dst* du paquet RREP, et initialise à la valeur zéro le champ nombre de sauts « *Hop Count*» dans le paquet RREP.

Le nœud destinataire copie la valeur *MY_ROUTE_TIMEOUT* (voir section 4.4 : Paramètres de configuration de l'AODV) dans le champ de durée de vie (Life Time) du paquet RREP. Chaque nœud peut réinitialiser (reconfigurer) ses valeurs pour *MY_ROUTE_TIMEOUT*, compte tenu de légères contraintes (voir section 4.4 : Paramètres de configuration de l'AODV).

IV.3.3.4.2 Génération d'une réponse (RREP) par un nœud intermédiaire

Si le nœud ayant généré le paquet RREP n'est pas la destination, mais un nœud intermédiaire sur le chemin entre la source et la destination, il recopie son *NSeq_dst* dans le champ *NSeq_dst* dans le paquet RREP.

Le nœud intermédiaire met à jour son entrée de table de routage en plaçant le dernier nœud (à partir duquel il a reçu la requête RREQ, comme indiqué dans le champ d'adresse IP source dans l'en-tête IP) dans la liste des précurseurs pour l'entrée dans la table de routage (i.e. l'entrée pour l'adresse IP destination).

Comme, il met à jour également son entrée de table de routage pour le nœud source qui a généré le paquet RREQ en plaçant le prochain nœud vers la destination dans la liste de précurseurs pour l'entrée du chemin de retour (i.e. l'entrée pour le champ adresse IP source du paquet RREQ).

Le nœud d'intermédiaire recopie sa distance en nombre de sauts de la destination (indiqué par le nombre de sauts dans la table de routage) dans le champ « *Hop Count* » du paquet RREP. Le champ « *Durée de vie* » de la requête RREP est calculé par soustraction du temps courant et le temps d'expiration de son entrée de table de routage.

IV.3.3.4.3 Génération d'une réponse RREP gratuit (gratuitous)

Après qu'un nœud reçoit un paquet RREQ et répond avec un paquet RREP, il élimine le paquet RREQ. Si le champ « G » du paquet RREQ est définit, et le nœud intermédiaire retourne un paquet RREP au nœud source, il doit aussi retourner en mode unicast un RREP gratuit (gratuitous RREP) à la destination. Le RREP gratuit contient les valeurs suivantes dans les champs du paquet RREP (Tableau 4.5):

Hop Count	Nombre de nœuds qu'il faut traverser pour atteindre le nœud source comme indiqué dans l'entrée de table de routage de la source.
Destination IP Address	L'adresse IP du nœud source qui a généré le paquet RREQ.
Destination Sequence Number	Le *NSeq_src* dans le paquet RREQ.
Originator IP Address	L'adresse IP du nœud destination dans le paquet RREQ.
Lifetime	Le temps restant de la durée de vie du chemin vers l'émetteur du paquet RREQ comme indiqué dans la table de routage du nœud intermédiaire.

Tableau 4. 5: description des champs d'un paquet RREP gratuit

Le RREP gratuit est ensuite envoyé vers le prochain nœud sur le chemin vers la destination, juste comme si le nœud destination avait déjà émis un paquet RREQ pour le nœud source et ce paquet RREP a été généré en réponse à ce paquet fictif RREQ. Le paquet RREP qui est envoyé à l'émetteur du paquet RREQ est identique avec le RREP gratuit que ce soit ou non le champ 'G' est définit.

IV.3.3.5 Réception et acheminement d'une réponse RREP

Quand un nœud reçoit un paquet RREP, il recherche une route vers le nœud précédent. En cas de besoin, une route est créée pour le nœud précédent, mais sans un *NSeq* valide (voir section 4.3.2 : Gestion de la Table de routage et les listes de précurseurs). Le nœud incrémente la valeur « *Hop Count* » de un dans le paquet RREP. La valeur incrémentée est nommée *" New Hop Count "* (Nouveau nombre de sauts).

Ensuite, une route vers cette destination est créée si elle n'existe pas auparavant. Autrement, le nœud compare le *NSeq_dst* dans le paquet à son propre *NSeq* pour l'adresse IP destination dans le paquet RREP.

Lors de la comparaison, l'entrée existante est mise à jour seulement dans les circonstances suivantes :

(i) le *NSeq* est marqué invalide pour l'entrée de la table de routage, ou

(ii) le *NSeq_dst* dans le paquet RREP est supérieure au *NSeq_dst* copié dans le nœud et la valeur connue est valide, ou

(iii) les numéros de séquence sont les mêmes, mais le chemin est marqué comme inactif, ou

(iv) les numéros de séquence sont les mêmes, et le nouveau nombre de saut dans le paquet est plus petit que le nombre de sauts dans l'entrée de la table de routage.

Si l'entrée de la table de routage vers la destination est créée ou mis à jour, les actions suivantes se produisent:

- la route est marquée comme active,
- le *NSeq_dst* est marqué comme valide,
- le prochain nœud dans l'entrée de la table de routage est le nœud à partir duquel le paquet RREP est reçu, ce qui est indiqué dans le champ adresse IP source dans l'en-tête IP,
- le champ nombre de sauts (*Hop Count*) est fixé à la valeur *"New Hop Count*",
- le délai d'expiration est fixé au temps courant plus la valeur de la durée de vie dans le paquet RREP, et
- le *NSeq_dst* est le *NSeq_dst* dans le paquet RREP.

Le nœud actuel peut ensuite utiliser cette route pour transmettre des paquets de données vers la destination. Si le nœud actuel n'est pas le nœud indiqué par l'adresse IP source dans le paquet RREP et la route pour le transfert a été créée ou mise à jour comme décrit ci-dessus, le nœud consulte son entrée de la table de routage du nœud source pour déterminer le prochain nœud pour router le paquet RREP vers la source en utilisant les informations dans l'entrée de la table de routage.

Si un nœud qui envoi un paquet RREP sur un lien susceptible d'avoir des erreurs ou qu'il soit unidirectionnel, il doit définir l'indicateur 'A' pour exiger au récepteur du paquet RREP à retourner un acquittement sous forme d'un paquet RREP-ACK (voir la section I.V.3.3.6 : Opération sur des liens unidirectionnels).

Quand n'importe quel nœud transmet un paquet RREP, la liste des précurseurs pour le nœud destination est mise à jour en y ajoutant le prochain nœud a partir duquel le paquet RREP est transmis.

Pour chaque nœud du chemin de retour utilisé pour transmettre un paquet RREP, sa durée de vie est changée pour être le maximum de (*la durée de vie existante, (le temps courant + ACTIVE_ROUTE_TIMEOUT*)).

Finalement, la liste des précurseurs pour le prochain nœud vers la destination est mise à jour pour contenir le prochain saut vers la source.

I.V.3.3.6. Opération sur des liens unidirectionnels

Il est possible que la transmission de RREP échoue, surtout si la transmission de RREQ déclenchant le RREP se produit sur une liaison unidirectionnelle. Si aucun autre RREP généré pour la même de découverte de route (RREQ) tente d'atteindre le nœud émetteur du paquet RREQ, la source relance une nouvelle tentative de découverte de route après un délai (voir section 4.3.3.1 : Génération de Requête de Route RREQ). Toutefois, le même scénario pourrait bien se répéter sans aucune amélioration, et aucune route n'est découverte après plusieurs tentatives. Si des mesures correctives sont prises, cela peut se produire même lorsque les routes bidirectionnelles entre source et destination existent. En utilisant la transmission par diffusion des RREQ, la couche liaison ne sera pas en mesure de détecter la présence de tels liens unidirectionnels. Dans l'AODV, n'importe quel nœud agit seulement sur le premier RREQ avec le même RREQ_ID et ignore tous les RREQs ultérieures. Supposons, par exemple, que le premier RREQ arrive le long d'un chemin qui a un ou plusieurs liens unidirectionnel(s). Une série de RREQ peut arriver par une liaison bidirectionnelle (en supposant que ces liaisons existent), mais il sera ignoré.

Pour éviter ce problème, quand un nœud détecte que la transmission d'un paquet RREP a échoué, il enregistre le prochain nœud de l'échec de RREP dans une "liste noire". Ces échecs peuvent être détectés par l'absence de la couche liaison ou des informations de la couche réseau (par exemple, RREP-ACK). Un nœud ignore tous les RREQs reçus de n'importe quels nœuds de sa liste noire. Les nœuds sont retirés de la liste noire après une période *BLACKLIST_TIMEOUT* (voir section 4.4 : Paramètres de configuration de l'AODV). Cette période devrait être fixée à la limite supérieure du temps nécessaire pour effectuer le nombre autorisé de tentatives de découverte de routes tel que décrit à la section 4.3.3.1 (Génération de Requête de Route RREQ).

Le temps au quel le *RREP-ACK* est reçu viendra probablement juste après le moment où le RREP a été envoyé avec le bit «A». Cette information devrait être suffisante pour fournir une assurance à l'émetteur de RREP sur un lien actuellement bidirectionnel, sans aucune dépendance réelle sur le paquet RREP particulier pour être reconnu.

IV.3.3.7 Le paquet HELLO

Un nœud peut offrir les informations de connectivité en diffusion des paquets HELLO. Seuls les nœuds faisant partie d'une route active (i.e. une route où il existe des échanges de messages) qui devront utiliser ce type de paquet (Hello).

A chaque *HELLO_INTERVAL* millisecondes, le nœud vérifie s'il a diffusé un paquet (par exemple, un paquet RREQ ou un message approprié de la couche 2) dans le dernier *HELLO_INTERVAL*. Si ce n'est pas le cas, il peut diffuser un paquet RREP avec un *TTL* = 1, appelé HELLO, avec les champs de RREP définis comme suit (Tableau 4.6):

Destination IP Address	l'adresse du nœud destination
Destination Sequence Number	Le dernier *NSeq* du nœud.
Hop Count	égale à zéro
Lifetime	*ALLOWED_HELLO_LOSS * HELLO_INTERVAL*

Tableau 4. 6: Valeurs des champs d'un paquet Hello

Un nœud peut déterminer la connectivité en écoutant les paquets provenant de l'ensemble de ses voisins. Si, dans la période *DELETE_PERIOD* passée, il a reçu un message Hello d'un voisin, et ensuite, le voisin n'a reçu aucun paquets (messages Hello ou autres) pour plus de *ALLOWED_HELLO_LOSS * HELLO_INTERVAL* millisecondes, le nœud devrait supposer que le lien vers ce voisin est actuellement perdu. Quand cela se produit, le nœud devrait procéder comme dans la section : Paquet d'erreur de route, Échéance et suppression de route (envoi d'un paquet d'erreur RERR et la route devient inactive).

Chaque fois qu'un nœud reçoit un paquet Hello d'un voisin, le nœud doit s'assurer qu'il dispose d'un chemin actif vers ce voisin, ou il crée un, en cas de besoin. Si une route existe déjà, alors la durée de vie pour la route devrait être augmentée pour être au moins *ALLOWED_HELLO_LOSS * HELLO_INTERVAL*. La route vers

le voisin, si elle existe, doit contenir le dernier *NSeq_dst* du paquet Hello que le nœud actuel peut maintenant commencer à l'utiliser pour transmettre des paquets de données. Les liaisons qui sont créées par les paquets Hello et qui ne sont pas utilisées par les autres routes actives ont des listes de précurseurs vide.

IV.3.4 Maintenance des routes avec l'AODV

Dès qu'une route est établie entre un nœud source et une destination, un mécanisme de maintenance est déclenché automatiquement. Ce mécanisme se base essentiellement sur la gestion de la connectivité c'est à dire comment détecter une défaillance et comment y remédier.

AODV maintient les routes aussi longtemps que celles-ci sont actives. Une route est considérée active tant que des paquets des données transitent périodiquement de la source à la destination sur ce chemin. Lorsque la source stoppe l'émission des paquets des données, le lien expirera et sera effacé des tables de routage des nœuds intermédiaires. Si un lien se rompt sur une route active, il est considéré défaillant.

Afin de détecter cette défaillance, l'AODV utilise les paquets de contrôle « HELLO » qui permettent de vérifier la connectivité ou plutôt l'activité des routes. Ce paquet n'est autre qu'un RREP contenant l'adresse de l'émetteur avec un TTL égal à 1 pour éviter qu'il ne soit propagé plus loin dans le réseau [138].

La vérification se fait par la diffusion du message "HELLO". La période de diffusion étant fixée à une durée de "*HELLO_INTERVAL*" (en ms). Si trois messages « HELLO » ne sont pas reçus consécutivement à partir d'un nœud voisin, le lien en question est considéré défaillant. Le mouvement des nœuds qui ne participent pas dans le chemin actif, n'affectent pas la consistance des données de routage.

L'AODV maintient les adresses des voisins à travers lesquels les paquets destinés à un certain nœud arrivent. Un voisin est considéré actif, pour une destination donnée, s'il délivre au moins un paquet de données sans dépasser une certaine période (appelée *ACTIVE_TIMEOUT_PERIOD*).

Dans le cas de défaillance de liens, toutes les entrées des tables de routage faisant partie du chemin actif et qui sont concernées par la défaillance seront supprimées. Cela est accompli par la diffusion d'un message d'erreur entre les nœuds actifs [135].

Les défaillances des liens sont généralement dues à la mobilité des nœuds [139]:

- si c'est le nœud source qui se déplace et rompt la liaison avec son successeur, alors il relancera la procédure d'établissement de routes s'il en a encore besoin.
- si le nœud qui s'est déplacé est un nœud intermédiaire ou la destination, alors le nœud source doit être informé par le message RERR qui doit être généré par le nœud le plus proche parmi les deux (Figure 4. 7). L'initiateur du RERR va lister ses nœuds précurseurs sur la route défaillante en leur envoyant le paquet RERR. En recevant RERR, un nœud marque la route vers cette destination (dont l'adresse figure dans le RERR) invalide en met la valeur du champ distance correspondant à l'infini (Distance = infinie), et à son tour renvoie le RERR vers ses précurseurs sur cette route. Lorsque le nœud source reçoit le RERR, il entame alors un nouveau processus de découverte d'une nouvelle route s'il en a encore besoin.

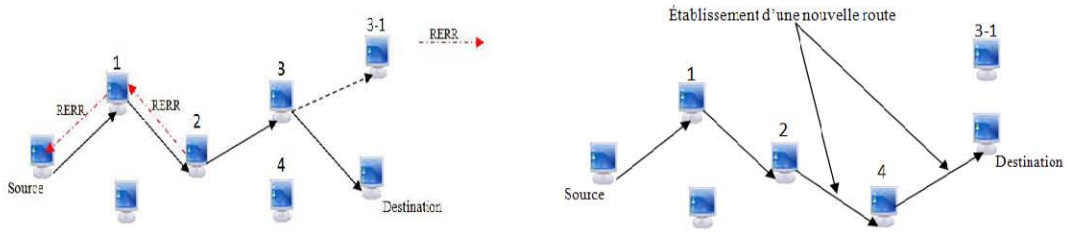


Figure 4. 7: génération de RERR à cause de défaillance du nœud 3

Dans ce qui suit, nous proposons une algorithmique de la procédure de maintenance de routes pour un nœud quelconque « N ».

Par diffusion périodique d'un paquet spécial HELLO, aux voisins directs, que chacun d'eux doit répondre.

Si un voisin qui était auparavant actif ne répond pas, soit qu'il est hors portée radio ou qu'il est déconnecté.

Cette Information est utilisée pour purger les routes qui ne marchent plus. Pour chaque destination possible, le nœud « N » garde en mémoire l'identificateur des nœuds qui lui ont communiqué récemment un paquet à router vers cette destination. Ce sont les voisins actifs de « N » pour cette destination.

« N » maintient une table de routage indexée par destination, contenant

- le nœud voisin à utiliser pour atteindre la destination,
- le nombre de sauts jusqu'à la destination,
- le *NSeq_dst* le plus récent,
- la liste des voisins actifs pour la destination.

Lorsqu'un voisin devient inaccessible en « N » :

- inspection dans la table de routage des destinations dont les routes passent par le voisin disparu, pour purger les entrées correspondantes dans la table de routage.
- récupération des voisins actifs pour ces destinations, et leur indiquer que leur route via « N » est maintenant invalide et doit être purgée de leur table.
- les voisins actifs vont aviser leurs voisins actifs respectifs, et ainsi de suite de proche en proche, et ainsi toutes les routes qui dépendaient du nœud disparu sont purgées des tables de routage.

En fait, lors de la rupture d'un lien d'une route active, l'AODV tente de faire une réparation localement en diffusant une requête de recherche de route dans le voisinage. Si cette tentative échoue, alors la route est purgée comme expliqué précédemment, et une nouvelle découverte de route est lancée par la source.

IV.3.5 Paquet d'erreur de route (RERR) : (Échéance de route et suppression de route)

Le paquet d'erreur de route (RRER) permet de prévenir l'ensemble du réseau qu'une route est brisée et qu'elle doit être supprimée des tables de routage des nœuds faisant partie de cette route.

Généralement, les erreurs de routes et les traitements de rupture de liens requièrent les étapes suivantes :

- Annulations des routes existantes.
- lister les nœuds destinations touchés par ces erreurs.
- déterminer les éventuels voisins touchés par ces erreurs.
- Livraison d'un paquet RERR approprié à ces voisins.

Le paquet RERR peut être envoyé en diffusion (s'il y a beaucoup de précurseurs), en unicast (s'il y a un seul précurseur), ou transmis de façon itérative en unicast à tous les précurseurs (si la diffusion est inadéquate). Même lorsque le paquet RERR est itérativement transmis en unicast à plusieurs précurseurs, on le considère comme un paquet de contrôle simple. Avec cela, un nœud ne devrait pas générer plus de *RERR_RATELIMIT* paquets RERR par seconde.

Un nœud initie le traitement pour un paquet RERR dans trois situations :

(i) s'il détecte une rupture de lien pour le prochain nœud pour un chemin actif dans sa table de routage lors de la transmission de données (et la réparation de route, si elle est lancée, alors elle est sans succès), ou

(ii) s'il reçoit un paquet de données destiné à un nœud pour lequel il n'a pas de chemin actif et qu'il n'a pas réparé (en cas d'utilisation de la réparation locale), ou

(iii) s'il reçoit un paquet RERR d'un voisin pour une ou plusieurs routes actives.

Pour le cas (i), le nœud établie d'abord une liste de destinations inaccessibles composée de voisins inaccessibles et de toutes les destinations supplémentaires (ou sous-réseaux) dans la table de routage locale qui utilise le voisin inaccessible comme prochain nœud. Dans ce cas, si un chemin de sous-réseau se trouve de nouveau inaccessible, une adresse IP destination pour le sous-réseau est construite en ajoutant des zéros au préfixe de sous-réseau comme indiqué dans l'entrée de la table de routage. Ceci n'est pas ambigu, puisque le précurseur est connu pour avoir l'information de la table de routage avec une longueur de préfixe compatible pour ce sous-réseau.

Pour le cas (ii), il n'y a qu'une seule destination inaccessible, qui est la destination du paquet de données qui ne peut pas être livré.

Pour le cas (iii), la liste devrait comporter les destinations dans le paquet RERR pour lesquelles il existe une entrée correspondante dans la table de routage locale qui a le prochain nœud comme émetteur du paquet RERR reçu.

Une partie des destinations inaccessibles dans la liste pourraient être utilisées par les nœuds voisins qui doivent envoyer un nouveau paquet RERR. Ce paquet doit contenir les destinations qui font partie de la liste créée pour les destinations inaccessibles qui ont une liste des précurseurs non vide.

Le ou les nœud(s) voisin(s) qui reçoivent le paquet RERR sont tous ceux qui appartiennent à une liste de précurseurs d'au moins l'une des destinations inaccessibles dans le nouveau paquet RERR créé. Le paquet RERR doit être transmis en unicast dans le cas où il n'y a qu'un seul voisin pour le recevoir. Sinon, ce paquet

sera diffusé à l'adresse locale (Destination IP =255.255.255.255, TTL =1) avec les destinations inaccessibles, et leurs numéros de séquence destinations, inclus dans le paquet. Le champ nombre de destination « *DestCount* » du paquet RERR indique le nombre de destinations inaccessibles inclus dans ce paquet.

Juste avant de transmettre le paquet RERR, certaines mises à jour sont faites sur la table de routage qui pourront avoir un impact sur les numéros de séquence destination pour les destinations inaccessibles. Pour chacune de ces destinations, l'entrée de la table de routage correspondante est mise à jour comme suit:

1. Le *NSeq_dst* de cette entrée de la table de routage, s'il existe et s'il est valide, il est incrémenté dans les cas (i) et (ii) ci-dessus, et copié à partir du paquet RERR entrant dans le cas (iii) ci-dessus.
2. L'entrée est invalidée en marquant l'entrée de la route invalide
3. Le champ Durée de vie est mis à jour à la valeur du temps courant plus *DELETE_PERIOD*.

NB : Avant ce temps, l'entrée ne devrait pas être supprimée.

Notons que le champ *Durée de vie* dans la table de routage joue un double rôle : il est le temps d'expiration pour une route active, et il est le temps de suppression pour une route invalide. Si un paquet de données est reçu via une route invalide, le champ Durée de vie est mis à jour au temps courant plus *DELETE_PERIOD*. La détermination de *DELETE_PERIOD* est discutée à la section : Paramètres de configuration de l'AODV.

IV.3.6 La réparation locale

Lorsqu'une rupture de lien sur une route active se produit, le nœud en amont de cette rupture peut choisir entre réparer le lien au niveau local si la destination n'est pas plus loin que *MAX_REPAIR_TTL* sauts.

Pour réparer le lien cassé, le nœud incrémente le *NSeq* pour la destination, puis diffuse un RREQ pour cette destination. Le TTL (durée de vie) de RREQ devrait être initialement fixé à la valeur suivante : *Max (MIN_REPAIR_TTL, 0.5 * #hops) + LOCAL_ADD_TTL*, où #hops est le nombre de sauts vers l'émetteur du paquet non actuellement délivré. Ainsi, les tentatives de réparation locale sont souvent invisibles pour le nœud source, et ont toujours un *TTL > = MIN_REPAIR_TTL + LOCAL_ADD_TTL*.

Le nœud ayant initié la réparation, attend une « *période de découverte* » pour recevoir les paquets RREPs en réponse à la requête RREQ.

Durant la réparation locale, les paquets de données doivent être sauvegardés (mis en mémoire tampon). Si, à la fin de la période de découverte, le nœud chargé de la réparation n'a pas reçu un paquet RREP (ou d'autres paquets de contrôle de création ou de mise à jour de la route) pour cette destination, il transmet un paquet RERR pour cette destination.

D'autre part, si le nœud reçoit un ou plusieurs paquets RREPs (ou d'autres messages de contrôle de création ou de mise à jour de la route vers la destination souhaitée) au cours de la période de découverte, il compare d'abord le nombre de sauts de la nouvelle route avec la valeur dans le champ nombre de sauts de l'entrée de table de routage invalide pour cette destination. Si le nombre de sauts de la route nouvellement déterminée pour la destination est plus grand que le nombre de sauts de la route trouvée précédemment, le nœud doit délivrer un message RERR pour la destination avec le bit 'N' définit. Puis il procède comme décrit à la section 4.3.3.5 (Réception et acheminement d'une réponse RREP), en mettant à jour les entrées de sa table de routage pour cette destination.

Un nœud qui reçoit un message RERR avec l'indicateur 'N' définit, ne doit pas supprimer la route vers cette destination. La seule action à prendre, doit être la retransmission du message, si le paquet RERR est reçu à partir du prochain nœud sur cette route, et s'il y a un ou plusieurs nœuds précurseurs pour cette route jusqu'à la destination. Lorsque le nœud source reçoit un paquet RERR avec l'indicateur 'N' définit, si ce message arrive de son prochain nœud le long de sa route vers la destination, alors le nœud source peut choisir entre relancer la découverte de route, tel que décrit à la section 4.3.3.1 (Génération de Requête de Route).

La réparation locale de liens rompus pour certaines routes aboutit parfois à augmenter la longueur des chemins pour ces destinations. Réparer le lien localement est susceptible d'augmenter le nombre de paquets de données qui doivent être livrés à leurs destinations, puisque les paquets de données ne seront pas abandonnés quant le paquet RERR se dirige vers le nœud source. Envoyer un paquet RERR au nœud source après une réparation local du lien cassé peut permettre à la source de trouver une nouvelle meilleure route vers la destination, sur la base des positions du nœud courant.

Quand un lien se brise le long d'une route active, il y a souvent plusieurs destinations qui deviennent inaccessibles. Le nœud situé en amont du lien cassé tente une réparation locale immédiate pour seulement la destination vers laquelle un paquet de données était en transit. Les autres routes utilisant le même lien

doivent êtres marquées comme invalides, mais le nœud chargé d'assurer la réparation locale doit indiquer que les routes nouvellement perdues sont en réparation locale; cet indicateur de réparation locale dans la table de routage doit être remis à zéro lorsque le temps de transit est écoulé (i.e. après que la route a été inactive pour *ACTIVE_ROUTE_TIMEOUT*). Avant que le délai d'attente s'écoule, ces autres routes seront réparées en cas de besoin lorsque les paquets arrivent pour les autres destinations. Sinon, en fonction de la congestion locale, le nœud peut commencer le processus de réparation locale pour les autres routes, sans attendre que de nouveaux paquets arrivent. De manière proactive lors de la réparation des routes qui ont été rompues par perte de lien, les paquets de données entrants pour ces routes ne seront pas soumis au délai de réparation de la route et peuvent êtres transmis sans délai.

La réparation de la route avant qu'un paquet de données est reçu pour cette dernière provoque le risque de réparer les routes qui ne sont plus en usage.

Par conséquent, en fonction du trafic local dans le réseau et que si la congestion est connue, le nœud peut de façon proactive réparer les routes avant un paquet de données est reçu, sinon, ça peut attendre jusqu'à ce que les données sont reçues, et puis commencer la réparation de la route.

IV.4 Paramètres de configuration de l'AODV

Cette section décrit les valeurs par défaut pour certains paramètres liés aux opérations du protocole AODV (Tableau 4.7). Les paramètres : *NET_DIAMETER, MY_ROUTE_TIMEOUT, ALLOWED_HELLO_LOSS, RREQ_RETRIES*, et éventuellement *HELLO_INTERVAL* peuvent être modifiés en cas de besoin. Le choix de ces paramètres peut affecter les performances du protocole.

Désignation des paramètres	Valeur
ACTIVE_ROUTE_TIMEOUT	3,000 milliseconds
ALLOWED_HELLO_LOSS	2
BLACKLIST_TIMEOUT	*RREQ_RETRIES * NET_TRAVERSAL_TIME*
DELETE_PERIOD	voir note ci-dessous
HELLO_INTERVAL	1,000 milliseconds
LOCAL_ADD_TTL	2
MAX_REPAIR_TTL	*0.3 * NET_DIAMETER*
MIN_REPAIR_TTL	voir ci-dessous
NET_DIAMETER	35
NET_TRAVERSAL_TIME	*2 * NODE_TRAVERSAL_TIME * NET_DIAMETER*
NEXT_HOP_WAIT	*NODE_TRAVERSAL_TIME + 10*
NODE_TRAVERSAL_TIME	40 milliseconds
PATH_DISCOVERY_TIME	*2 * NET_TRAVERSAL_TIME*
RERR_RATELIMIT	10
RING_TRAVERSAL_TIME	*2 * NODE_TRAVERSAL_TIME * (TTL_VALUE + TIMEOUT_BUFFER)*
RREQ_RETRIES	2
RREQ_RATELIMIT	10
TIMEOUT_BUFFER	2
TTL_START	1
TTL_INCREMENT	2
TTL_THRESHOLD	7
TTL_VALUE	voir ci-dessous

Tableau 4. 7: valeurs par défaut des paramètres de l'AODV

La valeur pour *MY_ROUTE_TIMEOUT* doit être au moins *2 * PATH_DISCOVERY_TIME*.

MIN_REPAIR_TTL doit être le dernier nombre de sauts connu pour la destination. Si les messages *HELLO* sont utilisés, la valeur d'*ACTIVE_ROUTE_TIMEOUT* doit être supérieure à la valeur (*ALLOWED_HELLO_LOSS * HELLO_INTERVAL*). La valeur *HELLO_INTERVAL* peut nécessiter un certain nombre d'ajustements pour une valeur donnée d'*ACTIVE_ROUTE_TIMEOUT*.

TTL_VALUE est la valeur du champ TTL dans l'en-tête IP tant que la recherche étendue progressivement est en cours d'exécution (voir la section : Contrôle de diffusion des paquets RREQ). *TIMEOUT_BUFFER* est aussi configurable, son rôle est de fournir un tampon pour le délai d'attente de sorte que si un paquet RREP est retardé en raison de la congestion, un délai d'attente est nécessaire tant que le paquet RREP est toujours en route vers la source. Pour omettre ce tampon, *TIMEOUT_BUFFER* doit être nul.

DELETE_PERIOD fournit une limite supérieure sur le temps pendant lequel un nœud en amont "A" peut avoir un voisin "B" comme prochain nœud actif pour la destination "D", tant que "B" a invalidé la route vers "D". Après cette période "B" peut supprimer (déjà invalidée) la route vers "D". La détermination de la borne supérieure dépend un peu des caractéristiques de la couche liaison sous-jacente. Si les messages HELLO sont utilisés pour déterminer la disponibilité continue des liens vers les prochains nœuds, *DELETE_PERIOD* doit être au moins *ALLOWED_HELLO_LOSS * HELLO_INTERVAL*. Si la couche de liaison est utilisée pour détecter la perte de liens, *DELETE_PERIOD* doit être au moins *ACTIVE_ROUTE_TIMEOUT*.

Si les messages HELLO sont reçus par un voisin, mais les paquets de données pour ce voisin sont perdus (par exemple, en raison de l'asymétrie temporaire des liens), des hypothèses plus concrètes sont faites sur la couche de liaison sous-jacente. On suppose que l'asymétrie ne peu avoir lieu au-delà d'un certain temps, disons, un multiple de « *k* » *HELLO_INTERVAL*. En d'autres termes, un nœud, recevra au moins un des "*k*" suivants messages HELLO d'un voisin si la liaison fonctionne et que le voisin n'envoie pas d'autres trafics. Pour couvrir toutes les possibilités, *DELETE_PERIOD = max * k (ACTIVE_ROUTE_TIMEOUT, HELLO_INTERVAL)* avec *k* = 5 est une valeur recommandée.

NET_DIAMETER mesure le maximum possible en nombre de sauts entre deux nœuds dans le réseau. *NODE_TRAVERSAL_TIME* est une estimation moyenne de parcours d'un nœud pour les paquets et il devrait inclure les délais de mise en files d'attente, les délais de traitement des interruptions et les temps de transfert. Si la couche de liaison est utilisée pour détecter la rupture de liens comme dans le standard IEEE 802.11 [16], la valeur *ACTIVE_ROUTE_TIMEOUT* doit être ajusté à une grande valeur (au moins 10.000 millisecondes). Si les messages HELLO sont utilisés pour la gestion de la connectivité locale, *TTL_START* devra être fixé à au moins 2.

BLACKLIST_TIMEOUT doit être convenablement augmenté si la recherche étendue progressivement est utilisée. Dans de tels cas et pour prendre compte les multiples tentatives de découverte de routes supplémentaires, il convient de prendre comme valeur : *{[(TTL_THRESHOLD - TTL_START) / TTL_INCREMENT] + 1 + RREQ_RETRIES}*NET_TRAVERSAL_TIME*.

La performance du protocole AODV dépend des valeurs choisies, des caractéristiques du protocole de la couche liaison sous-jacente, des technologies radios, etc.…

IV.5 Avantages et Inconvénients

L'un des avantages d'AODV est l'utilisation de numéros de séquences (*NSeq*) dans les messages. Ces *NSeq* permettent d'éviter le problème de boucles infinies et sont essentiels au processus de mis à jour de la table de routage. Un autre avantage est le rappel de l'adresse IP source dans chaque message. Ceci permet de ne pas perdre la trace du nœud à l'origine de l'envoi du message lors des différents relais.

L'inconvénient de l'AODV est qu'il n'existe pas de format générique des messages (i.e. chaque message a son propre format : RREQ, RREP, RERR) [140].

IV.6 Simulation du protocole AODV

IV.6.1 Introduction

L'objectif ici est d'analyser par simulation, les propriétés et les performances du protocole AODV dans un environnement de type ad hoc. Cette étude est faite pour justifier le choix de cet algorithme que nous avons utilisé comme algorithme de routage dans nos différentes contributions. L'étude portera surtout sur l'aspect charge de contrôle, efficacité et fiabilité.

IV.6.2. Description du scénario simulé

Modèle de simulation :
La simulation est faite sur un réseau ad hoc constitué de six nœuds mobiles. Le contexte de notre simulation est spécifié par les paramètres indiqués dans le tableau ci-dessous (Tableau 4.8) [141].

Paramètres	Valeur
Temps de simulation	150s
Protocole	AODV
Taille d'un paquet de données	512
Topologie de simulation	500x500
Taille du buffer associé aux nœuds	50
Vitesse d'émission (débit)	5 paquets/s

Tableau 4. 8: paramètres de simulation

Les paramètres standards pour le medium et le modèle de propagation radio sont : une capacité du médium de 2MB/s avec le modèle de propagation radio « Two Ground ». Le protocole d'accès au medium utilisé est IEEE 802.11. Le type d'interface de la file d'attente associée à chaque nœud est « Drop Tail » qui se base sur le principe FIFO pour l'ensemble des paquets venant de différents flux. En cas de file pleine, le dernier paquet arrivé est supprimé. Le nombre de paquets maximum dans le tampon d'émission au niveau de chaque routeur est de 50 paquets. Le tableau 4.9 illustre les paramètres internes du protocole AODV [141]:

Paramètres d'AODV	Valeur
Durée de vie des routes dans la table de routage	10
Intervalle des messages HELLO (*HELLO_INTERVAL*)	1
Nombre de fois que le RREQ est diffusé (*RREQ_RETRIES*)	3
TTL_START	5
TTL_INCREMENT	2
Nombre autorisé de messages HELLO perdus (*ALLOWED_HELLO_LOSS*)	3

Tableau 4. 9: Paramètres relatifs à l'AODV

Modèle de trafic :
Les sources de trafic utilisées sont de type CBR (*Constant Bit Rate)*. Ces sources de trafic modélisent la couche application sur des agents de transport UDP (*User Datagram Protocol*) et émettent les paquets de données à intervalles réguliers. Le but d'utilisation du protocole UDP est que les transferts se font de manière à éviter de gérer le contrôle de flux qui conduirait à une analyse plus complexe des résultats de simulation. Les nœuds du réseau ont des rôles différents. La source dans notre cas (i.e. le modèle de simulation utilisé) est le nœud 0 qui émet des paquets de taille 512 octets selon un débit de 5 paquets par seconde (voir Tableau 4.8) [141].

IV.6.3. Les paramètres à évaluer

Afin de bien évaluer les performances d'un réseau, un certain nombre de paramètres doivent être mesurés. Les plus importants sont :

- Le PDR (Packets Delivery Ratio) qui est le taux de paquets livrés avec succès sur le nombre total des paquets transmis. Ce paramètre nous renseigne sur l'efficacité du réseau,
- Le NOL (Normalized Overhead Load) qui n'est autre que le taux des paquets de contrôles émis par rapport au nombre de paquets reçus. Ce paramètre nous informe sur la quantité des paquets de contrôle générés par le protocole AODV lors de la phase de découverte et de maintenance de routes,
- Le taux des paquets perdus c'est-à-dire le pourcentage entre paquets émis et ceux reçus avec succès. Ce paramètre exprime la fiabilité du réseau, et
- Le débit (throughput en kbps) qui indique le taux de transfert de données. Avoir un réseau ou le débit est élevé est chose souhaité.

IV.6.4 Analyses et discussion des résultats de simulation

IV.6.4.1 Taux des paquets livrés avec succès (PDR)

Le tableau ci-dessous (Tableau 4.10) récapitule le nombre de paquets (émis, reçus et perdus) avec le pourcentage pour une durée de simulation de 150 secondes.

Paquets	Générés	Reçus	Perdus
Total	26144 (100%)	19555 (74.79%)	6589 (25.21%)
AODV	228 (0.87%)	199 (0.76%)	29 (0.11%)
CBR	6389 (24.43%)	2752 (10.52%)	3637 (13.91%)
ACK	6448 (24.66%)	3685 (14.09%)	2763 (10.57%)
ARP	24 (0.11%)	24 (0.11%)	0 (0%)
RTS	6607 (25.27%)	6447 (24.65%)	160 (0.62%)
CTS	6448 (24.66%)	6448 (24.66%)	0 (0%)

Tableau 4. 10: nombre de paquets (émis, reçus et perdus)

Le tableau 4.10, montre que dans cette simulation, le taux de livraison des paquets est de 75.75%. Le fait que ce taux soit différent de 100% est dû à la mobilité des nœuds ce qui engendre des coupures de liaisons et ça provoque la perte des paquets qui est représenté par un taux de 25.25%. Une bonne exploitation de ces résultats (Tableau 4.10) est schématisée par la Figure 4. 8. Cette figure montre que la plupart des paquets reçus sont celles qui contrôlent les canaux de transmission (RTS/CTS), par contre la moitié des données de types CBR utiles aux applications se retrouve perdue comme conséquence de la mobilité des nœuds qui augmente la longueur des routes et par conséquent, les messages HELLO échangés sont plus nombreux, en effet le nœud source, doit attendre pendant un certain temps durant lequel les paquets de données en cours de route vont être perdus [141].

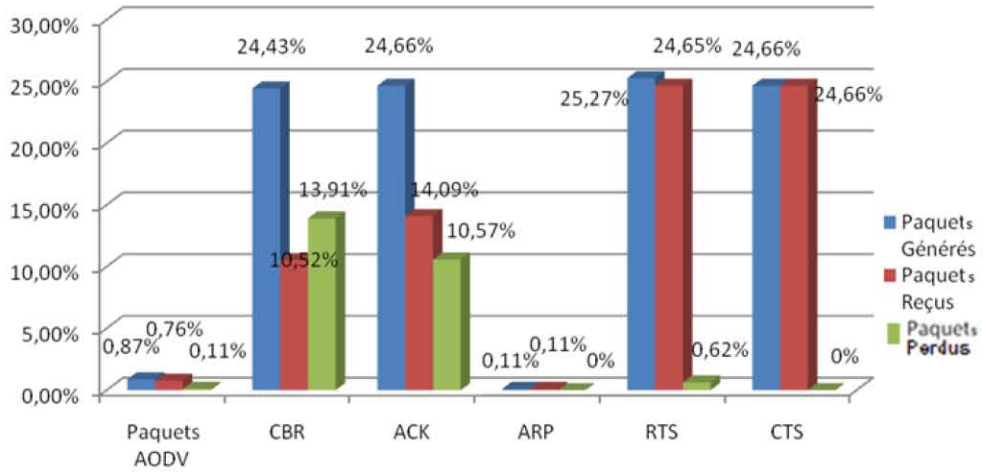


Figure 4. 8: taux des paquets émis, reçus et perdus.

Le schéma ci-dessous (Figure 4. 9), montre que le débit des paquets reçus en fonction du temps au niveau du nœud 0. Initialement le débit était faible au début de la simulation, ensuite il augmente et à la fin il chute suite aux mouvements des nœuds ce qui réduit considérablement le nombre de paquets reçus. Le débit lui aussi dépend de la stabilité des nœuds. Il augmente jusqu'à un débit maximal lors de la reconstruction de routes, ensuite il se stabilise pendant presque une période de 60 secondes (moins de mouvements des nœuds) et par la suite, il diminue jusqu'à ce qu'il devient nul suite à la perturbation de réseau et l'éloignement des nœuds provoquant la coupure des liaisons (rupture de connexion) pendant un temps de 10 secondes. Après reconstruction des liens cassés, le débit se stabilise de nouveau et ce jusqu'à la fin de la simulation [141].

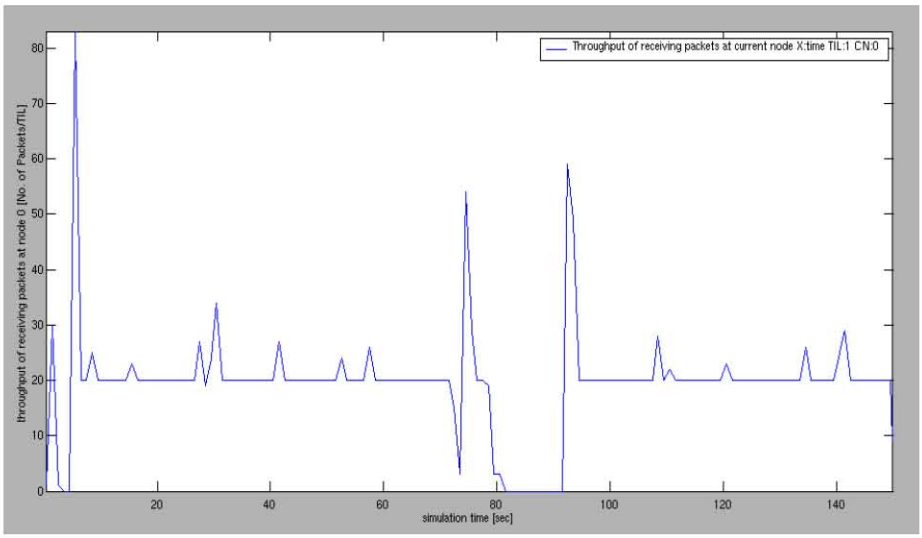


Figure 4. 9: débit des paquets reçus dans le nœud 0.

IV.6.4.2 Trafic de Contrôle (NOL)

La Figure 4. 10 montre l'évolution du pourcentage de trafic de contrôle du protocole AODV en fonction des types des paquets de contrôle émis (générés), reçus et perdus (RREQ, RREP et RERR).

Paquets	Emis	Reçus	Perdus
AODV	228 (100%)	199 (87.27%)	29 (12.71%)
RREQ	162 (71.05%)	156 (68.42%)	6 (2.63%)
RREP	47 (20.61%)	24 (10.53%)	23 (10.08%)
RERR	19 (8.34%)	19 (8.34%)	0 (0%)

Tableau 4. 11: taux des paquets de contrôle

La majorité des paquets générés sont des paquets de type requête de route (RREQ) (71.05%) (Tableau 4.11), ceci est du à la forte demande d'établissement de routes que ce soit au début ou suite aux liaisons cassées ce qui engendre de nouvelles reconstructions (d'autres paquets RREQ) et par conséquent de nouvelles émissions des paquets de RERR (8.33%) pour aviser les nœuds des routes coupées. Le taux des paquets de réponse de route RREP est de 20.62% [141].

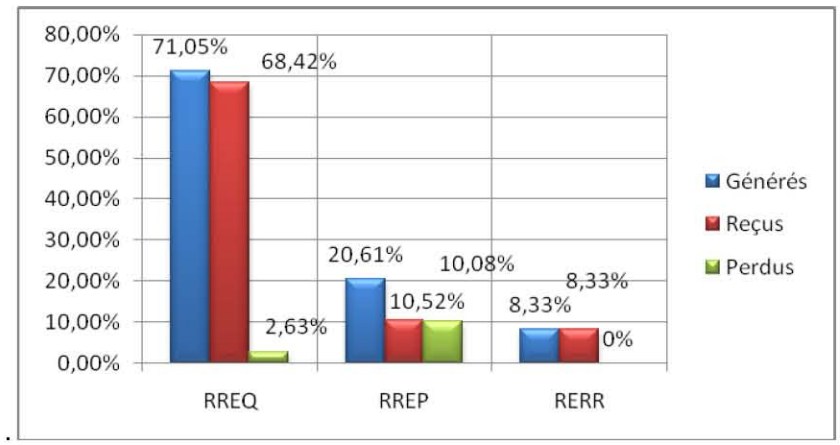


Figure 4. 10: Trafic de contrôle.

La Figure ci-dessous (Figure 4. 11) montre les paquets AODV (RREQ, RREP, RERR) reçus au niveau de tous les nœuds.

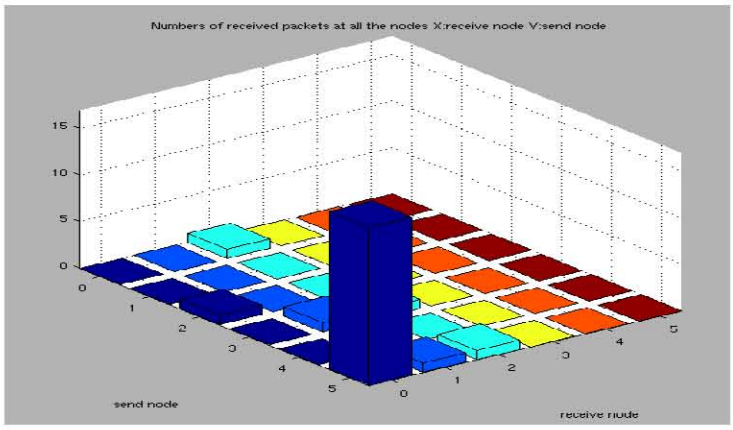


Figure 4. 11: les paquets AODV reçus.

D'après la Figure 4. 12, on peut constater que la perte de la majorité des paquets AODV est due soit :

- Lors du lancement de la simulation, ou
- Dans les périodes [~26, ~ 34] et [120, ~122], les intervalles considérés à forte mobilité des nœuds.

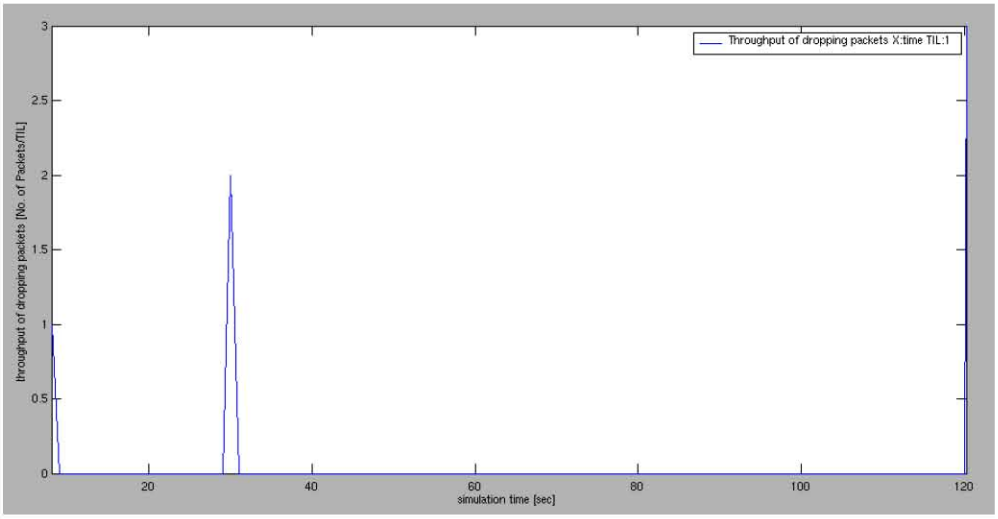


Figure 4. 12: débit des paquets AODV perdus sur le réseau.

Pour mieux connaitre l'impact de la mobilité sur le comportement de l'AODV, nous avons simulé le même réseau mais cette fois ci avec une faible mobilité (moins de mouvement des nœuds) pour bien voir comment ce protocole doit se conduire dans de telles situations. Dans ce qui suit, une simple comparaison entre le réseau précédent (forte mobilité) et ce dernier (faible mobilité) est faite:

En premier lieu, on s'intéresse aux taux de paquets livrés (Tableau 4.12). On constate que le taux des paquets perdus est inférieur et par conséquent le taux des paquets reçus est meilleur dans cette nouvelle situation (réseau à faible mobilité) par rapport à la précédente (forte mobilité) [141].

	Réseau à faible mobilité			Réseau à forte mobilité		
PAQUETS	Générés	Reçus	Perdus	Générés	Reçus	Perdus
Total	29160 (100%)	21884 (75.04%)	7276 (24.96%)	26144 (100%)	19555 (74.79%)	6589 (25.21%)
AODV	28 (0.09%)	23 (0.07%)	5 (0.02%)	228 (0.87%)	199 (0.76%)	29 (0.11%)
Autres Paquets	29132 (99.91%)	21861 (74.97%)	7271 (24.94%)	25916 (99.13%)	19356 (74.07%)	6560 (25.06%)

Tableau 4. 12: paquets émis, reçus et perdus dans les deux réseaux

D'après la Figure 4. 13 et le tableau 4.13 (ci-dessous), on constate que le trafic de contrôle dans un réseau à faible mobilité est inférieur que celui de la situation précédente (i.e. un taux de presque (~0.11%) qui est nettement inférieur au taux (~0.87%) dans le réseau à forte mobilité).

	Réseau à faible mobilité			Réseau à forte mobilité		
Paquets	Générés	Reçus	Perdus	Générés	Reçus	Perdus
AODV	28 (100%)	23 (82.13%)	5 (17.87%)	228 (100%)	199 (87.29%)	29 (12.71%)
RREQ	15 (53.57%)	15 (53.57%)	0 (0%)	162 (71.05%)	156 (68.42%)	6 (2.63%)
RREP	11 (39.29%)	6 (21.42%)	5 17.87%)	47 (20.61%)	24 (10.53%)	23 (10.08%)
RERR	2 (7.14%)	2 (7.14%)	0 (0%)	19 (8.34%)	19 (8.34%)	0 (0%)

Tableau 4. 13: paquets de contrôle dans les deux réseaux

Donc, on peut conclure que le trafic de contrôle croit avec l'augmentation de mouvement des nœuds. La mobilité des nœuds provoque généralement la rupture des liens (connexions) et par conséquent, il y a rémission de plus de paquets d'erreur (RERR) pour aviser les paires communicantes (source et destination) de ces déconnexions et de plus de paquets de découverte de route (RREQ) pour le rétablissement des liens coupés ou la reconstruction de nouvelles routes [141].

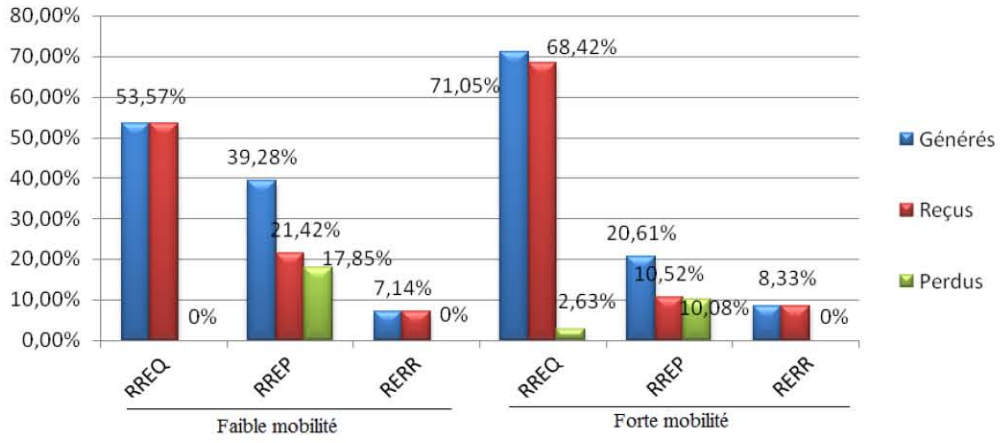


Figure 4. 13: Trafic de contrôle dans les deux réseaux

IV.7 Conclusion

Dans la première partie de ce chapitre, après une brève introduction, nous avons présenté la structure des paquets utilisés par le protocole AODV, suivi des procédures de découverte et de maintenance de routes.

Dans la seconde partie, nous avons essayé d'estimer les performances et le comportement de ce protocole dans un environnement ad hoc c'est-à-dire comment ce protocole réagit en fonction de la mobilité de ses nœuds, et son impact sur la charge du trafic circulant dans le réseau, le débit et la perte des paquets. Sur la base de ces simulations, nous pouvons dire que le protocole AODV est bien adapté au routage dans les réseaux ad hoc, comme il offre de bonnes performances (efficacité et fiabilité) et s'adapte bien au phénomène de mobilité.

Après avoir exposé et justifié le choix du protocole AODV, le chapitre suivant sera consacré à la description de notre contribution qui n'est autre qu'une série de modification qui portera sur cet algorithme

Chapitre 5 : Description de la Contribution

V.1 Motivations

Notre contribution se base essentiellement sur le protocole AODV pour plusieurs raisons. Il fait partie des protocoles standardisés, il est le plus traité en littérature, il combine les avantages de deux autres algorithmes (DSR et DSDV), et il offre de bonne performances dans un environnement mobile avec une grande population de nœuds (cf. chapitre 4).

Parmi les indications qui nous ont incitées à modifier ce protocole, on cite: les mécanismes qui causent la perte des paquets de données et ceux qui ralentissent le délai de transfert de ces derniers ce qui engendre une mauvaise consommation de la bande passante.

Pour le premier cas, deux situations sont possibles. La première est la cause directe de la rupture des liens due à la mobilité des nœuds, et que les nœuds en amont des liens coupés, continuent à envoyer les paquets de données sans se rendre compte de cette situation.

La deuxième quand un nœud lance une réparation locale d'un lien coupé, le nœud source n'est pas informé de cette situation et continue lui aussi l'envoi des paquets de données de façon normale vers la destination. A leurs arrivés (i.e. paquets de données) au niveau du nœud chargé d'exécuter la réparation locale, ils sont mis dans sa file d'attente. Deux cas peuvent avoir lieu. Le premier est que le temps mis pour réparer le lien est assez grand ce qui engendre un débordement de la file d'attente et le nœud se trouve dans l'obligation d'éliminer un nombre important de paquets (ce nombre est inversement proportionnel au temps mis pour réparer un lien et des paquets déjà dans la file d'attente). Le second cas est l'échec de l'opération de réparation locale ce qui nous amène à aviser la source par un paquet RRER pour qu'elle relance le processus de découverte de routes et le nœud ayant initié la réparation locale est dans l'obligation d'éliminer l'ensemble des paquets transmis par cette source.

Parmi les problèmes rencontrés lors d'une réparation locale on trouve: la génération des paquets de contrôle (RREQ) redondants, le temps consommé par cette opération et la génération des routes inutiles

Concernant le délai de transfert, il est la conséquence du phénomène de retransmission. Plus qu'un paquet est retransmis, plus son délai est élevé. Les retransmissions sont effectuées soit après une perte, soit lors d'un accès compétitif au canal dans le cas d'une gestion avec contention de type CSMA ou autres.

La perte peut être améliorée soit par une nouvelle gestion de la procédure de réparation locale, soit une meilleure prédiction de la rupture des liens en se basant sur la qualité du signal entre nœuds voisins faisant partie d'un chemin actif (i.e. en cours d'utilisation) ou soit en prévoyant des routes de secours qu'on utilisera si rupture aura lieu.

Le délai peut être aussi réduit par une amélioration de la procédure de gestion DCF dans la couche MAC.

Le reste de ce chapitre est organisé ainsi : après une brève description de l'architecture pour le support de la QdS, une étude détaillée sur la contribution proposée que ce soit au niveau routage (AODV-SR, M-AODV et PF-AODV) ou au niveau MAC (variation des valeurs de CW avec une nouvelle incrémentation de la valeur du Backoff) et nous terminons par une conclusion.

V.2. Architecture pour le support de QdS

V.2.1.Vue générale

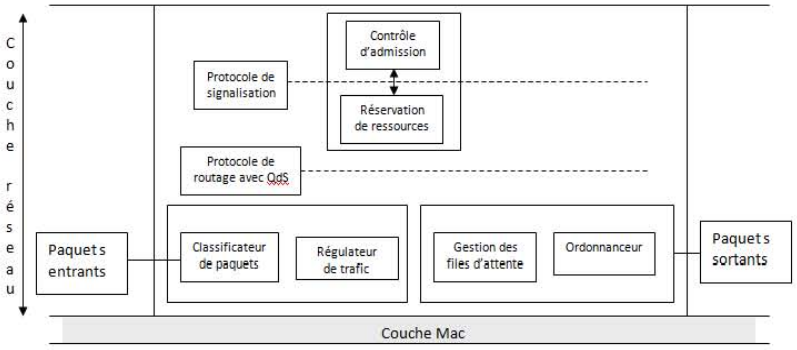


Figure 5. 1: Architecture de QdS

V.2.2. Mécanisme de QdS dans un nœud

Parmi les actions qu'un nœud doit assurer (Figure 5. 1), on a :

- **La classification des paquets** : elle se fait selon la valeur d'un champ particulier par exemple (TOS) ou sur plusieurs champs si besoin l'impose.

  Le classificateur permet de différencier entre les différents types de trafic (temps réel, best effort, etc…). A son arrivée à un nœud, le trafic est placé directement dans la file d'attente correspondante (i.e. une file par classe de QdS) selon un ensemble de métriques (le délai, la bande passante, etc…). Une file peut contenir les paquets appartenant à différentes applications avec la même classe de QdS.

  À chaque classe de trafic est associée une priorité. Le trafic de contrôle est le plus prioritaire, puis vient le trafic avec QdS et en dernier le trafic best effort.

- **Régulation des paquets** : elle se fait selon la méthode du seau percé ou le seau à jeton. Ce mécanisme est utilisé pour chaque classe de trafic. Il est déclenché dès qu'un état de congestion est détecté.
- **Gestion des buffers** (files d'attente) : elle peut être une gestion réactive ou préventive envers un phénomène de congestion. Les algorithmes de gestion d'une file *contrôlent* la taille allouée à la file en éliminant des paquets si cela est nécessaire (associer au classificateur). Un mécanisme de gestion active de la file permet aux nœuds de contrôler quand et combien de paquets seront perdus (i.e. contrôle de la congestion du réseau). Ce mécanisme permet de résoudre aussi les problèmes de saturation des files.
- **Ordonnancement des paquets** (packet scheduling) : ce module est chargé d'ordonner les paquets dans les files d'attente selon un des algorithmes d'ordonnancement (i.e. round robin). Ces algorithmes déterminent quel est le prochain paquet à envoyer sur le lien. L'ordonnanceur alloue les ressources aux flux en fonction des informations contenues dans la table des réservations.
- **Signalisation de la QdS** : elle est nécessaire pour propager les informations de signalisation (i.e. messages de contrôle), soit *in-band (i.e.* les paquets de contrôle sont véhiculés dans les paquets de données), soit *out-of-band (i.e. d*es paquets de contrôle spécifiques sont utilisés séparément), et sous quelle manière afin de réaliser une tâche particulière. La signalisation pour la QdS sert à réserver et libérer les ressources dans le réseau.
- **Contrôle d'admission** : est effectué au niveau de chaque nœud. Son rôle est l'allocation des ressources disponibles (i.e. la gestion et la régulation de la bande passante disponible). Selon la disponibilité des ressources, il décide entre l'acceptation ou le rejet d'un nouveau flux. A chaque arrivée d'un nouveau flux, des algorithmes de contrôle d'admission pour garanties déterministes de délai ou pour garanties statistiques (pertes) sont appliqués.

V.3. Description de la contribution

V.3.1 Contribution au niveau routage

Ces contributions porteront en un premier lieu sur la réduction de la charge de contrôle générée par ce protocole dans ces différentes phases (découverte, maintenance de routes et le transfert de données) pour mieux gérer la bande passante (débit). Cette réduction est réalisée par l'élimination de certaines informations de routage qui ne sont pas nécessaires et la diminution de l'espace dans lequel ces informations doivent être routées (gain en temps).

Le second point portera sur quelle stratégie adoptée lors de la rupture d'un lien de la route active durant une phase de transfert de données.

1. Augmenter la taille de la file d'attente associée au nœud en amont de la cassure pour recevoir le maximum de paquets et éviter leurs pertes en attendant que la réparation soit faite. Le problème ici est comment choisir cette taille par rapport au temps mis par la procédure de réparation. Il faut noter que la capacité (mémoire tampon) fait défaut aux nœuds ad hoc.
2. Est-ce que une réparation à la source (AODV-SR) [142] apporte une amélioration tout en réduisant le temps mis pour aviser la source pour qu'elle stoppe ses transmissions (moins de paquets perdus) et lancera une nouvelle découverte de routes. Ces actions sont à la charge du nœud ayant détecté la rupture.

Vu que les nœuds des réseaux ad hoc sont mobiles, les déconnexions sont fréquentes. Une troisième solution (M-AODV) [143] est de prévoir plus d'une route (multi chemins) et utiliser l'une d'elles pour le transfert de données et les autres comme routes de réserves (i.e. de secours ou alternatives). Le problème pour cette

solution est comment trouver un compromis entre la disponibilité de routes et le volume considérable des paquets de contrôle pour le maintien de la totalité de ces routes consommant assez de bande passante (compromis débit/perte).

Comme dernière proposition (PF-AODV) dans le contexte routage et toujours dans la modification portée sur le protocole AODV est de lancer en parallèle à l'opération de transfert de données, une action de prédiction des positions des nœuds en se basant sur la qualité du signal. Cette action, nous permet de se renseigner si un nœud est proche, loin ou il n'est plus dans le rayon de portée du nœud chargé de l'exécution de l'action de prédiction. Cette technique est utilisée pour prévoir à l'avance de nouveaux liens plus stables que ceux faisant partie de la route active ou il y a une forte probabilité de déconnexion.

V.3.1.1. Description

Comme il a été décrit dans le chapitre précèdent, la réparation locale d'une route après cassure consiste à incrémenter le numéro de séquence pour la destination, diffuser un paquet de découverte de routes (RREQ) et se met en attente pour une période de temps dite «temps de découverte». Si aucune réponse n'est reçue, le nœud chargé de la réparation locale informe la source qu'une rupture de lien s'est produite et que la source doit relancer une nouvelle phase de découverte de routes.

Notons, que la phase de réparation locale est la cause de plusieurs problèmes (perte de temps, perte des paquets, etc..) et par conséquent, elle à une influence directe sur la performance du réseau. Ce qui nous a amené à modifier la stratégie de l'AODV en déléguant cette phase qu'on a baptisée: AODV-SR (Source Repair) au nœud source. Dans AODV-SR si un lien est rompu entre une paire de nœuds, le nœud en amont du lien cassé envoi à la source un massage d'erreur RERR. Dés sa réception, la source lance une nouvelle phase de découverte de routes par la diffusion d'un nouveau paquet RREQ.

Dans ce qui suit, on va décrire les problèmes engendrés par la réparation locale dans le protocole AODV dans un réseau ad hoc. D'abord on va décrire une situation dans laquelle les nœuds qui font la réparation locale entrent dans un état similaire à un chevauchement (compétition ou course).

V.3.1.1.1. Chevauchement entre les réparations

Puisque la réparation locale est une opération autonome et décentralisée, il est difficile de savoir le nombre de réparations locales en cours d'exécution et les interactions entre elles. Quand deux processus de réparation locale sont lancés pour réparer les liens brisés pour la même route, un chevauchement entre réparations se produit qui peut avoir un effet négatif sur l'action de réparation elle même.

La Figure 5. 2 montre un état temporel, schématisant le phénomène de chevauchement entre les processus de réparation locale exécutés sur une même route, et presque en même temps sur différents liens coupés [144].

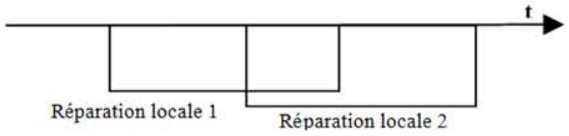


Figure 5. 2: État temporel de chevauchement entre réparations locales

V.3.1.1.2. Détection d'échec de route

Il existe trois méthodes qui permettent la détection de l'échec d'une route.

- La première repose sur l'échange de paquets Hello. Dans cette méthode (au niveau routage), tous les nœuds émettent périodiquement des paquets Hello à leurs voisins sur une route active pour les informer de leur présence. Si aucun message Hello, ou tout autre message, n'est reçu de la part d'un voisin durant un certain intervalle de temps (voir section 4.3.3.7 du chapitre 4), le lien avec le voisin est considéré comme perdu (cassé) et le cas d'échec de route est déclaré.
- La seconde méthode (au niveau MAC) se base sur les informations émanant directement de la sous-couche MAC de 802.11. Elle se base sur l'observation du niveau des signaux radio entre nœuds voisins. Un nœud considère un lien comme brisé lorsqu'il ne reçoit pas de CTS en réponse à un RTS ou bien d'ACK en réponse à un paquet de données. La sous-couche MAC de 802.11 permet de définir le nombre de retransmission de ses paquets de contrôle avant l'abandon de la transmission. Cette méthode offre une meilleure convergence (plus rapide) que la précédente et ne requiert pas de messages supplémentaires.
- La troisième méthode (au niveau routage) est basée sur la notification par un paquet RERR initié par un nœud faisant partie d'une route et qu'il est incapable de transmettre un paquet de données.

Dans le cas des deux premières méthodes, quand une réparation locale est initiée, d'autres réparations peuvent commencer sur la même route dans différents endroits suite à un déplacement rapide de quelques nœuds. Dans la troisième méthode, quand la route à une capacité de transmission faible (débit) et le taux de transmission de données est élevé, des paquets seront stockés dans les files d'attente des nœuds d'une route active, comme illustrée dans l'exemple 1 (Figure 5. 3); c'est le cas où plusieurs liens cassés peuvent être détectés presque en même temps. En outre, si le nombre de nœuds d'une route est élevé, la probabilité d'avoir plus d'un lien cassé est possible, comme illustré dans l'exemple 2 (Figure 5. 4).

Dans le cas d'un trafic bidirectionnel, la présence d'une seule coupure de lien causera une situation de chevauchement, comme illustré dans l'exemple 3 (Figure 5. 5).

Dans les trois figures ci-dessous, les réparations locales se produisent presque au même temps [144].

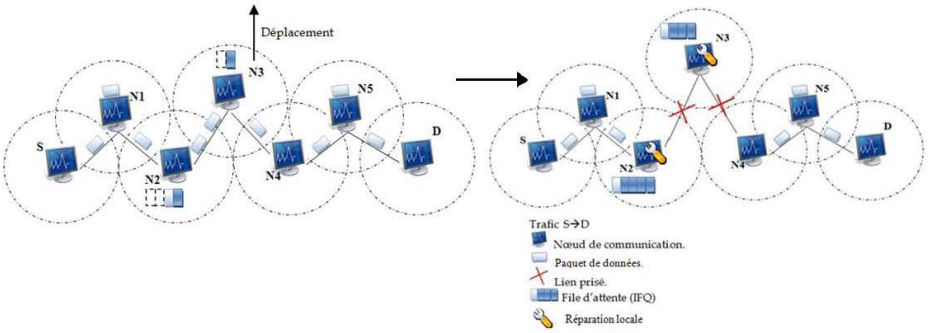


Figure 5. 3: Chevauchement exemple 1: plus d'une réparation locale (proches) [144]

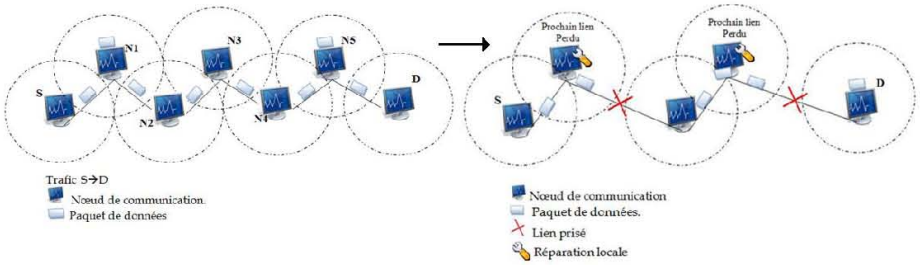


Figure 5. 4: Chevauchement exemple 2: plus d'une réparation locale (un peu éloignées) [144]

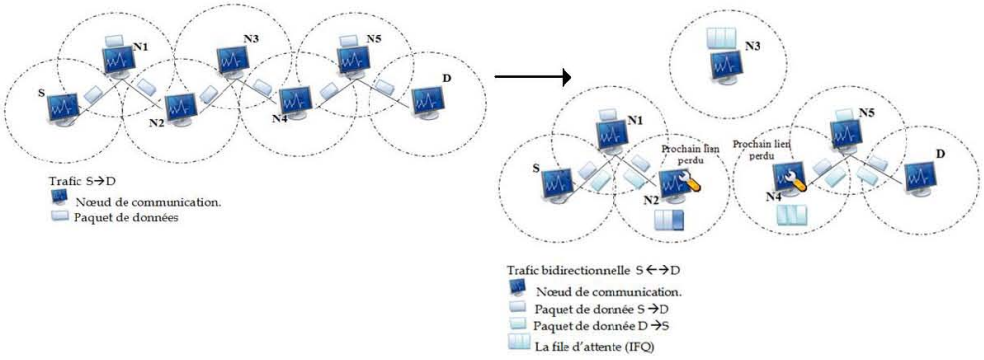


Figure 5. 5: Chevauchement exemple 3: plus d'une réparation locale (cas de trafic bidirectionnel) [144]

Quand deux réparations locales se produisent sur la même route, les nœuds proches de la destination (après le dernier lien cassé) vont recevoir tout les deux les messages RREQ de différents émetteurs (nœuds chargés de la réparation) et le phénomène de chevauchement se produit.

V.3.1.2. Problèmes provoqués par le phénomène de chevauchement

Lorsqu'un phénomène de chevauchement se produit, plusieurs problèmes surgissent [144]:

1. Génération de paquet RREQ redondant dans le réseau : lors d'une réparation locale d'une route à sens unique, les nœuds qui font la réparation locale incrémentent leurs numéros de séquence pour la destination et diffusent un paquet RREQ pour cette destination. La Figure 5. 6 illustre un cas de chevauchement avec inondation du réseau par les paquets RREQ.

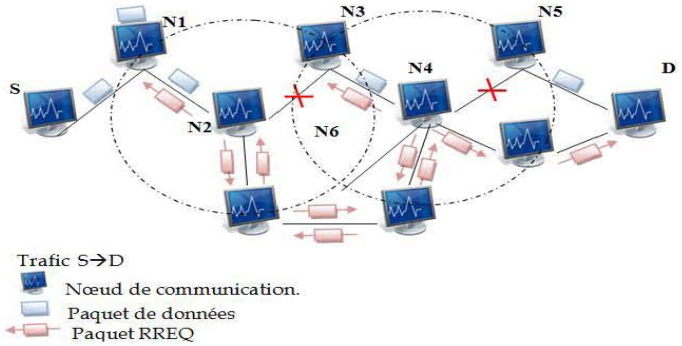


Figure 5. 6 : Génération de paquet RREQ redondant dans le réseau

2. Si le temps attribué à une réparation locale a écoulé et qu'aucun paquet RREP n'est reçu, les nœuds chargés de cette réparation vont avisés chacun de son coté la source par des paquets RRER qu'un cas d'échec s'est produit. Dans ce cas, on se heurte à un problème de génération de paquet RRER redondant dans le réseau sur les routes de retour, en plus des paquets RREQ redondant dans le sens direct. La Figure 5. 7 illustre un cas de course avec inondation du réseau par les paquets RRER.

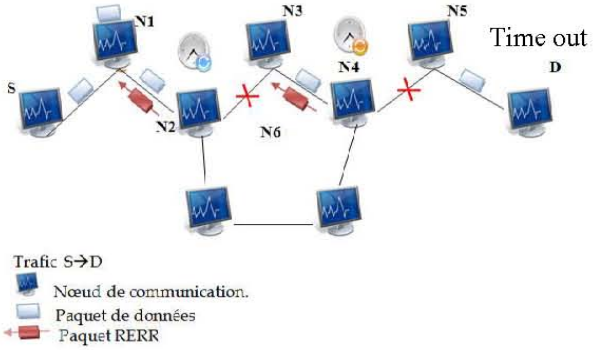


Figure 5. 7: Temps consommé par une réparation locale

Les Files d'attente des nœuds ayant déclenchés un processus de réparation locale seront chargées en attendant l'aboutissement de processus. En cas d'échec tous les paquets dans les files d'attente seront éliminés ce qui augment le taux de perte des paquets en plus du temps totale élevé de la réparation ce qui influe considérablement sur la performance de réseau.

3. Génération des routes inutiles: dans le cas de réparation d'une route bidirectionnel, il y a deux genres de chevauchements qui peuvent se produire soit dans le même sens ou dans le sens opposé.

V.3.1.3 Chevauchement de même sens

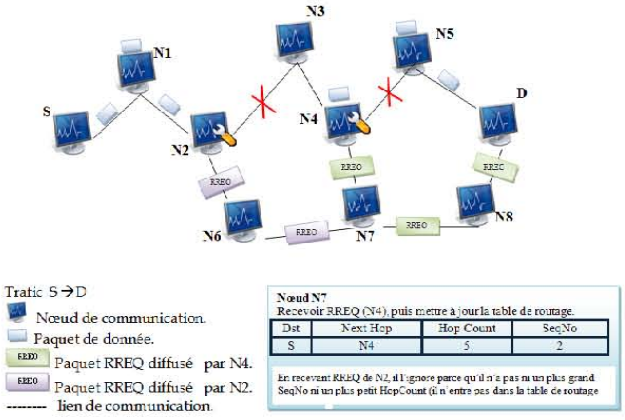


Figure 5. 8: Exemple de chevauchement dans le même sens (détection de rupture)

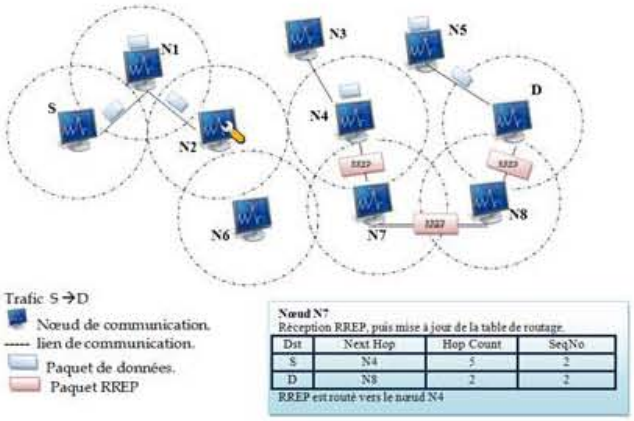


Figure 5. 9 : Un exemple de chevauchement dans le même sens (échec de réparation)

Un chevauchement dans le même sens est un chevauchement dans laquelle deux processus de réparation locale essayent de réparer la même route vers la même destination. Ce phénomène peut avoir comme conséquence soit l'échec de l'opération de réparation locale elle même ou la génération d'une route redondante inutilement.

La Figure 5. 8 montre un exemple de chevauchement de même sens entre deux processus de réparation locale pour une route donnée. Entre le nœud source (S) et le nœud destination (D), on dispose d'un chemin (S-N1-N2-N3-N4-N5-D) pour véhiculer le trafic de données.

Quand les liens N2-N3 et N4-N5 sont brisés, le nœud N2 lance une réparation locale en envoyant une requête de route (RREQ (N2)), et presque en même temps le nœud N4 lance une autre réparation locale en envoyant une autre requête de route (RREQ (N4)). Si par exemple le paquet RREQ (N4) atteint en premier le nœud N7. Ce dernier crée une entrée dans la table de routage pour le nœud S avec le nœud N4 en tant que prochain nœud.

A son tour, le nœud N7 diffuse le paquet (RREQ(N4)) à ses nœuds voisins et après un moment (plus tard) le paquet RREQ(N2) atteint également le nœud N7. Vue qu'il y'a une entrée pour le nœud S avec le même nombre de séquence dans la table de routage de N7, et le nombre de sauts (Hop count) de N2 n'est pas supérieur à celui dans l'entrée de la table de routage de N7 ; ce qui provoquera la suppression du paquet RREQ(N2). Le nœud D a renvoyé un RREP quand il a reçu RREQ (N4) et quand le paquet RREP arrive au nœud N7, il le fait router vers le nœud N3 en se basant des informations contenues de la table de routage (Figure 5. 9). Comme il n'y a aucun lien a partir de N3, le paquet RREP ne peut être expédié à aucuns autres nœuds et l'opération de réparation de la route est vouée à l'échec.

Après l'expiration du temps accordé à l'opération de réparation locale, le nœud N2 envoi un paquet RERR au nœud source pour notifier l'échec de cette opération et ce qui permettra au nœud source de lancer une nouvelle phase de découverte de routes.

Dans ce cas-ci, non seulement une route partielle N4-N7-N8-D (Figure 5. 9) est maintenue inutilement dans le réseau jusqu'à ce qu'elle soit écartée parce qu'il n'est pas utilisée au cours d'une période de temps prédéterminée, alors que la partie en amont est brisée et la transmission de données doit être suspendue jusqu'à ce que une nouvelle phase de découverte de routes soit initiée. Les paquets stockés dans la fille d'attente des nœuds N2 et N4 seront éliminés suite à l'échec de l'opération de réparation locale. Ainsi la performance du réseau se détériore en raison de ce phénomène de chevauchement [144].

V.3.1.4. Chevauchement en sens opposé

Le chevauchement dans des sens opposés se produit lorsqu'un un lien commun d'une route bidirectionnelle se coupe et les nœuds aux extrémités de ce lien tentent chacun de son coté de réparer la route.

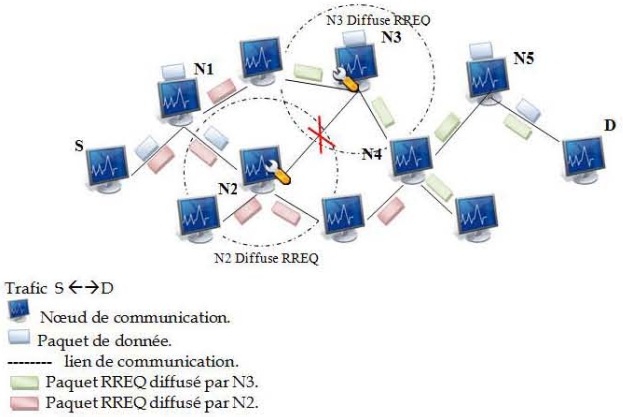


Figure 5. 10: exemple de chevauchement en sens opposés

La Figure 5. 10 illustre un exemple de ce type de chevauchement. En effet, après rupture du lien (N2-N3), les nœuds extrémités de ce lien en l'occurrence N2 et N3 détectent cette situation et chacun d'eux lance une réparation locale vers sa destination. Pour l'un c'est dans les sens (N2-N3) et pour l'autre c'est dans le sens de (N3-N2) (i.e. en aval pour l'un et en amont pour l'autre). En réalité, c'est une seule et unique route qui va être réparée des deux cotés par les deux processus de réparation locale initiés par les nœuds N2 et N3. Ce phénomène va générer inutilement beaucoup de paquets de contrôle dans le réseau [144].

V.3.2. AODV-SR (AODV Source Repair) : AODV avec réparation à la source

V.3.2.1. Motivations

Dans l'AODV original, après la rupture d'un lien d'un chemin actif, soit par mobilité ou soit par un manque d'énergie, une procédure de réparation locale (Local Repair) est activée qui prend à sa charge la reconstruction de la route à partir de ce nœud. Si un cas d'échec de la réparation est déclaré, une reconstruction à la source est initiée, et le nombre de tentatives (RREQ_RETRIES) est décrémenté de un "1", jusqu'au succès ou l'échec de l'établissement de la liaison. Cette procédure génère un volume considérable des paquets de contrôle.

Dans cette modification nommée AODV-SR, on propose d'éliminer la phase de réparation locale pour alléger la tâche au protocole AODV; le processus de découverte de routes est délégué à la source pour un ensemble de tentatives RREQ_RETRIES. Elle est motivée par le fait qu'elle est une forme très simplifiée et réduite du protocole de base AODV.

Théoriquement simplifié un processus c'est dans un but de le maitriser, mais dans ce cas, simplifier c'est réduire le volume des paquets de contrôle utilisé lors de la réparation locale. Une autre action nommée "Erreur temporaire" est intégrée à cette modification dont le but est d'aviser la source le plus tôt pour qu'elle stoppe ses transmissions, et de relancer une nouvelle découverte de routes pour minimiser la perte de paquets et rationnaliser l'utilisation de la bande passante ce qui permet d'avoir des délais de transit assez élevé.

V.3.2.2. Principe

Le protocole AODV-SR (SR pour dire **S**ource **R**epair) relance la découverte de routes chaque fois qu'un paquet dit « Erreur temporaire » lui est adressé par le nœud ayant détecté la rupture du lien. Ce paquet est identique au message RRER. Dans l'AODV de base, la procédure de réparation locale opère selon le principe de la Figure 5. 11.

Après détection d'une rupture, le nœud en amont de cette situation commence par aviser la source en utilisant le paquet RRER pour qu'elle arrête de transmettre ses données afin d'éviter un débordement de la file d'attente associée au nœud, et en second lieu lance la procédure de réparation locale. Le paquet Erreur temporaire est routé en Unicast vers la source selon le même principe de la réponse de route (RREP).

Lorsque le nœud "C" lance une réparation locale. Il diffuse un paquet RREQ vers "D" et avise "S" par un paquet RRER pour stopper ses transmissions.

Avant que la source "S" se rend compte que le lien est coupé, elle continue a envoyé les données jusqu'à ce que la file associée au nœud en amont de la rupture sera saturée. Dans le pire des cas et après échec, une recontruction à la source est initiée et c'est la cas de AODV-SR (une réparation à la source dans toutes les situations) qui nous fait gagné le temps pris par la phase de réparation locale et minimise la perte de paquets.

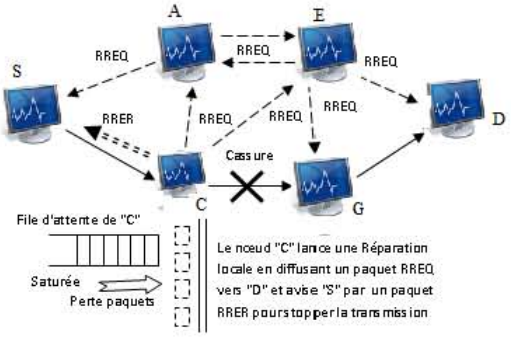


Figure 5. 11: phase de réparation locale

V.3.3. M-AODV

V.3.3.1. Motivation

Cette modification baptisée M-AODV est basée sur l'amélioration des mécanismes qui génèrent la perte des paquets de données (rupture de lien ou saturation des files associées aux nœuds), l'utilisation rationnelle de la bande passante (débit) et la réduction du délai de transit des paquets.

Deux cas sont à l'origine des pertes de paquets. Le premier est dû à la mobilité des nœuds qui provoquent des ruptures de liens dans un chemin actif (en cours d'utilisation) et que ses voisins en aval continuent à envoyer les acquittements et en amont les paquets de données pour une certaine période avant de se rendre compte que le lien est défaillant. Le deuxième est quand un nœud lance la procédure de réparation locale suite à la détection d'un lien rompu, la source n'étant pas informée de cette situation, continue donc à envoyer ses paquets de données normalement et provoque soit une saturation de la file associée au nœud sans que ces données ne soient retransmises vers leur destination.

La perte peut être donc améliorée en modifiant les mécanismes de découverte et de maintien des routes offrant ainsi une disponibilité presque permanente des liens entre les paires communicantes (multi chemins) qui exigent un volume additionnel des paquets HELLO pour maintenir les différents chemins.

Rationaliser l'utilisation de la bande passante revient à permettre plus de transfert de données utiles et moins de données de contrôles comme les paquets de recherche (RREQ, RREP, etc.), de maintien de routes (HELLO) ce qui réduit considérablement le problème de la surcharge dont souffre la quasi totalité des réseaux ad hoc.

V.3.3.2. La modification proposée (M-AODV)

La version originale de l'AODV maintient une seule route par destination. En cas de rupture une réparation locale est initiée et si elle échoue, une nouvelle découverte de routes est déclenchée ce qui génère un volume considérable des paquets de contrôle. Pour remédier à ceci, il sera préférable d'avoir des routes de réserves au niveau de chaque nœud.

Donc, le protocole M-AODV est une extension de l'AODV pour supporter le multi chemins. Il garde pour chaque nœud une route principale et des routes alternatives ou de secours qu'on utilisera quand la route principale est rompue (Figure 5. 12).

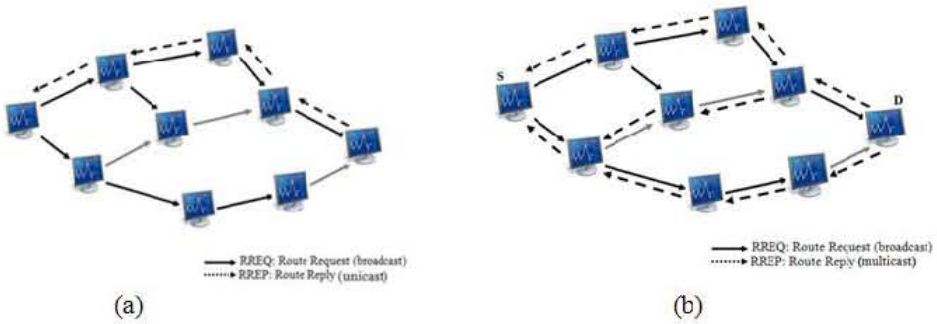


Figure 5. 12 : Routage dans (a) : AODV (b) : M-AODV

Dans cette modification l'opération de réparation locale est désactivée, et les différentes routes doivent avoir le même numéro de séquence pour simplifier leur gestion.

Quand un lien est brisé, le nœud en amont (extrémité du lien cassé) sélectionne une des routes de secours en sa possession. Si c'est le cas, il ***réoriente le transfert*** sur cette nouvelle route et supprime la route à lien cassé de sa table de routage, sinon, il envoi un paquet RRER à son voisin en amont (à un saut). De façon récurrente, ce nœud procède en cas de disponibilité de routes comme son voisin (le nœud précédent) (sélection de route et réorientation du transfert) et il élimine le paquet RRER, sinon, il le fait passer à son voisin en amont (à deux sauts) et ainsi de suite jusqu'à ce que le paquet RRER atteigne la source qui va déclencher une nouvelle découverte de routes si elle ne dispose d'aucunes routes de secours.

L'opération de maintien des routes est à la charge des nœuds faisant partie d'une des routes secondaires.

Si la route primaire (la première choisie) est utilisée pour une longue période, un message de rafraichissement (paquet HELLO) est utilisé pour les routes secondaires jusqu'à la destination.

Dans ce genre de routes, si le nombre de liens communs est élevé et s'ils sont affectés par le phénomène de coupure, cela réduit le nombre de routes établies précédemment (routes initiales).

V.3.3.3. Description de protocole M-AODV

Le protocole de routage multi-chemins M-AODV est basé principalement sur le calcul de plusieurs routes entre les paires de nœuds communicantes (Figure 5. 13). La route avec le minimum de sauts par exemple est dite route principale et les autres secondaires ou routes alternatives. Les routes sont maintenues dans les tables de routage associées aux nœuds, qu'on utilisera quand la route principale est rompue.

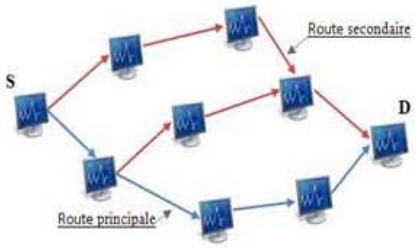


Figure 5. 13: Routage multi-chemins

Les chemins entre paires communicantes soient ils ont des liaisons communes, soient ils sont totalement disjoints.

- **Chemins totalement disjoints** : la rupture au niveau d'un lien n'affecte pas le reste des chemins et par conséquent l'utilisation d'une autre route est possible pour acheminer les données utiles (Figure 5. 14).

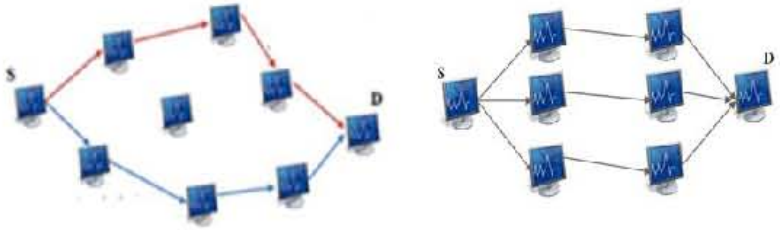


Figure 5. 14: Routes totalement disjointes (à nœuds disjoints)

- **Chemins ayant des liaisons communes** : parfois une liaison entre deux nœuds quelconques appartient à plusieurs routes et sa rupture entraîne l'annulation (rupture) de l'ensemble des routes passant par ce lien et on se retrouve avec un nombre très réduit de routes disponibles ajouter a cela on est dans l'obligation de générer un trafic additionnel très important pour avertir la source de cette déconnexion (Figure 5. 15).

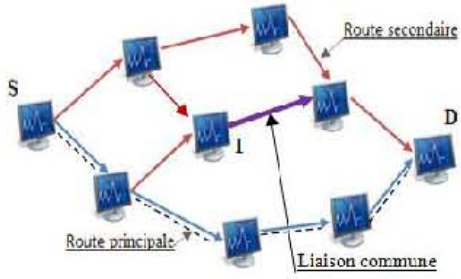


Figure 5. 15: Routes à liaison commune

Dans cette figure, le faite d'avoir une liaison commune implique l'existence de nœuds communs (non disjoints). La mobilité d'un nœud commun (i.e. qui appartient à plusieurs routes) par exemple le nœud I ou la liaison commune (trait en gras) dans la Figure 5. 15 provoque la déconnexion de toutes les routes passant par ce nœud ou par cette liaison.

V.3.3.4.1. Stratégie de routage

L'idée principale de M-AODV est de chercher plusieurs routes pendant la même phase de découverte de routes. Par la suite une seule dite principale est utilisée pour le transfert des paquets de données. La route sélectionnée en tant que principale est celle qui présente le moins de sauts ou vérifie les critères de QdS. Les autres routes calculées ne seront utilisées que lorsque la route principale est rompue (devient invalide). Ce protocole est mieux approprié pour les réseaux ad hoc où la mobilité des nœuds est importante et par conséquence la rupture de routes est fréquente. M-AODV est basé sur deux mécanismes essentiels :

– Une règle de mise à jour des routes : pour maintenir des multiples routes sans boucles de routage.

– Un mécanisme distribué entre les différents nœuds du réseau : pour calculer des routes disjointes.

Cette modification utilise aussi les deux phases (Découverte et maintien de routes) mais avec certaines modifications [144].

Table de routage MAODV vs AODV

La Figure 5. 16 ci-dessous représente la structure des tables de routage du protocole multi chemin M-AODV et celui à chemin unique AODV.

- L'*Advertised_Hop_count* de M-AODV remplace le *Hop_count* de l'AODV.
- L'entrée *route_list* remplace *Next_hop* et définit essentiellement les multiples sauts suivants des *Hop_count* correspondants.

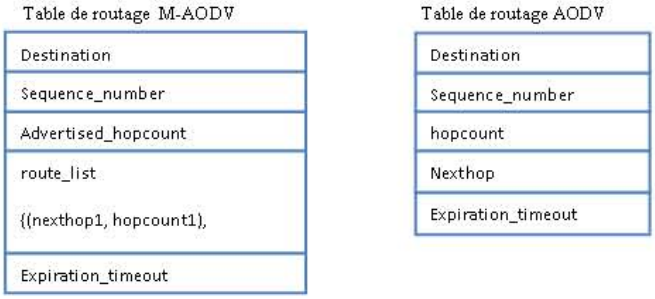


Figure 5. 16: Structure des entrées des tables de routage de M-AODV vs AODV

Le champ *Advertised_Hop_count* d'un nœud S pour une destination D représente le maximum du nombre de sauts (*Hop count*) des routes multiples disponibles pour S vers la destination D. Le maximum *Hop count* est considéré comme le nombre de sauts qui ne change jamais pour le même numéro de séquence. Le protocole permet d'accepter seulement les routes alternatives ayant un *Hop_count* inférieur à l'*Advertised_Hop_count*. Cette condition est nécessaire pour garantir des routes sans boucles de routage.

Après avoir présenté la méthode avec laquelle M-AODV peut construire des routes multiples sans des boucles de routage, le paragraphe « recherche de routes disjointes » explique comment le protocole M-AODV procède pour que ces routes soient disjointes [144].

Découverte de routes :

Dès que la source désire émettre, elle consulte sa table de routage pour une éventuelle route valide pour la destination souhaitée. Si ce n'est pas le cas, elle lance la procédure de découverte de routes en diffusant en un paquet de contrôle RREQ (Figure 5. 17).

Dés qu'un nœud quelconque dispose de routes vers la destination, il répond à la source par un paquet RREP et si ce n'est pas le cas c'est la destination qui répond à la source en diffusant en Multicast un paquet de contrôle RREP (Figure 5. 17) afin de retracer toutes les routes possibles.

Après découverte de tous les chemins possibles, celui qui présente le plus court chemin en nombre de sauts est sélectionné en premier et qui répond bien sûr au critère de QdS imposé par l'utilisateur.

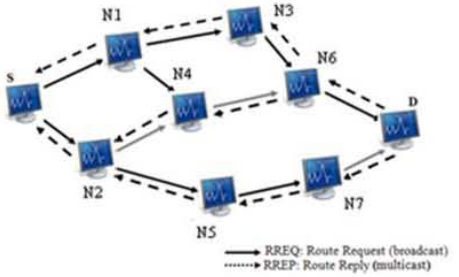


Figure 5. 17: Découverte de routes dans M-AODV

Les différentes routes possibles selon la Figure 5. 17 sont:

(S, N1, N3, N6, D), (S, N1, N4, N6, D), (S, N1, N4, N5, N7, D), (S, N1, N4, N7, D), (S, N2, N5, N7, D), (S, N2, N4, N6, D), (S, N2, N4, N7, D), (S, N2, N5, N4, N6, D) et (S, N2, N5, N4, N7, D)

Dès que le nœud N4 est hors portée, les différentes routes passant ce nœud ne seront plus valides

Les différentes routes sont sélectionnées parmi les routes passant par les nœuds de faible degré : (S, N1, N3, N6, D) et (S, N2, N5, N4, N7, D), l'une sera prise comme route principale et les autres comme routes secondaires ou de secours (Figure 5. 18).

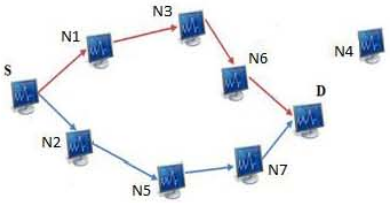


Figure 5. 18 : Routes disjointes dans M-AODV

Recherche des routes disjointes :

M-AODV permet de construire des routes multiples à liens disjoints (Figure 5. 19), c'est-à-dire des routes multiples qui n'ont pas des liens en commun entre les différentes routes entre les paires de nœuds Source-Destination. Des modifications peuvent être mises en place dans le processus de découverte de routes dans le M-AODV pour permettre la formation des routes à nœuds disjoints (Figure 5. 19 (b)).

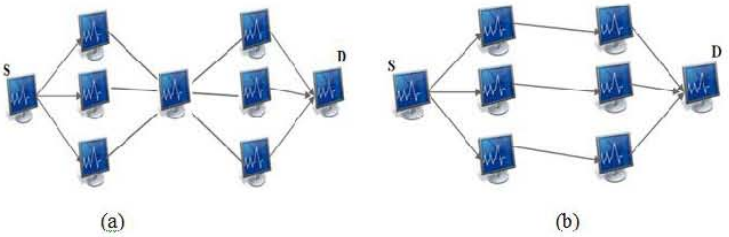


Figure 5. 19: Routes à liens disjoints : (a) à nœuds communs (b) à nœuds disjoints.

M-AODV ajoute un nouveau champ appelé *Firsthop* pour chaque paquet RREQ. Ce champ indique le premier saut (voisin de la source) qui l'a acheminé. En plus, chaque nœud maintient une liste, *Firsthop_list*, pour chaque RREQ pour garder une trace de la liste des voisins de la source à partir desquels une copie de RREQ a été reçue. Dans les nœuds intermédiaires, les copies dupliquées de RREQ ne doivent pas être immédiatement supprimées comme dans le cas de l'AODV. Chaque copie est examinée pour voir si elle procure un nouveau chemin à nœuds disjoints vers la source. Cette vérification est assurée grâce au champ *Firsthop* du paquet RREQ et la liste *Firsthop_list* du nœud. Dans le cas ou le paquet RREQ apporte un nouveau chemin, le protocole M-AODV invoque la règle de mise à jour pour vérifier si le chemin de retour peut être mis en place ou non. Si c'est le cas, le nœud intermédiaire envoie un RREP vers la source.

Comme pour les nœuds intermédiaires, la destination doit elle aussi vérifier que les chemins de retour peuvent être mis en place. Elle doit garantir que les liens sont disjoints, uniquement avec ses voisins. Au-delà du premier saut, le RREP suit les chemins de retour mis en place et qui sont déjà formés de nœuds disjoints. A son arrivée à un nœud intermédiaire, chaque RREP peut suivre plus d'un chemin de retour bien sûr dans le cas où plusieurs chemins sont déjà disponibles. La destination répond à k copies de RREQ. Le paramètre k est utilisé pour contrôler le nombre de RREP et pour éviter d'avoir un paquet RREP Storm [144].

Le routage multi chemins est basé sur trois mécanismes : la découverte des routes, la maintenance des routes et l'acheminement du trafic de données. Avec cette stratégie, ce dernier présente des avantages incontestables, à savoir :

- **Augmentation de la capacité**: La mise en œuvre d'un routage multi chemins permet une utilisation plus équilibrée des ressources du réseau. En effet, vu que les nœuds d'un réseau ad hoc souffrent souvent d'une capacité de traitement, le fait de répartir le flux sur plusieurs chemins, fait répartir par la même occasion l'utilisation des ressources des nœuds ainsi que le débit utilisé sur les liens.
- **Augmentation de la fiabilité**: A cause des échecs de liaison fréquents dans un réseau ad hoc, la disponibilité de plusieurs routes permet de garantir une livraison maximale des paquets de données. Car en répartissant les paquets successifs sur différents chemins, la perte de paquets touchera seulement ceux émis sur le chemin défectueux en attendant de trouver d'autres routes.

Maintenance de routes : Le maintien de routes est à la charge de la source, par contre les nœuds intermédiaires ne sont responsables que de leurs tables de routage.

La maintenance de l'ensemble de routes suit le même principe que celui de chemin unique c'est-à-dire l'utilisation des paquets "HELLO" (Figure 5. 20) entre voisins de l'ensemble des routes secondaires et ceci en parallèle avec l'opération de transfert de données.

En cas de non réception de trois messages HELLO consécutifs, le lien est considéré défaillant et un message RRER est envoyer vers la source pour qu'elle supprime la route passant par ce lien et le nœud ayant détecté cette coupure doit supprimer l'e-ntrée de la table de routage associée à cette route.

La période de diffusion de paquets HELLO est fixée à une durée de "*HELLO_INTERVAL*" (en ms).

Dans cette solution, le nombre de chemins totalement disjoint est fixé au début à un nombre inférieur ou égale à "n_0" avec un seuil de "s_0". Une fois le seuil est atteint, en parallèle du transfert des données, la recherche de nouvelles routes est initiée pour déterminer d'autres chemins jusqu'à atteindre "n_0".

Malgré que théoriquement cette solution engendre une charge de contrôle assez importante, elle présente l'avantage de disponibilité de routes à tout moment.

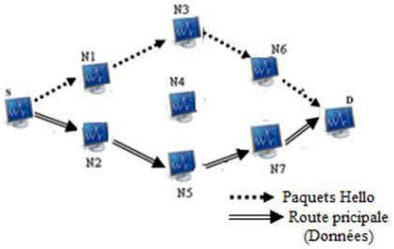


Figure 5. 20: Transfert de données et phase de maintenance

Le chemin S, N2, N5, N7, D est considéré comme principal et S, N1, N3, N6 comme secondaire (secours).

En cas de rupture de lien, la source stoppe la transmission et réitère l'opération après le choix d'une nouvelle route parmi les chemins de secours dont elle dispose.

V.3.3.4.2. Exemple de routage M-AODV

Lorsque le nœud "S" désire communiquer avec le nœud "D", il diffuse un paquet RREQ en direction du nœud "S". Lorsque ce dernier reçoit le paquet RREQ, il répond par l'envoi d'un paquet RREP pour chaque paquet RREQ reçu. Comme illustré dans la figure suivante (Figure 5. 21) [144] :

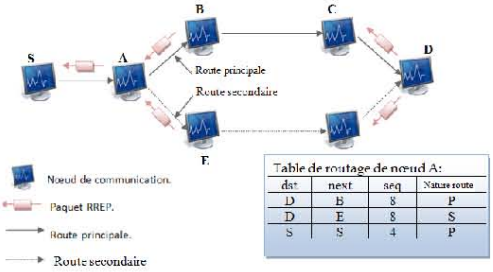


Figure 5. 21: Établissement de routes entre S et D

Une fois la route établie, le nœud S commence l'envoi de ses paquets de données sur la route principale, et en parallèle, il actualise les routes alternatives par envoi périodique de paquets HELLO (Figure 5. 22).

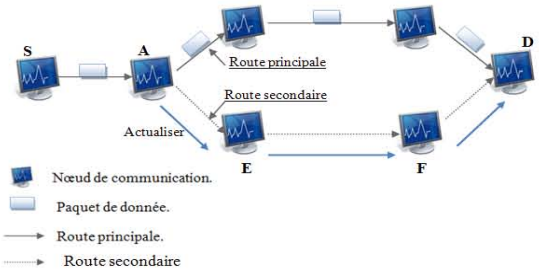


Figure 5. 22: Transfert de données et phase de maintenance de routes dans M-AODV

A un moment donné, le lien entre les nœuds B et C est rompu, un paquet RERR est envoyé vers la source par le nœud en amont de la cassure (B) pour l'aviser de cette rupture. Lorsque le paquet RERR est reçu par le nœud source. Dans le cas ou cette dernière possède une route secondaire vers D, elle relance la transmission de données, Autrement, une nouvelle phase de découverte de routes est initiée (Figure 5. 23) [144].

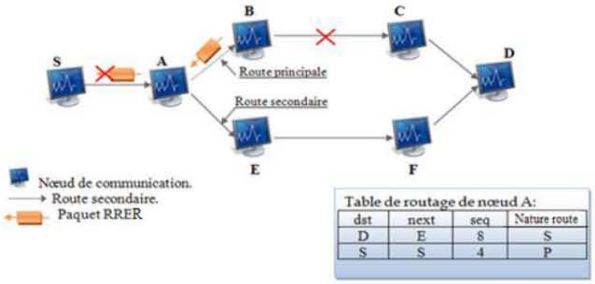


Figure 5. 23: Utilisation de la route secondaire après échec de la route principale

Dans nos simulations (Chapitre 6), nous avons fixé le nombre de chemins de secours, et aussi modifier les temps d'attente et la fréquence de rafraichissement des routes afin de minimiser les paquets de contrôle.

V.3.4. Le protocole PF-AODV : (Prédict Failure in AODV)

V.3.4.1. Motivation

Dans un environnement dynamique, les fréquentes déconnexions causent une perte considérable des paquets de données en raison de l'absence de chemins alternatifs (de secours) et la reconstruction d'une nouvelle route génère un volume additionnel des paquets de contrôles d'où la nécessité de prévoir (prédire) toutes les déconnexions (ruptures) probables sur le chemin actif (en cours d'utilisation ou primaire) en se basant sur la valeur de la puissance du signal entre nœuds voisins. Cette valeur nous informe sur la qualité du lien et elle dépend de la mobilité des nœuds. La puissance du signal permet de déterminer, si la qualité du lien s'améliore (plus stable) ou se dégrade (probabilité d'une rupture) ce qui nous permet, non seulement de rendre la gestion de liens plus robustes, et d'anticiper sur la rupture du lien et donc d'améliorer la QdS.

Lorsque la qualité du signal est en baisse suite à l'éloignement du nœud voisin, une découverte d'un morceau de route de secours de deux sauts au plus sera établie qu'on utilisera en cas de déconnexion.

Pour bien illustrer notre idée, prenons l'exemple suivant (Figure 5. 24). Les nœuds x1, x2, & xi sont voisins deux à deux et voisins à xs d'un coté et à xd de l'autre. S et sont D respectivement la source et la destination.

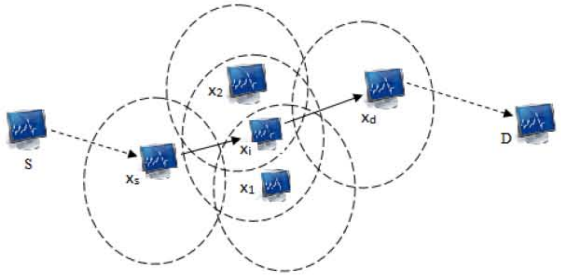


Figure 5. 24: Graphe de voisinage entre nœuds

Supposant que le chemin établit est (Figure 5. 25) :

Figure 5. 25: chemin entre S et D

Lorsque le nœud xi s'éloigne due à une mobilité, son signal s'affaiblit (tend vers zéro) et la probabilité de déconnexion augmente. Deux cas sont à envisager (respectivement Figure 5. 26 et Figure 5. 28):

1. xi se déplace vers xd et quitte la portée de xs (coupure du segment [xs, xi] ; Figure 5. 26), alors xs essaye de reconstruire le morceau du chemin [xs, xp, xp, xi] (Figure 5.27) où p est un des voisins a un saut de xs (dans l'exemple : p=1 ou 2 c'est-à-dire xp=x1 ou xp=x2).

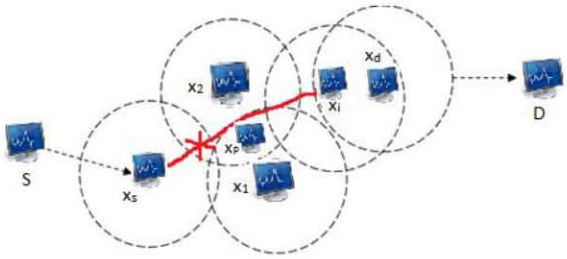


Figure 5. 26: déplacement de xi du coté de xd avec coupure de [xs-xi]

Figure 5.27 : Reconstruction du coté de xs

2. xi se déplace vers xs et quitte la portée de xd (coupure du segment [xi, xd] ; Figure 5. 28), alors xs essaye de reconstruire le morceau du chemin [xi, xp, xp, xd] (Figure 5. 29) où p est un des voisins à un saut de xs (dans l'exemple : p=1 ou 2 c'est-à-dire xp=x1 ou xp=x2) et un saut de xd.

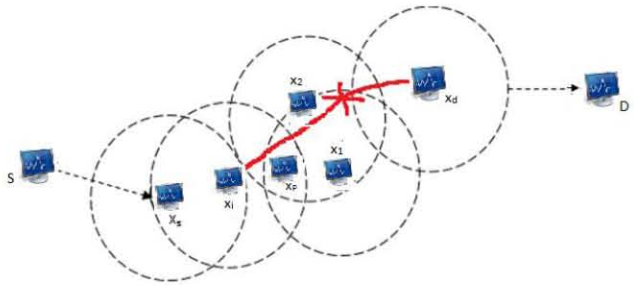


Figure 5. 28: déplacement de xi du coté de xs avec coupure de [xi-xd]

Figure 5. 29: Reconstruction du coté de xd

Comment sont reconstruit ces segments de routes (Figure 5.27 et Figure 5. 29) chose qui est faite bien sûr en parallèle avec le transfert de données sur une route active est l'idée implémentée dans le protocole PF-AODV dans sa phase de routage où une métrique nommée « puissance du signal » est utilisée pour déterminer si le lien est stable ou non.

Il faut noter qu'il est très difficile de prédire la probabilité exacte qu'un lien sera brisé dans un avenir proche mais il est possible d'estimer la stabilité relative de la liaison sur la base des récentes valeurs de puissance du signal reçues sur le lien.

Le calcul de la puissance du signal obéit à l'un des modèles SBM ou ASBM [145].

V.3.4.2. Modèles de mesure de la puissance du signal

La puissance du signal transmis nous informe sur l'état du lien, et plus précisément de sa « stabilité ». Son calcul est effectué soit à partir de mesures réelles du signal ou selon le modèle analytique.

Dans le modèle SBM (*Signal strength Based link stability estimation model)* (Figure 5. 30), chaque nœud mobile mesure la puissance du signal (SS : Signal Strength) des nœuds voisins. Plus le signal reçu d'un voisin est fort (par rapport à un seuil) plus ce voisin est proche et le lien est considéré comme stable.

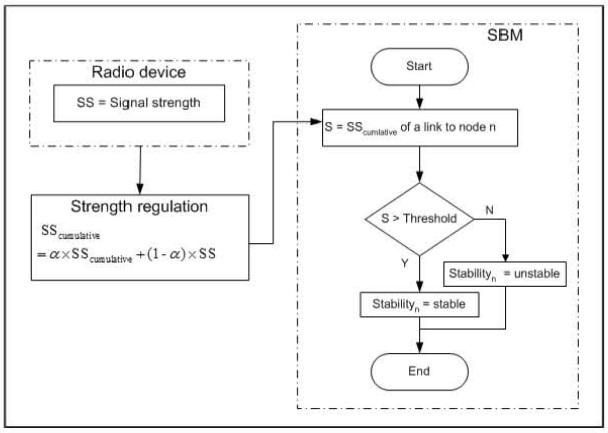


Figure 5. 30: Schéma fonctionnel de SBM

La stabilité du lien est déterminée par l'envoi de signaux pilotes avec le modèle PBM (*Pilot signal based link stability estimation model)* (Figure 5. 31).

Quand un nœud reçoit de ses voisins le signal pilote, il l'enregistre. S'il reçoit continuellement des signaux d'un voisin et que le nombre de signaux reçus consécutivement dépasse un certain seuil, il considère le lien entre eux comme stable. Si un nœud ne peut pas recevoir de signal pilote d'un lien dans un délai bien fixé, il considère alors ce lien comme instable.

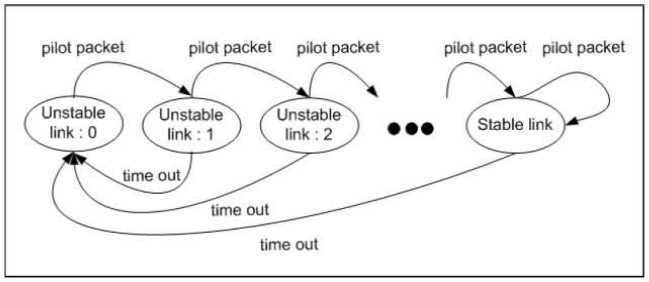


Figure 5. 31: Modèle de stabilité du signal pilote

Le modèle ASBM (Figure 5. 32) s'appuie sur le modèle SBM (Figure 5. 30) mais légèrement modifié pour prendre en compte, en sus de la puissance de signal (SS), la dérivée de sa mesure (DSS : differentiated signal strength). Si DSS augmente, cela est interprété par le rapprochement des deux nœuds et une durée de vie plus longue. Alors que dans SSA, seuls les liens dont la puissance du signal dépasse une certaine limite (seuil), sont considérés comme stables mais, dans ASBM les liens ayant un signal faible mais qui augmente, cas de nœuds qui se rapprochent, sont également considérés comme stables.

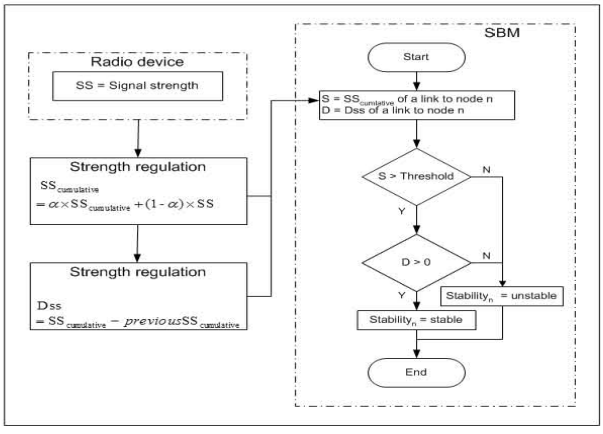


Figure 5. 32: Schéma fonctionnel de «ASBM»

V.3.4.3 Principe de routage dans PF_AODV

Le routage dans les réseaux ad hoc se compose de deux phases : la découverte et le maintien de routes. La phase de découverte est sans changement et elle est identique à celle de l'AODV de base.

a) Phase de maintenance de routes :

Comme dans l'AODV, elle est basée sur l'envoi de messages HELLO aux voisins initié par les nœuds de la route active à des intervalles réguliers et très courts.

Le paquet HELLO est modifié pour contenir des informations sur la puissance du signal du voisinage (i.e. la valeur de la puissance du signal : Val-SS). Selon Val-SS, on distingue deux type de paquets HELLO : Ordinaire (H_Ordinaire) et de Liaison (H_Liaison) respectivement pour la maintenance et la reconstruction de liens.

Un paquet H_Liaison est caractérisé par une Val-SS négative. On prend une valeur négative pour signaler au nœud voisin qu'on est en train de reconstruire un lien de secours et non une opération de maintenance.

Le paquet H_Odinaire est initié par les nœuds d'un chemin actif à leurs voisins pour une opération de maintenance; mais le H_Liaison est initié par un nœud dit de liaison qui est sélectionné par les voisins à lien le plus stable et délégué pour l'envoi de ce message à ses voisins.

La puissance du signal est utilisée pour prévoir et déterminer l'ensemble des nœuds voisins qu'on rangera dans une table des voisins "NGR_SIG" au niveau de chaque nœud par rapport à une valeur seuil choisie de telle sorte qu'on pourra reconstruire un segment de chemin dit de secours ou cache avant que le chemin actif ne soit coupé.

NB : l'intervalle entre deux (02) envois consécutifs du paquet H_Ordinaire doit être choisi de telle sorte qu'il permet le traitement de deux (02) paquets H_Liaison entre les extrémités d'un lien à reconstruire.

En recevant un paquet HELLO, le nœud décide de mettre à jour la table NGR_SIG et insère seulement les nouveaux voisins et la Val-SS associée ou il met à jour cette dernière pour les anciens voisins. Pour les voisins du chemin actif où Val-SS a atteint le seuil, une procédure notée "Liaison" est lancée pour déterminer un autre lien qui sera utilisé en cas de rupture du chemin actif et ceci par rapport à n'importe quel nœud.

La structure de cette table est la suivante :

b) La structure de la table NGR_SIG

```
Class NGR_SIG {
    Friend class PF_AODV;
 Public:
    NGR_SIGe (u_int32_t ng) { ng_addr = ng; }
Protected:
    LIST_ENTRY (NGR_SIG)  ng_link;
    nsaddr_t      ng_addr;  // @ du nœud
    int           Val-SS ;   // la valeur de la puissance pour ce nœud
};
```

Pour la gestion de la table NGR_SIG, deux fonctions (Ajout et Enlever) sont implémentées.

NGR-Insert (voisins du nœud, Val-SS) : permet d'insérer les adresses des voisins d'un nœud du chemin actif avec la valeur de puissance (Val-SS).

NGR-Delete (voisin du nœud) : permet de supprimer les voisins du nœud d'un chemin actif (une entrée dans NGR_SIG) dont Val-SS devient inférieur à la valeur "Seuil".

Parmi les champs importants à véhiculer dans le paquet H_Liaison on trouve : @ IP nœud, @ IP nœud suivant stable, nombre de saut pour atteindre la destination, numéro de séquence, temps d'expiration du lien.

Le paquet H_Liaison est envoyé entre voisins et chacun de son coté (les extrémités ou les nœuds adjacents) doit modifier sa table de routage surtout pour le paramètre *Next Hop* et *Life time*.

En général chaque nœud doit :

- Vérifier la puissance du signal (Val_SS) des nœuds en aval et en amont faisant partie du chemin actif, à chaque intervalle de temps "th".
- Détecter si le lien avec le nœud suivant ou précédent d'un lien actif, sa puissance est en baisse par rapport à la valeur "Seuil" pour lancer la procédure de "Liaison".

Si le nœud xi est proche de xs alors la liaison est initiée par xs (coupure probable du segment xi—xd).

Pour un prédécesseur (xp) de xi, trouver le meilleur voisin xh tel que xh est aussi voisin de xi (c'est-à-dire prendre seulement les voisins du coté de xi. Le nœud xh est considéré comme **nœud de liaison**. Si xh à une route à xd, alors le segment [xs---xh---xd] est reconstruit sinon une réparation locale comme dans l'AODV est initiée.

Si le nœud xi est proche de xd alors la liaison est initiée par xs (coupure probable du segment xs—xi) alors xs est considéré comme **nœud de liaison**.

La procédure Evaluer_Lien est exécutée par chaque nœud d'un lien actif à la réception d'un H_Ordinaire.

c) Procédure Evaluer_Lien

1. Comparer "Val_SS" au "Seuil"
2. Si elle est supérieure au seuil, le lien est toujours stable (i.e. ne rien faire). Donc aller à 4 (Fin).
3. Sinon lancer la procédure de "Liaison" avec un paquet "H_Liaison".
4. Fin.

d) Procédure Liaison (Paquet : H_Liaison, nœud : xi, nombre d'exécution)

1. Trouver le meilleur voisin (xp) (la plus grande Val-SS de (xi))
2. S'il n'existe pas, une réparation locale comme dans l'AODV sera initiée si coupure aura lieu.
3. S'il existe, c'est le nœud H (nœud de liaison), diffuser un paquet "H_Liaison" entre P et H puis modifier le saut suivant (Next hop) de P pour D par H.
4. Vérifier si H à une route vers D
5. Si c'est le cas, le segment est reconstruit, aller à 7 (Fin)
6. Sinon lancer de nouveau la procédure "Liaison" pour le nœud H (pour une seule exécution).
7. Fin

NB : le nombre d'exécution est au maximum égal à 2 puisque on s'intéresse à une reconstruction de deux (02) segments au plus (c'est-à-dire un TTL= 2, décrémenté à chaque exécution).

V.3.5. La Contributions au niveau de la couche MAC

Les mécanismes de la norme IEEE 802.11 rencontrent un grand défi pour le support de la QdS puisque la DFC telle que définit actuellement ne peut supporter une différentiation pour les différentes classes de trafic (temps réel, multimédia, etc…) et d'un autre coté la PCF peut fournir seulement un support pour trafic temps réel. Par conséquent la norme IEEE 802.11 est conçue pour un service best effort et non pour les applications de multimédia et/ou temps réel avec des exigences de QdS.

Le protocole de routage avec QdS, détermine les routes qui répondent aux exigences de QdS telles que le délai et la bande passante sans ***assurer leurs réservations***. Dans une méthode d'accès au support avec contention comme CSMA/CA, la réservation est très difficile à cause de la présence de collisions. Au niveau de la couche MAC, Les paquets subissant des collisions lors d'un accès compétitif au canal doivent être retransmis. Ces retransmissions consomment assez de bande passante et réduisent le délai de transfert de bout en bout des paquets de données. Pour faire face à ce phénomène, et de manière générale les modifications apportées à la méthode DCF portent soit sur la fonction d'incrémentation de la valeur du Backoff et par conséquent la fenêtre de contention (CW), soit sur la variation de la valeur du SIFS et par conséquent DIFS (i.e. DIFS=2*SIFS) et soit ajuster la taille maximale de la trame de données.

Notre troisième contribution tente de proposer une nouvelle forme d'incrémentation du Backoff qui conduit à des meilleures valeurs de la fenêtre de contention (CW) ce qui engendre moins de collisions et par conséquent moins de retransmissions.

V.3.5.1. Modification des valeurs de CW

Introduction

Le temps de Backoff est un nombre de slots time, il est calculé de façon aléatoire. Ce compteur est utilisé pour différer l'accès au canal (i.e. temps que les stations doivent attendre avant d'accéder au canal). Tant que le médium est occupé, les stations décrémentent leurs valeurs respectives du temps de Backoff. Dès que, la valeur de Backoff pour une station donnée atteint zéro, cette dernière commence à émettre quelque soit la nature du canal (occupé ou libre).

Le choix de la valeur du Backoff à un effet direct sur le nombre de collisions qui peuvent se produire.

Dans cette section, on va présenter la dernière proposition au niveau de la couche MAC.

Modification proposée [13][20][146][147]:

Cette proposition porte sur la manière d'incrémenter la fenêtre de contention (**CW**) dans le mode de fonctionnement de la procédure d'accès au médium DCF avec RTS et CTS dans la couche MAC pour améliorer certains attributs de la QdS comme la perte, le délai et le débit.

Dans la procédure DCF, Le mécanisme de Backoff limite les risques de collision mais ne les supprime pas complètement. Aussi, si une collision se produit quand même (détectée grâce à l'absence d'acquittement), un nouveau Backoff va être tiré au hasard. Mais à chaque collision, la taille de la fenêtre va doubler afin de diminuer les chances que de telles collisions se répètent.

Les valeurs de **CW** autorisées ne sont que des puissances de 2 moins 1(i.e. un décalage vers la gauche d'une position) prisent dans l'intervalle [CW_{Min} et CW_{Max}] (valeurs limites définies par la norme) [13].

Pour cette proposition, on va tester deux fonctions pour incrémenter la valeur de **CW** par deux manières (types de décalage) qu'on notera fonction 1 et fonction 2.

Le pas d'incrémentation a une influence sur la taille de la CW. Les grandes valeurs pour ce pas mènent à de large CW et par conséquent un grand temps de Backoff ce qui augmente le temps d'attente et génère un volume de contrôle important, par contre, les petites valeurs conduisent à de petite CW ce qui augmentent le nombre de collisions. Pour les ces deux raisons évoquées ci dessus, on essaye de combiner les deux idées.

Description de la proposition

Le temps de backoff pour DCF de base est ßi * (SlotTime) où ßi est donnée par la formule mathématique suivante : ßi = 2^{i+k} - 1 où i (initialement égale à 1) est le nombre de tentatives de retransmission, k dépend du type de la couche PHY et SlotTime est fonction des paramètres de la couche physique. Il y a une limite supérieure pour i au-dessus duquel l'intervalle aléatoire (CW_{Max}) reste le même. Quand un paquet est transmis avec succès, la valeur de CW est remise à CW_{Min}.

Dans la standard 802.11, les valeurs choisies sont CWmin = 31, Cwmax = 1023 et pour k on prend la valeur 4, de telle sorte ßi devient (ßi = 2^{i+4} - 1) et i prend les valeurs de 1 à 6 (i = {1, 2, 3, 4, 5, 6}). Dans ce cas ßi est une fonction de décalage de 1 bit à qui on ajoute la valeur 1 (décalage de 1 bit +1). Ainsi, après chaque collision, les valeurs possibles de CW sont: {31, 63, 127, 255, 511, 1023} (Figure 5. 33 (a)).

La fonction 1 est basée sur un décalage de deux bits auxquels on ajoute la valeur 3, où le nombre 3 est utilisé pour remplacer les deux bits égaux à zéro après l'opération de décalage (décalage par 2 bits +3) et ßi devient (ßi =). Le nombre de tentatives de retransmission après calcul est égale à (04) (i = {1, 2, 3,4}), si i est supérieur à 3, on met CW = 1023 où bien on ajoute 1023. Ainsi, après chaque collision, les valeurs possibles de CW sont : {31, 127, 511, 1023} (Figure 5. 33 (b)).

La fonction 2 est basée sur un décalage de trois bits auxquels on ajoute la valeur 7, où le nombre 7 est utilisé pour remplacer les trois bits égaux à zéro après l'opération de décalage (décalage par 3 bits +7) et ßi devient (ßi =). Le nombre de tentatives de retransmission après calcul est égale à (03) (i = {0, 1, 2}), si i est plus grand que 2, on met CW = 1023 où bien on ajoute 1023. Ainsi, après chaque collision, les valeurs possibles de CW sont: {31, 255, 1023} (Figure 5. 33 (c)).

Les intervalles des temps de Backoff des deux dernières fonctions, sont inférieurs à ceux de la fonction originale ; ceci permet de déclarer le cas d'échec d'une transmission le plus tôt (i.e. aller plus vite vers les valeurs extrêmes de CW) si le phénomène de collision persiste.

Les algorithmes des trois fonctions sont décrits ci-dessous [147].

Fonction de base: décalage par 1 bits +1

```
Procédure  inc_cw ( )
   Début
      CW = (CW << 1) +1      // << : décalage de la valeur de CW d'un bit //
          Si  (CW >  CWMax ) //dèsque la valeur de CW dépasse CWMax, elle est remis à CWMin //
              CW = CWMin
   Fin
```

Fonction 1 : décalage par 2 bits +3

```
Procédure  inc_cw ( )
   Début
      CW = (CW << 2) +3   // << : décalage de la valeur de CW de deux bits //
          Si  (CW >  CWMax ) // dèsque la valeur de CW dépasse CWMax, elle est remis à CWMin //
              CW = CWMin
   Fin
```

Fonction 2: décalage par 3 bits +7

```
Procédure  inc_cw ( )
   Début
      CW = (CW << 3) +7    // << : décalage de la valeur de CW de trois bits //
          Si  (CW >  CWMax )  // dèsque la valeur de CW dépasse CWMax, elle est remis à CWMin //
              CW = CWMin
   Fin
```

La figure 5.32 récapitule les valeurs possibles de CW pour les trois fonctions.

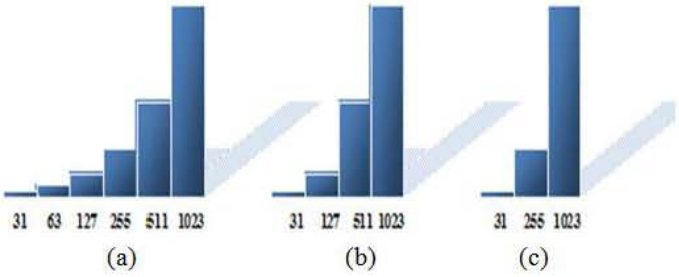


Figure 5. 33: Valeurs possibles de CW pour les 3 fonctions

V.4 Conclusion

Dans ce chapitre, nous avons décrit et motivé notre proposition que ce soit au niveau routage ou au niveau MAC. La première proposition porte sur la modification du protocole AODV et trois cas on été discutés (AODV-SR, M-AODV et PF-AODV). La seconde modification est au niveau de la couche MAC. Elle porte sur une nouvelle forme de variation sur un des paramètres relatifs à cette couche (la valeur du temps de Backoff ou la taille de la fenêtre de contention). Afin de valider ces propositions et montrer leurs apports en QdS, une étude à base de simulation sous le simulateur NS2 (Network Simulator) sera faite dans le chapitre suivant.

Chapitre 6 : Simulation et Evaluation de performances

VI.1 Introduction

La simulation est un des outils les plus utilisés pour évaluer les performances de systèmes avant leurs implémentations réelles. Le but de ces simulations est de pouvoir tester les différentes propositions dans différents contextes afin montrer leurs apports en QdS dans un réseau ad hoc. Ces simulations sont conduites sous Network Simulator (NS2) (version 2.31) [148].

Dans la suite de ce chapitre, après une brève description de NS2, nous présenterons les contraintes ou les conditions sous lesquelles sont réalisées nos simulations. En suite, les paramètres utiles pour évaluer les protocoles sont décrits. Le reste du chapitre sera consacré à l'analyse des résultats trouvés pour les différentes propositions et nous terminerons par une conclusion.

VI.2. Présentation de Network Simulator 2

Network Simulator (NS2) est développé dans un but de recherche. Il est aujourd'hui le simulateur de réseau probablement le plus utilisé par la communauté scientifique. Il fournit un environnement assez détaillé permettant de réaliser des simulations de TCP, du routage et des protocoles multicast aussi bien sur des liens filaires que sans fil. C'est un simulateur open source, il est très utilisé dans les environnements ad hoc.

NS2 est un simulateur à événements discrets, orienté objet, fruit de la collaboration entre les universités de Berkeley, de Southern California et Xerox PARC dans le cadre du projet VINT (*Virtual Inter Network Testbed*) soutenu par le DARPA (*Defense Advanced Research Projects Agency*).

Il est écrit en C++ et en OTCL (Figure 6. 1). Il utilise le langage oriente objet OTCL dérivé de TCL pour la description des conditions de simulation sous forme de script (i.e. configure le système de communication). Dans le script l'utilisateur fournit la topologie du réseau, les caractéristiques des liens physiques, les protocoles utilisés, le type de trafic généré par les sources, les événements, etc.

NS2 est fourni avec divers outils d'analyse complémentaires eux-mêmes écrits en C/C++ ou TCL/Tk.

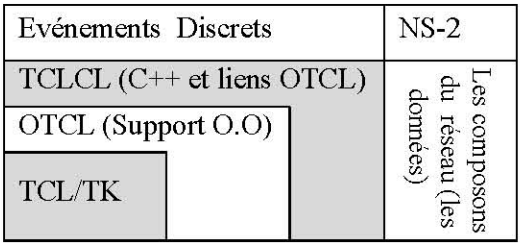


Figure 6. 1: Structure du simulateur NS 2.

Il permet d'exécuter tous types de scénarios sur des topologies définies par l'utilisateur. Le réseau est modélisé par ses sources de trafic (applications), ses protocoles (UDP, TCP), ses routeurs (avec leurs files d'attente) et les liens qui les relient.

Le résultat d'une simulation est un fichier trace de type texte (d'extension .tr) contenant tous les événements de la simulation. Un traitement ultérieur de ce fichier permet d'en soustraire l'information souhaitée grâce à des opérations de filtrage par des outils comme Perl, Awk [149], etc….

Par ailleurs, le simulateur permet la création d'un fichier d'animation (d'extension .nam), permettant de visualiser la simulation sur l'interface graphique NAM. Cette visualisation fournit une représentation du graphe du réseau sur laquelle on peut voir les paquets circuler, suivre le niveau des files d'attente et observer le débit courant des liaisons. Une représentation graphique (courbes, histogrammes) des résultats filtrés est possible grâce à un module xgraph.

VI.3. Contraintes de simulation

Dans un réseau ad hoc, les nœuds sont *mobiles* ce qui entraine un changement fréquent de la topologie. De plus les nœuds disposent d'une quantité limitée *d'énergie* qu'il est impératif de la gérer au mieux le plus longtemps possible. La *densité* (le nombre moyen de nœuds voisins par mètre carré) et la *taille* (l'espace) en mètre du réseau ont tous un impact sur les performances du réseau ad hoc.

VI.3.1. La mobilité

Dès qu'une route est établie, elle sera maintenue pour toute la période de transmission. Mais comme les nœuds sont mobiles, ils ne seront plus dans la portée des voisins et par conséquent la ou les routes dont ils font parties deviennent invalides. On est amené dans ce cas à relancer le processus de découverte de routes, ce qui génère un volume additionnel des paquets de contrôle.

L'objectif ici est de tester le comportement du protocole de routage AODV et ses variantes (reconstruction à la source : AODV-SR, multi chemins : M-AODV et par prédiction de liens : PF-AODV) sous la contrainte de mobilité afin de savoir laquelle des trois versions minimise le volume des informations de contrôle générées, pour l'établissement ou la reconstruction des routes, le délai de transit des paquets de données, et au même temps la perte des paquets des données transmises.

VI.3.2. Energie

En générale la consommation de l'énergie des nœuds est proportionnelle au nombre de paquets traités et de la nature du traitement effectué (émission/réception). Il est à noter que l'émission d'un paquet demande plus d'énergie que la réception. On s'intéresse ici, à savoir comment la quantité d'énergie est consommée dans le temps et quels sont les paramètres qui ont un effet direct sur la consommation excessive de cette ressource.

VI.3.3 Le passage à l'échelle

Actuellement la plupart des protocoles de routage dans les réseaux ad hoc souffrent du problème de passage à l'échelle. L'objectif ici est de savoir si cette contrainte est bien respectée et que le protocole garde ses performances face à un changement de l'échelle.

VI.4. Paramètres de simulations

Afin de tester les performances d'un système dans une simulation, il est intéressent de déterminer les paramètres ou critères qui peuvent plus au moins nous renseignés sur ses performances.

Parmi les paramètres qui caractérisent la QdS et les plus utilisés en simulation pour évaluer les performances de réseaux on trouve :

VI.4.1. La perte des paquets

En effet la perte des paquets est un élément crucial pour l'évaluation des performances d'un protocole de routage. Assurer un transfert à zéro perte est une chose espérée, mais est ce que cela est possible dans les réseaux ad hoc ?

Un protocole est efficace, s'il capable de minimiser au maximum la perte de paquets quelque soit la condition à laquelle il est confronté (un grand ou petit nombre de nœuds (densité), une petite ou large échelle, une forte ou une faible mobilité…etc.).

VI.4.2. La charge de contrôle

Est un élément déterminant pour mesurer la performance d'un protocole. Plus le volume de contrôle est volumineux, plus les performances se dégradent et la bande passante est plus utilisée par les paquets de contrôle que par les données utiles. La mesure d'un tel paramètre peut justifier le choix d'utilisation de tel ou tel protocole.

VI.4.3. Le débit

Le débit (Throughput) ou par abus de langage bande passante indique le taux de transfert de données. Il est mesuré en Kbps. Plus le débit est élevé plus il y a une meilleure exploitation du réseau qui est une chose souhaitée.

VI.4.4. Le délai moyen de bout en bout

Le délai moyen de bout en bout (e2e : average end-to-end delay) exprime le temps mis entre l'émission et la réception d'un paquet de données. Plus le délai est court plus le réseau est sollicité.

VI.4.5. Le délai de sélection de route

C'est le temps mesuré en tant que la différence entre l'instant où l'initiateur d'une connexion reçoit la confirmation de route et l'instant où il avait émis la requête de route correspondante. Ce paramètre à un effet direct sur le délai de transmission des données.

VI.4.6. La latence

Elle est définie par le délai de transfert de bout en bout d'un paquet d'un flux. Les applications interactives ont une latence maximale tolérable. Si un paquet subit un retard important, au-delà de la valeur tolérable, les données qu'il contient deviennent inutiles pour l'application.

Le dernier paramètre est la valeur de ***l'énergie consommée*** par chaque nœud que ce soit dans tous le réseau ou seulement dans la phase de routage pour tester son impact sur la performance du réseau.

VI.5. Paramètres à évaluer

VI.5.1. Taux de paquets livrés avec succès (PDR)

Ce paramètre représente le taux des paquets livrés à leurs destinations par rapport aux paquets émis dans le réseau. Il exprime si le réseau est efficace ou non, et se calcule selon la formule suivante :

$$Packet\ Delivery\ Ratio:\ PDR = \frac{\sum(\text{Paquets Reçus})}{\sum(\text{Paquets Emis})} \quad (\%)$$

VI.5.2. Trafic de contrôle (Trafic overhead)

Ce paramètre nous informe sur la quantité des paquets de contrôle générés par le protocole pour la recherche, l'établissement et le maintien de routes. Il montre donc à quel point un protocole consomme de la bande passante avec ses propres messages de routage.

$$Traffic\ Overhead = \sum(\text{Paquets de contrôle})$$

VI.5.3. Taux de trafic de contrôle

(Normalized Routing Load Ratio : NRL ou Normalized Overhead Load Ratio : NOL) : indique le taux des paquets de contrôle émis par rapport au nombre de paquets de données reçus. Il exprime la surcharge du réseau.

$$NRL = \frac{\sum(paquets\ de\ contrôle)}{\sum(paquets\ reçus)}\ (\%)$$

VI.5.4. Délai moyen de bout en bout (e2e)

Indique le temps mis pour que les paquets arrivent à destination. Il est utile pour les applications qui exigent un certain délai. Il est donné par la formule suivante:

$$Average\ end\ to\ end\ Packet\ Delay = \frac{\sum(T_R(i) - T_S(i))}{\sum(Paquets\ Reçus)} \quad (s)$$

Où

T_R **(i)** : instant où le paquet de donnée " ***i*** " est reçu par l'agent de transport destination.

T_S **(i)** : instant où le paquet de donnée " ***i*** " est émis par l'agent de transport source.

VI.5.5. Débit utile (Throughput)

C'est le débit total en réception. Il mesure le nombre de paquets transmis avec succès à leur destination dans un intervalle de temps donné (exprimé en octets/seconde). Il se calcule par :

$$Débit\ du\ réseau = \frac{\sum(Bit\ Reçus)}{\sum(Délai\ de\ bout\ en\ bout)} \quad (octet/s)$$

VI.5.6. Energie de réseau

Elle indique la quantité d'énergie des nœuds du réseau à un instant donné (exprimée en joules) et se calcule de la façon suivante :

$$Energie\ de\ réseau(t) = \sum_{i=1}^{n} Energie(N)_i \quad (joules)$$

Où n : *nombre* de nœuds, t : temps.

N_i : Le Nœud numéro " ***i*** " parmi les nœuds de réseau.

On peut mesurer l'énergie des nœuds qui participeront au routage par la formule suivante :

$$Energie\ des\ nœuds\ de\ routage\ (t) = \sum_{i=1}^{m} Energie(M)_i \quad (joules)$$

Où m: nombre des nœuds qui font le routage en fonction de temps t.

M_i : Le nœud numéro " ***i*** " parmi les nœuds de routage.

VI.6. Modèle de génération de mouvements sous NS2

Le simulateur NS2 offre la possibilité de générer les mouvements des nœuds dans le temps sur une topologie bien déterminée et avec une vitesse bien choisie avec la commande « setdest ».

Le format de la commande est :

./setdest -n <num_of_nodes> -p <pausetime> -s <maxspeed> -t <simtime> -x <maxx> -y <maxy> > <outdir>/<scenario-file>

Le temps de pause est inversement proportionnel à la mobilité, c'est-à-dire si le temps de pause est petit alors la mobilité est élevée. Cette dernière est faible pour un temps de pause plus grand.

Pour un exemple de 10 nœuds, une vitesse de 5m/s, un temps de pause d'une seconde et sur une topologie de 1500 x 1500 m^2 on écrit :

setdest1 -v 1 -n 10 -p 1 -M 5 -t 200 -x 1500 -y 1500 > scenario-file

VI.7. Modèle de génération de trafic sous NS2

Il permet de générer le trafic (CBR ou autres) de façon aléatoire pour un nombre de nœuds et avec un débit choisi selon le besoin de l'application avec le script « cbrgen.tcl »

Le format du script est :

ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate rate] > <traffic-file>

Pour un exemple de 10 nœuds, une vitesse de 1m/s, et 20 connections et un débit de 0.5 octets/s, on écrit :

ns cbrgen.tcl -type cbr -nn 10 -seed 1 -mc 20 -rate 0.5 > traffic-cbr

NB : les scenarios de simulation et le trafic utilisé de nos simulations sont générés respectivement à l'aide de cette commande et ce script.

VI.8. Cas des contributions au niveau routage : courbes et discussions

VI.8.1. AODV-SR

VI.8.1.1. Introduction

Le contexte de simulation est indiqué dans le tableau ci-dessous (Tableau 6. 1) :

Paramètres	Valeur
Nombre de nœud	40
Topologie du réseau	2200x1500
Temps de simulation	1200
Taille du buffer	50
Taille du paquet	512
Intervalle de transmission	0.02s
Transmission effectuée	entre les nœuds 1 et 2
Modèle de mobilité	Aléatoire
Protocoles simulés	AODV, AODV-SR

Tableau 6. 1: Paramètres de simulation

VI.7.1.2. Résultats et discussions

Le tableau ci-dessous (Tableau 6. 2) récapitule le nombre de paquets émis, reçus, perdus, le taux des paquets livrés avec succès (PDR) pour une faible et forte mobilité.

Paquets	Emis		Reçus		Perdus		PDR	
Mobilité	Faible	Forte	Faible	Forte	Faible	Forte	Faible	Forte
AODV	23963	23897	22825	21904	1122	1991	0,95251	0,9166
AODV-SR	23946	23877	23395	22175	554	1697	0,97699	0,928718

Tableau 6. 2: Récapitulatif des paquets (émis, reçus, perdus)

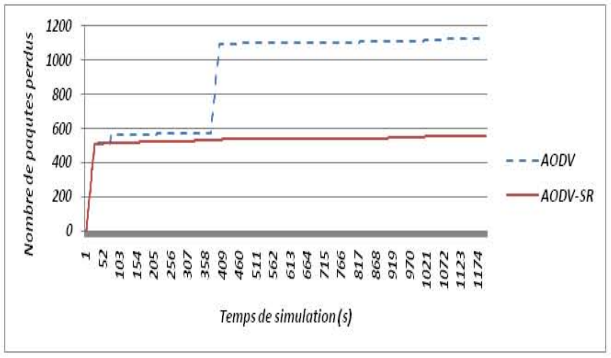


Figure 6. 2 : Perte de paquets Vs Faible mobilité

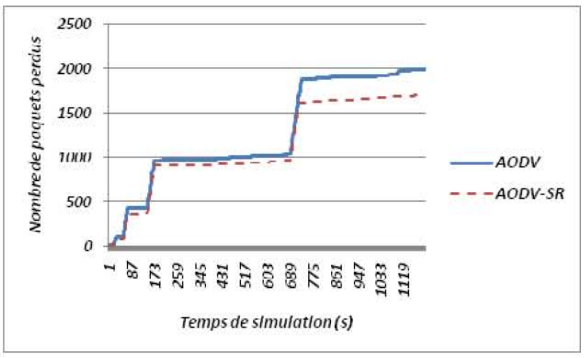


Figure 6. 3 : Perte de paquets Vs Forte mobilité

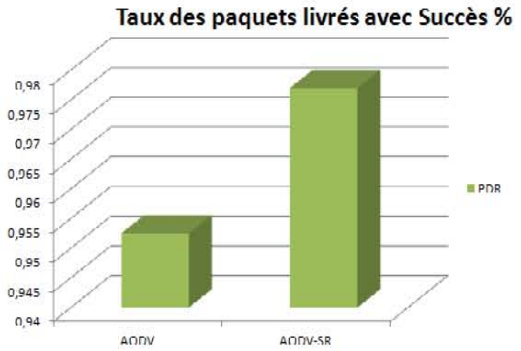


Figure 6. 4: PDR Vs Faible mobilité

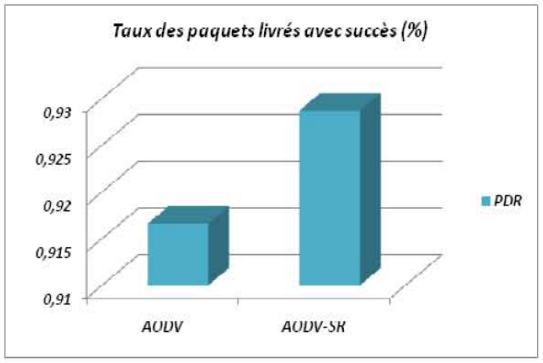


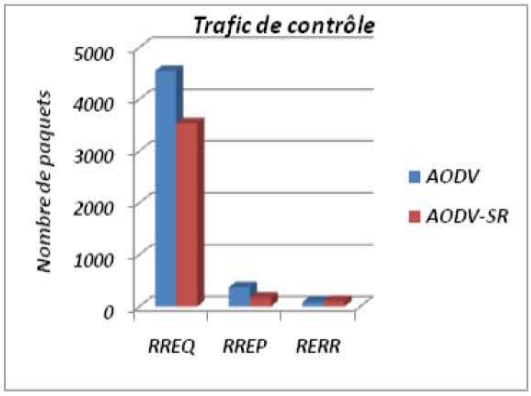
Figure 6. 5: PDR Vs Forte mobilité

On constate d'après les Figure 6. 2 & Figure 6. 2, que l'utilisation de la version AODV-SR (reconstruction à la source) réduit grandement la perte de paquets que la version de base AODV pour une forte mobilité et elle est presque de la moitié pour une faible mobilité. La même remarque est à faire pour le taux des paquets livrés avec succès (PDR) (Figure 6. 4 &Figure 6. 4).

Le Table 6. 3, regroupe le nombre des différents paquets de routage (i.e. de contrôle) utilisés par les deux variantes (AODV & AODV-SR) pour une forte et faible mobilité.

Paquets	RREQ		RREP		RERR		Total	
Mobilité	Faible	Forte	Faible	Forte	Faible	Forte	Faible	Forte
AODV	4546	11566	370	1559	86	398	5002	13523
AODV-SR	3532	9270	166	632	106	470	3804	10372

Table 6. 3: Paquets de contrôle



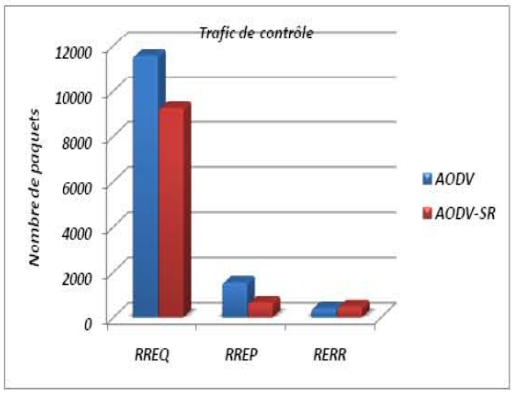


Figure 6. 6 : (à gauche): Paquets de contrôle pour AODV & AODV-SR Vs faible mobilité
Figure 6. 7 : (à droite): Paquets de contrôle pour AODV & AODV-SR Vs forte mobilité

Eliminer la phase de réparation locale fait diminuer le volume de paquets de contrôle généré durant cette procédure et par conséquent l'AODV-SR rivalise l'AODV original dans la majorité des cas et génère moins de paquets de contrôles que la version de base (originale) (Figure 6. 6 & Figure 6. 7). Notons que pour une mobilité forte, le nombre de paquets de contrôle est plus important que pour une faible mobilité puisque ceci

est dû aux fréquentes déconnexions. Par contre le nombre de paquets RRER est plus grand dans la version proposée car on a ajouté une action qui permet d'aviser le plus rapidement la source pour qu'elle stoppe ses transmissions via un paquet RRER.
Le Tableau 6. 4, regroupe les valeurs du délai moyen et total pour les deux variantes dans un contexte de faible et forte mobilité.

Protocoles	AODV		AODV-SR	
Mobilité	Faible	Forte	Faible	Forte
Délai_total	1084,52	1721,44	662,64	1001,83
Délai moyen	0,0475147	0,07859	0,028324	0,0451783

Tableau 6. 4: Délai moyen et Délai Total

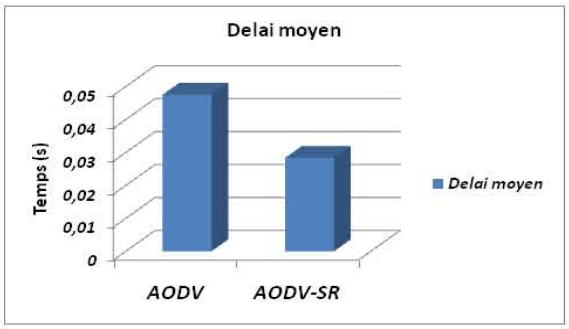


Figure 6. 8: Délai moyen Vs faible mobilité

Figure 6. 9: Délai moyen Vs forte mobilité

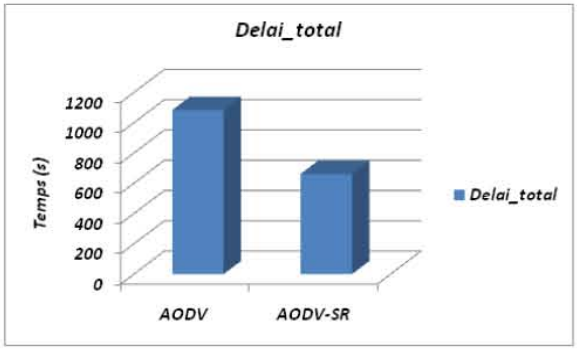


Figure 6. 10: Délai total Vs Faible mobilité

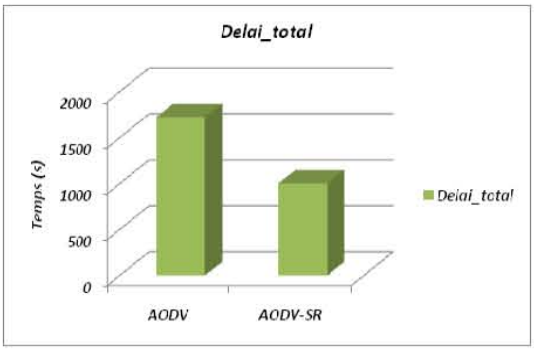


Figure 6. 11: Délai total Vs Forte mobilité

La même remarque est à faire pour le délai de transit (moyen ou total) (Figure 6. 8, Figure 6. 8, Figure 6. 10 & Figure 6. 10Figure 6. 10) c'est-à-dire que la version modifiée est meilleure que la version de base. L'ordre de grandeur du délai dépend de la mobilité, c'est-à-dire dans un cas de forte mobilité, les déconnexions sont fréquentes ce qui causent la perte de données, ce qui entraine leurs retransmissions et par conséquent fait augmenter le délai de transit.

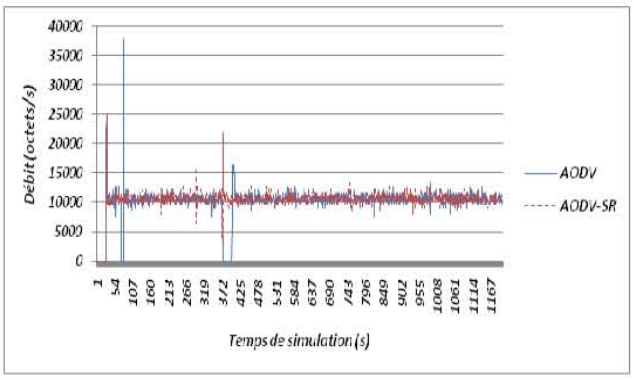


Figure 6. 12 : Débit Vs Faible mobilité

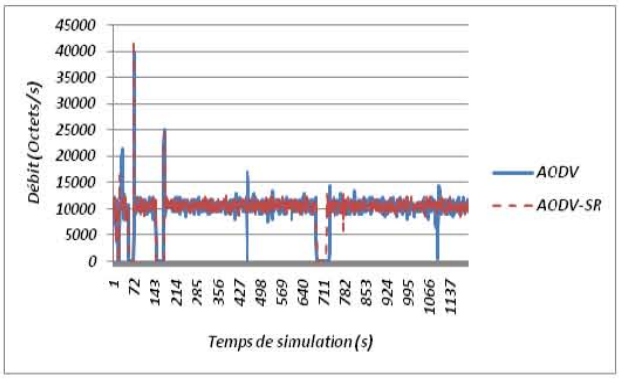


Figure 6. 13 : Débit Vs Forte mobilité

Un taux élevé de paquets livrés avec succès implique moins de retransmission c'est-à-dire peu de perte de paquets. Un volume de trafic de contrôle réduit implique une gestion rationnelle de la bande passante et par

conséquent le débit est plus stable pour l'AODV-SR que ce soit pour une forte ou faible mobilité (Figure 6. 12 &Figure 6. 12).

VI.7.1.3. Conclusion

La modification proposée vise à alléger la version de base de l'AODV, en éliminant l'opération de réparation locale et tout est délégué dans ce cas à la source. Les mesures conduites par simulation (i.e. une étude comparative entre la version originale du protocole AODV et la version modifiée AODV-SR) sous NS2 ont montrés que l'AODV-SR améliore certain paramètres de QdS dans les réseaux mobiles ad hoc selon les conditions proposées (forte et faible mobilité).

VI.7.2. M-AODV

VI.7.2.1. Introduction

Afin d'étudier et d'analyser le fonctionnement et le comportement du protocole M-AODV; nous avons pris le contexte de simulation qui consiste en 20 nœuds dans une région de 800x600m^2. La portée de transmission est de 250m pour un espace idéale (sans obstacle) avec le modèle de mobilité Random Way Point (RWP). Les nœuds se déplacent à une vitesse maximale de 5m/s (vitesse moyenne). Un trafic CBR de 512 octets selon le protocole UDP généré automatiquement d'une manière aléatoire au moyen du script cbrgen.tcl de NS2. Les différents scénarios de mobilité sont aussi produits avec le programme Setdest de NS2.

La durée de la simulation est de 120s pour l'ensemble des jeux d'essais et de 300s pour l'évaluation de la consommation de l'énergie ou chaque nœud à initialement une énergie de 100 joules.

VI.7.2.2. Exemples de scripts utilisés dans la simulation

VI.7.2.2.1. Selon le temps de pause

- ***Cas du trafic CBR:***

Notation «cbr xp » ou x est le nombre de nœuds et p les différentes valeurs de temps de pause utilisées.

Le script est le suivant: ns cbrgen.tcl -type cbr -nn 20 -seed 5 -mc 20 -rate 4.0 > cbr20-20p

Ici on utilise un seul modèle de trafic avec nn = 20 nœuds, une vitesse moyenne (seed=5 m/s) et un débit de transmission constant (rate = 4.0)

- ***Cas des mouvements de nœuds (scènes) :***

Notation « scen xp » ou x est le nombre de nœuds et p un temps de pause.

Le script est le suivant: setdest1 -v 1 -n 20 -p z -M 5 -t 120 -x 800 -y 600 > scen20-z

Avec z = 0, 10, 50, 100, 150 et 200 d'une mobilité très élevée jusqu'à une mobilité presque nulle (z = 200).

Le nombre de nœuds est n = 20, la vitesse moyenne est M = 5 m/s, le temps de simulation (t = 120s) et x, y (800x600) l'espace utilisé.

VI.7.2.2.2. Selon le nombre de nœuds

- ***Cas du trafic CBR:***

Notation «cbr x » ou x indique les différentes valeurs utilisées pour le nombre de nœuds.

Le script est le suivant: ns cbrgen.tcl -type cbr -nn x -seed 1 -mc 20 -rate 0.5 > cbr x

Avec pour valeurs de x : 10, 20, 50, 80, 100 et 150.

On utilise une vitesse faible (seed=1 m/s) et un débit de transmission constant (rate =0.5)

- ***Cas des mouvements de nœuds (scènes) :***

Notation « scen x » ou x indique les différentes valeurs utilisées pour le nombre de nœuds.

Le script est le suivant: setdest1 -v 1 -n x -p 10 -M 5 -t 120 -x 800 -y 600 > scen x

Avec une vitesse moyenne (M = 5 m/s), un temps de simulation (t =120s) et x, y (800x600) l'espace utilisé.

Les valeurs de x sont identiques à ceux du trafic CBR.

VI.7.2.2.3. Selon le débit de transmission

- ***Cas du trafic CBR:***

Notation « cbr x r z » ou x est le nombre de nœuds, r pour dire rate (débit) et z les différents débits de transmission utilisés. On prend pour z les valeurs : 10.0, 8.0, 4.0, 1.0, 0.5 et 0.2.

Le script est le suivant : ns cbrgen.tcl -type cbr -nn 20 -seed 1 -mc 20 -rate x > cbr20r-z

Dans ce cas, on prend nn = 20 nœuds, une vitesse faible (seed = 1m/s) et mc = 20 connexions.

- ***Cas des mouvements de nœuds (scènes):***

Notation « scen x p z » ou x est le nombre de nœuds, p le temps de pause et z les différents débits de transmission utilisés.

Le script est le suivant : setdest1 -v 1 -n 20 -p 10 -M 5 -t 120 -x 800 -y 600 > scen20r-5

Le nombre de nœuds (n=20), le temps de pause (p=10), la vitesse (M=5), le temps de simulation (t=120) et x, y (800x600) l'espace utilisé.

VI.7.2.2.4. Selon le nombre de connexions

- ***Cas du trafic CBR:***

Notation «cbr xcy » ou x est le nombre de nœuds, c pour dire connexion et y les différents nombre de connexions utilisées.

Le script est le suivant : ns cbrgen.tcl -type cbr -nn 20 -seed 5 -mc y -rate 4.0 > cbr20-cy

Avec pour valeurs de y : 5, 10, 15, 20, 25 et 30.

Le nombre de nœuds (nn=20), une vitesse moyenne (seed=5 m/s) et un débit de transmission (rate = 4.0).

- ***Cas des mouvements de nœuds (scènes) :***

Notation « scen xpc » ou x est le nombre de nœuds, c pour dire connexion et p le temps de pause.

Le script est le suivant: setdest1 -v 1 -n 20 -p 10 -M 1 -t 120 -x 800 -y 600 > scen20-10c

Le nombre de nœuds (n=20), un temps de pause constant (p = 10), une vitesse faible (M=1m/s), un temps de simulation (t=120s) et x, y (800x600) l'espace utilisé. Les valeurs de c sont identiques à ceux de y pour le cas du trafic CBR.

VI.7.2.3. Courbes & discussions

Les paramètres qu'on désire évaluer par simulation sous différents contextes (mobilité, densité, etc..) sont:

- Le débit (en kbps) : indique le taux de transfert de données.
- Le délai moyen de bout en bout (e2e): exprime le temps mis entre l'émission et la réception d'un paquet.
- Le PDR : le taux de paquets livrés avec succès sur le nombre total des paquets transmis.
- Le NRL : le taux des paquets de contrôles émis par rapport au nombre de paquets reçus, sa valeur exprime la surcharge du réseau.
- Le taux des paquets perdus c'est-à-dire le taux entre paquets émis et ceux reçus avec succès exprime la fiabilité du réseau, et
- L'énergie consommée par chaque nœud soit dans tout le réseau ou seulement dans la phase de routage pour voir lequel des deux algorithmes gère au mieux cette ressource.

Dans ce qui suit, on étudiera l'impact de la mobilité (respectivement le nombre de nœuds) sur ces paramètres en passant d'une mobilité très élevée c'est-à-dire un temps de pause égale à zéro jusqu'à un état de non mobilité pour un temps de pause de 200s et plus. (Respectivement on faisant varier la taille du réseau du moins dense de 10 nœuds jusqu'à un réseau plus dense de 100 nœuds) (Voir section 6.7.2.2).

Courbes :

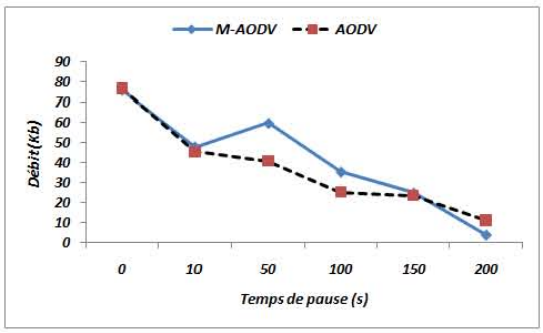


Figure 6. 14: Débit Vs Temps de Pause

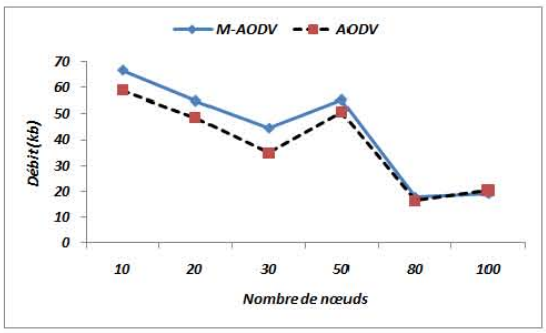
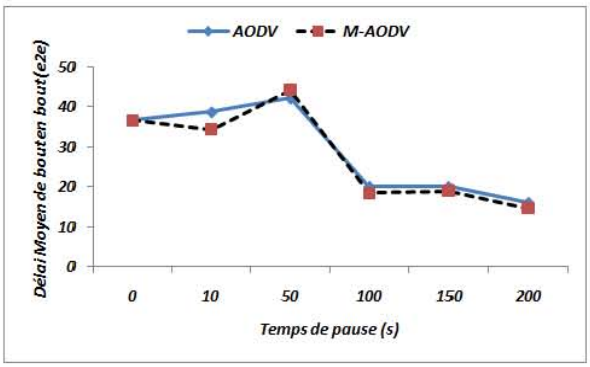


Figure 6. 15: Débit Vs Nombre de nœuds



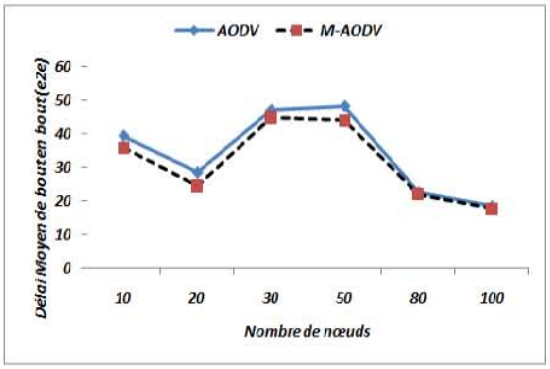
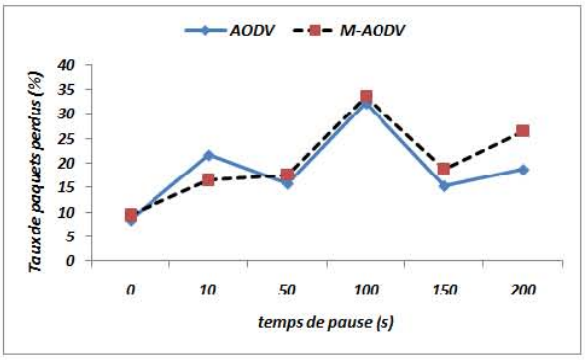


Figure 6. 16 (à gauche): Délai Moyen de bout en bout Vs Temps de Pause
Figure 6. 17 (à droite): Délai Moyen de bout en bout Vs Nombre de nœuds



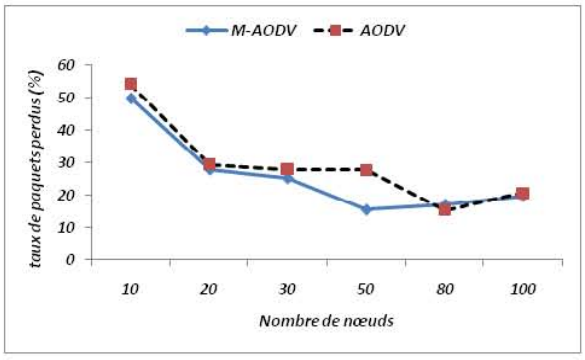
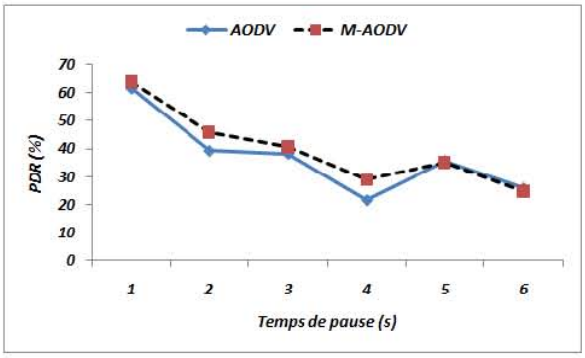


Figure 6. 18 (à gauche) : Taux de paquets perdus Vs Temps de Pause
Figure 6. 19 (à droite) : Taux de paquets perdus Vs Nombre de nœuds



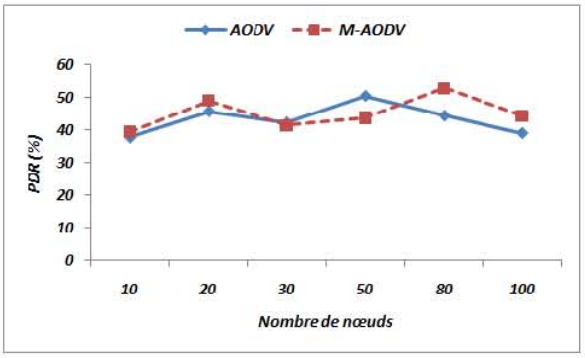
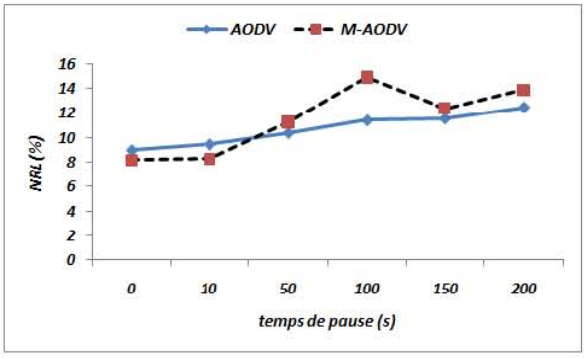


Figure 6. 20 (à gauche): Taux de paquets délivrés Vs Temps de Pause
Figure 6. 21 (à droite):Taux de paquets délivrés Vs Nombre de nœuds



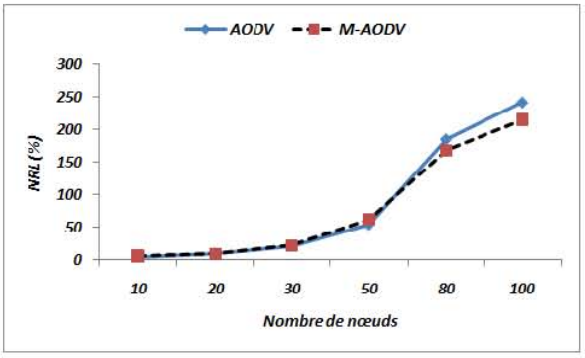


Figure 6. 22 : Charge de routage Vs Temps de Pause

Figure 6. 23 : Charge de routage Vs Nombre de nœuds

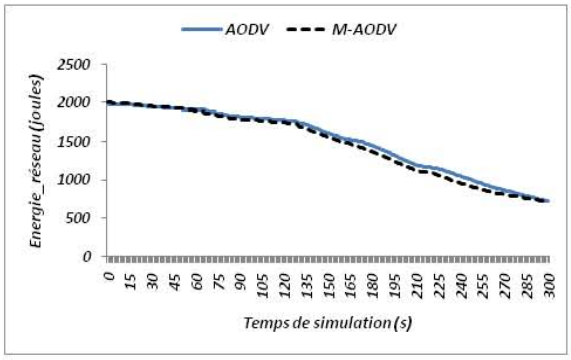


Figure 6. 24: Energie réseau Vs Temps de simulation

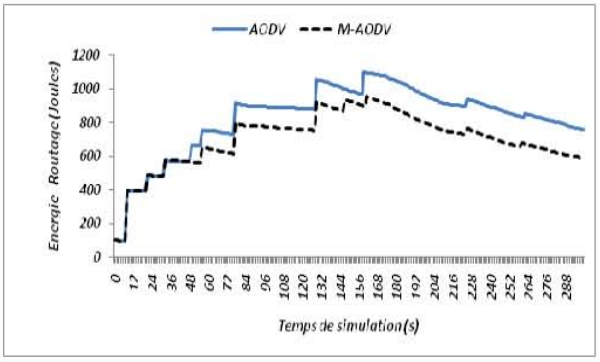


Figure 6. 25: Energie routage Vs Temps de simulation

Discussion :

On constate d'après les Figure 6. 14 & Figure 6. 14 que la présence de chemins de secours (multi chemins) dans la version M-AODV améliore mieux le débit surtout pour une forte et moyenne mobilité et pour des réseaux moins dense à la limite jusqu'à un nombre de 80 nœuds tandis que le délai moyen de bout en bout, sauf pour des réseaux denses (au delà de 80 nœuds) ou les deux protocoles offrent presque les mêmes performances. C'est toujours la version modifiée qui prend le dessus excepté pour une mobilité moyenne (un temps de pause de l'ordre de 50 s) ou la version de base (AODV) présente un délai meilleur que celui de M-AODV (Figure 6. 16 & Figure 6. 17).

Le Taux de perte des paquets est plus petit dans M-AODV pour une faible et une moyenne mobilité due au nombre de routes disponibles et pour des réseaux allant jusqu'à une taille de 75 nœuds (Figure 6. 18 &Figure 6. 19).

Le PDR dans de la version de base affiche des valeurs acceptables que ceux de la version M-AODV pour des réseaux d'une taille entre 30 et 70 nœuds (Figure 6. 21). En cas de très faible mobilité, les deux protocoles affichent les mêmes caractéristiques, mais la version modifiée est plus performante dans les autres cas (moyenne et forte mobilité) (Figure 6. 20).

Le taux NRL est plus ou moins équilibré c'est-à-dire parfois élevé due à la génération de plus de paquets de contrôle dans les deux variantes (plus de routes dans M-AODV avec donc plus de maintien et plus d'opérations de découverte dans AODV) (Figure 6. 22 &Figure 6. 22).

Concernant l'énergie consommée par l'ensemble des nœuds, les valeurs trouvées montrent que M-AODV gère mieux cette ressource que ce soit dans la phase de routage ou pour tout le réseau (Figure 6. 24 &Figure 6. 24).

VI.7.2.4. Conclusion

Nous pouvons conclure que la modification proposée M-AODV à base du protocole AODV sans la phase de réparation locale, à permis d'améliorer la QdS dans les réseaux mobiles ad hoc. Le protocole M-AODV permet de générer dans un premier temps, un nombre fixe de routes possibles entre une source et une destination (multi chemins), et les utilisera par la suite en cas de besoin (minimiser l'opération de redécouverte de routes) c'est-à-dire en cas de rupture d'un lien en cours d'utilisation et de veiller à avoir un nombre de routes avoisinant le nombre fixé auparavant et ceci en relançant la procédure de découverte de routes en parallèle à la phase de transfert des paquets de données.

VI.7.3. PF-AODV

VI.7.3.1. Contexte de simulation

Afin d'étudier et d'analyser le fonctionnement et le comportement de PF-AODV; nous avons repris le même contexte de simulation que celui utilisé dans M-AODV (section 6.7.2).

Dans le reste de cette section, on étudiera l'impact de la mobilité (respectivement : Débit de transmission en Kb/s) sur ces paramètres en faisant varier le temps de pause de zéro à 200s (Respectivement en faisant varier le taux de transmission d'une valeur de 0.2 à 10 Kb/s) (Voir section 6.7.2.2).

VI.7.3.2. Courbes & discussions

Courbes :

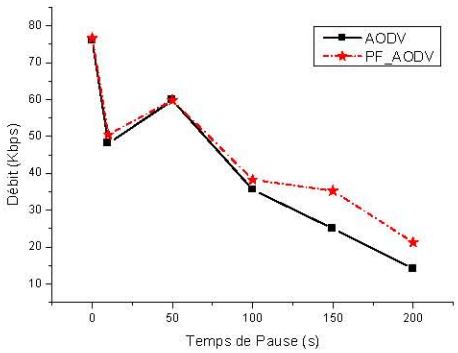


Figure 6. 26: Débit utile Vs Temps de Pause

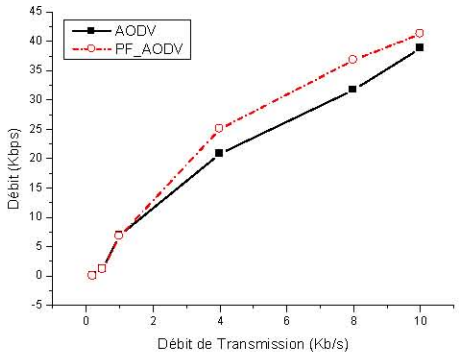
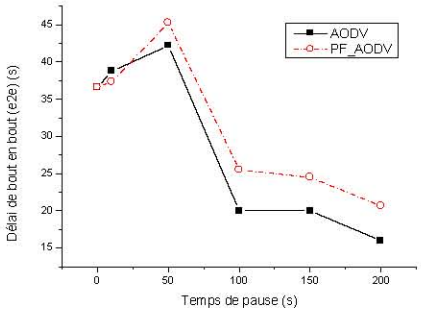


Figure 6. 27: Débit utile Vs Débit de Transmission



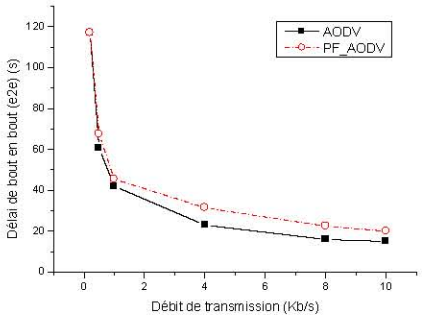
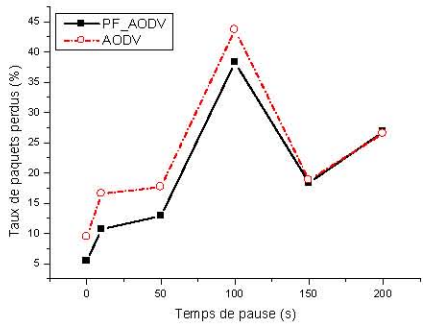


Figure 6. 28 (à gauche): Délai moyen de bout en bout Vs Temps de Pause
Figure 6. 29 (à droite): Délai moyen de bout en bout Vs Débit de Transmission



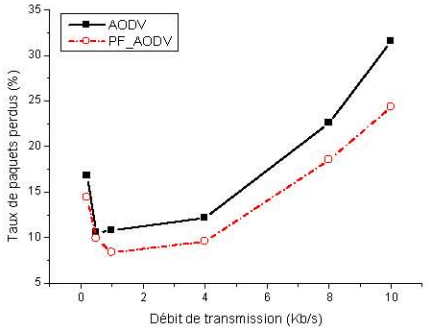
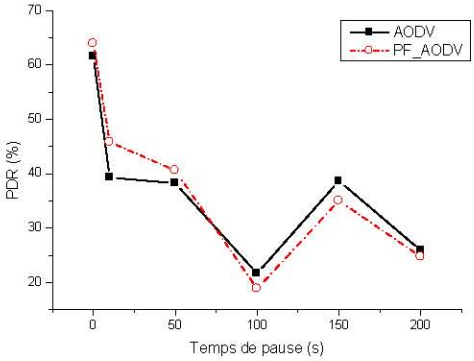


Figure 6. 30 (à gauche): Taux de Paquets Perdus Vs Temps de Pause
Figure 6. 31 (à droite): Taux de Paquets Perdus Vs Débit de Transmission



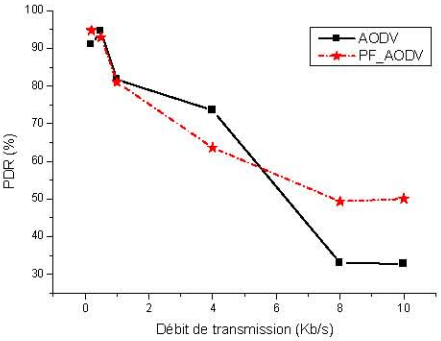


Figure 6. 32 (à gauche): Taux de Paquets Délivrés Vs Temps de Pause
Figure 6. 33 (à droite): Taux de Paquets Délivrés Vs Débit de Transmission

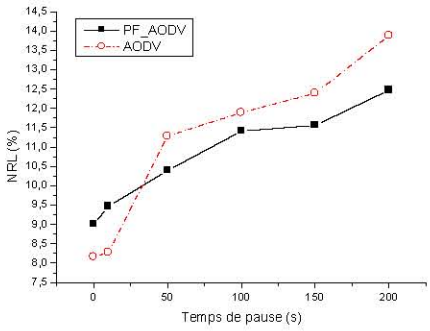


Figure 6. 34: Charge de routage Vs Temps de Pause

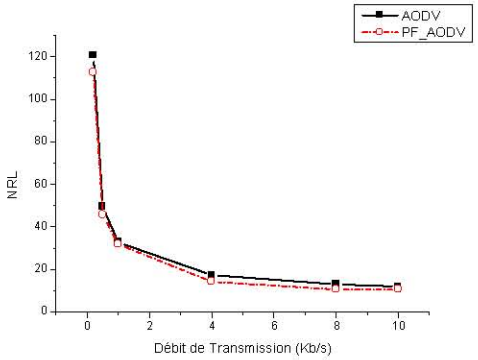


Figure 6. 35 : Charge de routage Vs Débit de Transmission

Discussion :

Les Figure 6. 26 &Figure 6. 26) montrent que le débit s'améliore dans la version modifiée (PF-AODV) pour une faible et moyenne mobilité (un temps de pause entre 50s et 200s) et un taux de transmission important (entre 1.5 et 10 Kb/s) tandis que pour une forte mobilité et des taux de transmissions faibles (entre 0 et 1.5 Kb/s) on remarque que les deux protocoles (AODV et PF-AODV) offres les mêmes performances.

Le délai moyen de transfert est d'un écart d'environ 10s pour des taux de transfert entre 2 et 10 Kb/s (Figure 6. 29) en faveur du protocole PF-AODV sur l'AODV mais cet écart est au environ de 5s (Figure 6. 28) toujours au profit de la version modifiée pour une mobilité faible et moyenne (jusqu'à un temps de pause de 20s) ceci est du à la disponibilité de plusieurs liens stables entre nœuds voisins faisant partie d'une route établie. Dans le cas des taux de transmissions faibles (< 2kb/s) et une mobilité très forte c'est la version originale qui offre les meilleurs résultats.

Le mécanisme de prédiction de rupture de routes a permis à la version modifiée (PF-AODV) de disposer de routes mieux que la version de base et par conséquent il minimise grandement la perte des paquets dans toutes les situations et selon les deux contraintes (mobilité et du débit de transmission) utilisées dans notre simulation (Figure 6. 30 & Figure 6. 31).

Le taux de paquets livrés avec succès à base du protocole PF-AODV affiche des valeurs acceptables que ce soit dans le contexte de mobilité ou du débit de transmission (Figure 6. 32 &Figure 6. 33). Le taux de charge de contrôle normalisée est plus ou moins équilibré et le rapport concernant les deux protocoles est d'un ordre avoisinant presque un c'est-à-dire plus il y a de transfert réussi, plus il y a de paquets de contrôle (Figure 6. 34). Pour les mesures à base de la mobilité, ce rapport est parfois élevé du à la génération de plus de paquets de contrôle pour le rétablissement des routes rompues surtout dans l'AODV (Figure 6. 34).

VI.7.3.3. Conclusion

En conclusion, nous pouvons dire que la modification désignée par PF-AODV a permis d'améliorer d'une manière significative la Qds dans les réseaux ad hoc (MANETs) selon les paramètres mesurés et bien sûr dans le contexte de simulation proposé. L'utilisation de la puissance du signal pour prédire une future déconnexion d'un chemin en cours d'utilisation et de prévoir une autre issue (lien de secours) avant que cette rupture aura lieu, minimise grandement le trafic de contrôle généré lors de la phase de reconstruction des chemins comme c'est le cas dans la version originale. Les résultats montrent bien que PF-AODV apporte de meilleures performances en termes de débit, de perte et de délai. De futures extensions du protocole AODV visant à ajouter un contrôle d'admission pour gérer chaque type de trafic séparément et voir comment favoriser les trafics prioritaires et quel genre de trafic doit on pénaliser pour libérer de la bande passante.

VI.8. Contribution au niveau de la couche Mac

La seule modification proposée au niveau de la couche MAC, porte sur la variation des valeurs de la fenêtre de contention avec une nouvelle incrémentation de la valeur de Backoff.

VI.8.1 Contexte de simulation

Les paramètres listés dans le Tableau 6. 5 représentent les valeurs utilisées dans NS-2 pour la couche IEEE 802.11 avec les paramètres de simulation utilisés.

Paramètres	Valeurs
Temps de simulation	100 s
Accès au medium	IEEE 802.11
Protocole de routage	AODV
Taille de buffer	50
Topologie de simulation	1200×1200
SlotTime	20 µs
SIFS	10 µs
CWMin	31
CWMax	1023
Nombre de nœuds	10

Tableau 6. 5: Paramètres de simulation et de la couche Mac

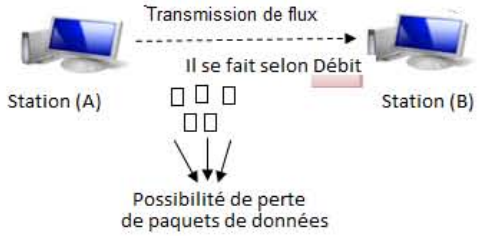


Figure 6. 36 : Transfert de flux et perte de données

La perte de paquets est chose possible lors d'une transmission d'un flux entre deux stations qui se fait selon un débit (Figure 6. 36).

Les paramètres qu'on va mesurer sont : le nombre de paquets perdus et le débit de transfert de données en (bits/s) en fonction du temps de simulation.

VI.8.2 Comparaison et Analyse des résultats

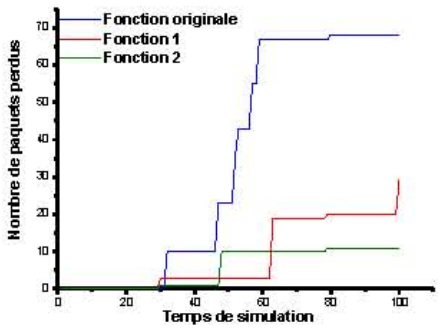
Pour analyser les performances de la modification implémentée dans la couche MAC (voir section 5.3.5.1 du chapitre 5); on a utilisé 6 scenarios différents (Tableau 6. 6) combinant deux contraintes (mobilité et densité). Une comparaison est faite entre la proposition et la version de base (i.e. originale).

La première étude est faite par scénario en fonction du temps de simulation, par contre la deuxième est faite sur l'ensemble des six scénarios [147].

Mobilité / Nœuds	Faible	Forte
10	Scénario 1	Scénario 2
20	Scénario 3	Scénario 4
50	Scénario 5	Scénario 6

Tableau 6. 6: les différents scenarios de simulation

Courbes :



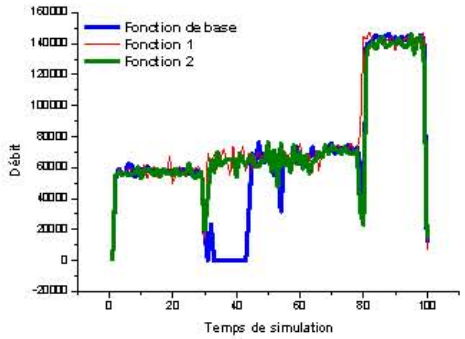


Figure 6. 37 : Nombre de paquets perdus (Scenario 1)

Figure 6. 38 : Débit Vs temps de simulation (Scenario1)

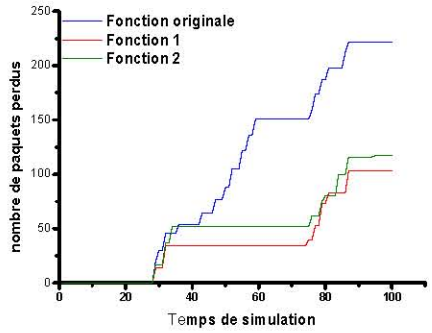


Figure 6. 39 : Nombre de paquets perdus (Scenario 2)

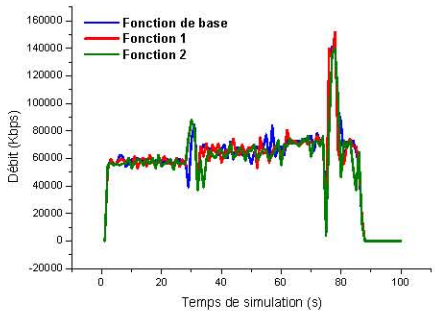


Figure 6. 40 : Débit Vs temps de simulation (Scenario2)

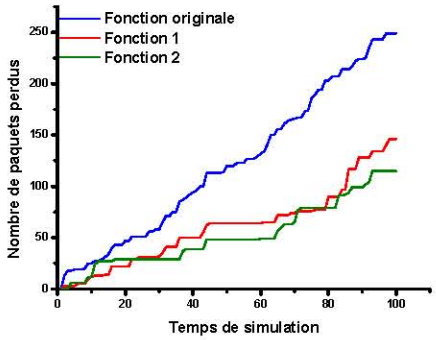


Figure 6. 41 : Nombre de paquets perdus (Scenario 3)

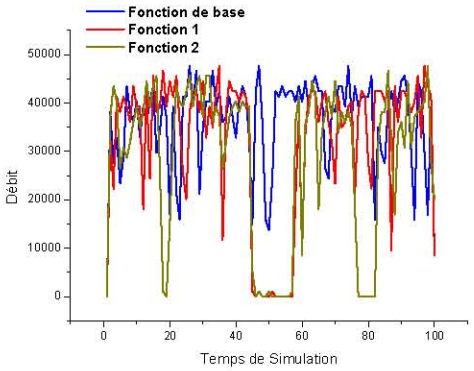


Figure 6. 42 : Débit Vs temps de simulation (Scenario3)

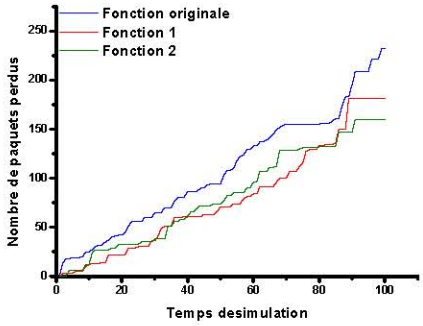


Figure 6. 43 : Nombre de paquets perdus (Scenario 4)

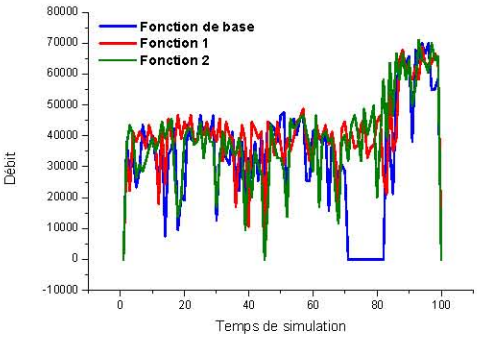


Figure 6. 44 : Débit Vs temps de simulation (Scenario4)

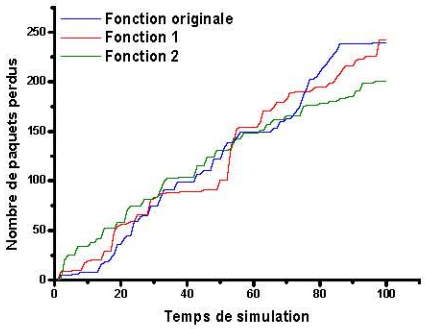


Figure 6. 45 : Nombre de paquets perdus (Scenario 5)

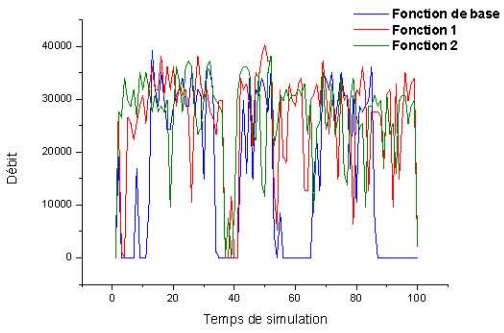


Figure 6. 46 : Débit Vs temps de simulation (Scenario5)

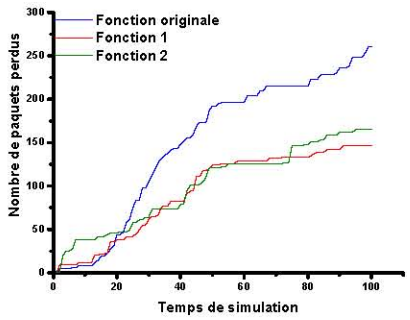


Figure 6. 47 : Nombre de paquets perdus (Scenario 6)

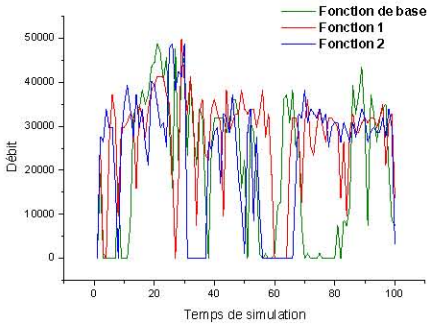


Figure 6. 48 : Débit Vs temps de simulation (Scenario6)

Discussion :

Dans le scenario 1 (i.e. un petit nombre de nœuds et une faible mobilité), on remarque que le nombre de paquets perdus pour les deux fonctions (1 et 2) proposées est nettement inférieur à celui de la fonction de base (Figure 6. 37). Par contre, on constate un léger apport (presque identique) pour le paramètre débit sur toute la durée de simulation sauf dans l'intervalle allant de ≈30s à ≈60s ou le débit dans la modification reste stable ou il affiche une petite augmentation par rapport à la période de temps précédente, tandis que le débit chute dans cet intervalle suite au nombre de retransmission assez élevé car la taille de la CW dans la DCF de base est très large (Figure 6. 37).

Dans le scenario 2 (i.e. un petit nombre de nœuds et une forte mobilité), on constate la même remarque que le scenario 1 pour le nombre de paquets perdus, et par conséquent la modification proposée (Fonction 1 & 2) offre de meilleures performances (Figure 6. 39) dans un contexte de forte mobilité. Quant au débit, on remarque qu'il est presque égal pour les trois fonctions (Figure 6. 39).

Pour le scenario 3 (i.e. un nombre moyen de nœuds et une faible mobilité) concernant le nombre de paquets perdus (Figure 6. 41), la même remarque est à faire que ceux des deux scenarios précédents. Pour le débit (Figure 6. 41), c'est la version originale qui prend le devant par rapport aux modifications (Fonctions 1 & 2) car dans ce cas, un nombre de retransmission élevé (large CW) de la fonction de base donne plus de chances aux stations d'accéder au canal puisque la topologie du réseau reste presque inchangée (i.e. la présence d'un nombre important de voisins avec un mouvement faible).

Pour les scénarios 4, 5 et 6 (Figure 6. 43, Figure 6. 43, Figure 6. 45, Figure 6. 45, Figure 6. 47 & Figure 6. 47), la modification offre de meilleurs résultats que la fonction de base que ce soit pour le débit ou le nombre de paquets perdus sauf pour le scenario 5 (Figure 6. 45) où les trois fonctions génèrent le même nombre de paquets perdus vu que la topologie du réseau reste presque stable (faible mobilité).

On peut conclure que la modification proposée présente de meilleures performances dans presque la totalité des scénarios testés.

VI.8.3 Récapitulatif des résultats trouvés : comparaison et analyse

Dans cette section, on essaye d'analyser le comportement des trois fonctions pour l'ensemble des six (06) scénarios vus précédemment. Une comparaison est faite entre les trois fonctions.

Les paramètres de QdS mesurés sont : le nombre de paquets émis, perdus, le taux de perte et le débit [147].

Collecte des résultats : Les Tableau 6. 7, Tableau 6. 8, Tableau 6. 9 et Tableau 6. 10) récapitulent respectivement les résultats de l'ensemble des scénarios réalisés sur les paramètres : paquet émis, perdus, taux de perte (i.e. (le nombre paquets émis sur le nombre paquets reçus) *100) et le débit moyen en bits/s pour l'ensemble des scénarios.

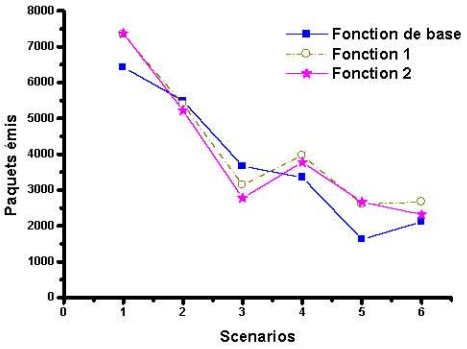
Paquets émis et perdus

Scenarios / Fonctions	1	2	3	4	5	6
DCF de base	6433	5491	3668	3353	1626	2118
Fonction 1	7346	5395	3147	3966	2608	2671
Fonction 2	7374	5224	2778	3779	2668	2321

Tableau 6. 7: paquets émis

Scenarios / Fonctions	1	2	3	4	5	6
DCF de base	68	222	249	232	239	260
Fonction 1	29	103	146	181	238	146
Fonction 2	11	117	115	160	201	165

Tableau 6. 8: paquets perdus



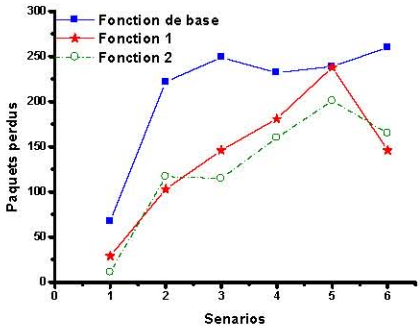


Figure 6. 49 : paquets émis

Figure 6. 50 : paquets perdus

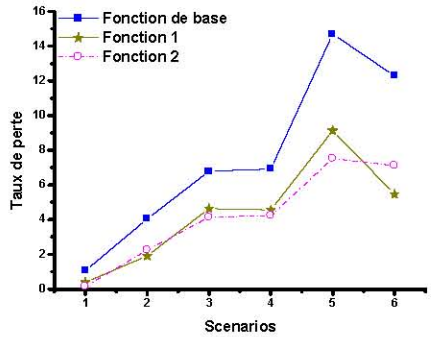
Taux de perte et Débit moyen

Scenarios / Fonctions	1	2	3	4	5	6
DCF de base	1,06	4,04	6,79	6,92	14,70	12,28
Fonction 1	0,39	1,91	4,64	4,56	9,13	5,47
Fonction 2	0,15	2,24	4,14	4,23	7,53	7,11

Tableau 6. 9 : Taux de perte(%)

Scenarios / Fonctions	1	2	3	4	5	6
DCF de base	67968	57039	36083	33338	15519	20448
Fonction 1	77688	56541	31716	40662	25886	27264
Fonction 2	74709	54654	28419	38744	26702	23649

Tableau 6. 10: Débit moyen



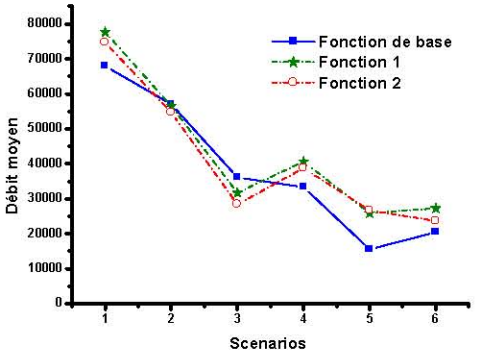


Figure 6. 51 : Taux de perte

Figure 6. 52 : Débit moyen

Discussion :

On constate que pour le cas des paquets émis (Figure 6. 49), il y a plus d'émission dans la modification proposée sauf dans le scenario 3 (i.e. la topologie du réseau reste presque stable) où la DCF de base affiche un nombre d'émission supérieure dû au nombre de retransmission.

Pour le cas des paquets perdus (Figure 6. 49), l'apport des deux propositions (Fontion1 & 2) est nettement meilleur dans la totalité des scénarios sauf pour le scenario 5 ou la fonction 1 est identique à la fonction de base. La même remarque est à faire pour le cas du taux de perte (Figure 6. 51) dans la majorité des scénarios (i.e. les deux fonctions proposées réduisent considérablement la perte de paquets).

Concernant le débit observé (Figure 6. **51**) pour la modification (Fonction 1 & 2), il est supérieur que celui de la fonction de base dans la majorité des scenarios sauf pour le scenario 3 où la version originale offre un débit meilleur que la modification proposée.

Concernant la proposition faite dans la couche MAC, et sur la base des résultats des différents scénarios, on peut conclure que les fonctions 1 & 2 ont montrés des résultats meilleurs par rapport à ceux de la DCF de base pour les paramètres mesurés (Débit & nombre de paquets perdus) sous les deux contraintes (mobilité &t nombre de nœuds).

VI.9. Conclusion

Dans ce chapitre, nous avons essayé d'analyser les résultats trouvés par simulation pour l'ensemble des contributions. Dans la première partie, on a présenté les contraintes (mobilité, énergie et passage à l'échelle) sous lesquelles les simulations sont faites et les paramètres (perte des paquets, la charge de contrôle, le débit et le délai) qu'on désire évaluer. Une discussion approfondie est faite dans la partie deux sur chaque contribution, qui nous a permis de conclure que l'ensemble des modifications proposées sont d'un apport significatif, ce qui permet d'améliorer la QdS dans les contextes de simulation réalisés.

Conclusion générale

L'objectif de cette thèse a été la proposition d'une architecture pour le support de la Qualité de service dans les réseaux sans fil ad hoc.

Le mémoire a comporté deux grandes parties. La première a été consacrée à un état de l'art dans le domaine des réseaux mobiles ad hoc, et à un bilan relatif aux différents protocoles de routage ainsi qu'à une étude détaillée de la QdS et des différents modèles de QdS dans les MANETs.

La contribution qui a fait l'objet de la deuxième partie, a concerné essentiellement des modifications et des extensions du protocole AODV sur lequel ont porté nos propositions pour le support de la qualité de service et qui ont été expérimentées par simulation sous NS2

Notre thèse s'intéresse à la qualité de service dans les réseaux mobiles ad hoc qui représente un sujet de recherches d'actualité ouvert et très délicat. Les solutions ou modèles de QdS proposés dans la littérature pour ces réseaux font partie d'une seule couche ou d'une combinaison de couches de la pile protocolaire : MAC, Réseau et Application. Chacune de ces solutions et modèles tente d'améliorer la QdS pour un ensemble réduit de paramètres.

Notre contribution s'est articulée autour des volets suivants:

1. Une amélioration de l'algorithme AODV pour un meilleur apport en QdS (au niveau de la couche réseau),
2. Optimisation des collisions à base de la DCF (au niveau de la couche MAC) et
3. Une proposition d'une architecture pour le support de la QdS.

Les solutions proposées au niveau routage ont porté en premier lieu sur la réduction de la charge de contrôle générée par ce protocole pour mieux gérer la bande passante, et ce par élimination de certaines informations de routage qui ne sont pas souvent nécessaires.

La première solution a porté sur la stratégie à adopter durant une phase de transfert en permettant au nœud ayant détecté la rupture, d'aviser le plutôt la source pour qu'elle stoppe ses transmissions en octroyant une taille suffisante à la file d'attente associée aux nœuds pour recevoir le maximum de paquets et éviter leurs pertes en attendant que la réparation soit faite en local ou à la source (AODV-SR) (compromis espace, délai /perte).

La deuxième solution prévoit plus d'une route (multi chemins) qu'on utilisera en cas de rupture de liaison (M-AODV). L'insuffisance de cette solution a été comblée en proposant un compromis entre la disponibilité de routes et la charge de contrôle utilisée pour leur maintien qui consomme assez de bande passante (compromis débit/perte).

La dernière proposition a consisté à doter les nœuds d'un mécanisme de prédiction de rupture de liens (PF-AODV) en se basant sur la force du signal, ce qui permettra de prévoir un lien de secours qu'on utilisera si la rupture aura lieu.

Le protocole de routage avec QdS, détermine les routes qui répondent aux exigences de QdS (i.e. le délai et la bande passante) sans assurer leurs réservations. La réservation est une tache très difficile à cause de la présence de collisions dans une méthode d'accès au support comme CSMA/CA. Au niveau de la couche MAC, les paquets subissant des collisions doivent être retransmis. Ces retransmissions consomment assez de bande passante et réduisent le délai de transfert des données. Pour faire face à ce phénomène, notre contribution à ce niveau a consisté à suggérer une modification de la procédure d'accès au médium DCF de IEEE 802.11, en proposant de meilleures valeurs de la fenêtre de contention (CW) afin de minimiser les collisions pour permettre le respect des contraintes de QdS.

En fait, nous avons essayé d'obtenir une meilleure QdS en contrôlant le débit, le délai, la charge de contrôle et la perte des paquets pour les différentes propositions.

Comme perspectives pour la poursuite de nos travaux, nous pensons orienter nos recherches vers les directions suivantes :

Nous comptons en un premier temps, tester les solutions proposées pour un réseau de grande échelle et les comparer avec des travaux similaires (i.e. gérant la QdS) dans les réseaux ad hoc, ensuite, développer la phase de réservation de ressources après détermination des routes répondant à une certaine QdS par le protocole de routage. En second lieu, nous visons à étudier et proposer des solutions permettant d'améliorer le temps NAV dans la couche MAC pour minimiser le temps d'attente et par conséquent augmenter le délai de transfert.

Nous pensons compléter notre étude en comparant le mécanisme d'admission centralisé et distribué pour les solutions proposées, et développer des solutions à entreprendre lorsque les ressources déjà réservées se dégradent, c'est-à-dire répondre à la question comment choisir les flux à éliminer et à quel instant faut-il initier une nouvelle recherche de routes pour certaines contraintes de QdS.

Nous projetons aussi proposer des mécanismes pour mieux gérer l'énergie des différents nœuds et permettre ainsi une meilleure utilisation et une longue durée de vie et exploiter notre contribution dans les réseaux de capteurs (WSN).

Bibliographie

[1] R. RAHIM, "*Etudes des protocoles de mobilité inter-domaine des réseaux sans fil*", Diplome d'Etudes Approfondies, Université saint-Joseph, LIBAN, Soutenue le 23 décembre 2003.

[2] M. ABDELADIM, E.M.B. LOUHAIDIA, "*Implémentation et évaluation de performances d'un protocole de contrôle de flux dans les réseaux Ad hoc*", Mémoire d'ingénieurs, Ecole nationale Supérieure d'Informatique (ESI), Alger, Promotion : 2008/2009.

[3] E. CONCHON, "*Définition et mise en oeu vre d'une solution d'émulation de réseaux sans fil*", Thèse de Doctorat, Institut National Polytechnique De Toulouse, Soutenue le 27 Octobre, 2006.

[4] J.P. CHANET, "*Algorithme de routage coopératif a qualité de service pour des réseaux Ad hoc agri-environnementaux*", Thèse de doctorat, Université Blaise Pascal - Clermont II, Soutenue le 20 avril 2007.

[5] L. AOURAGH, L. GUETTALA, "*Etude Comparative des protocoles de routages sous Ns2*", mémoire d'ingenieur, Université de Batna, promotion 2007.

[6] T. LEMLOUMA, *Le Routage dans les Réseaux Mobiles Ad-hoc*", Min projet, Université des Sciences et de la Technologie Houari Boumediene, Institut d'Informatique, Septembre 2000.

[7] S. CORSON, J. MACKER, "*Mobile Ad hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations. Internet Request for Comments RFC 2501*", Internet Engineering Task Force (IETF), january 1999.

[8] R. MERAIHI, "*Gestion de la qualité de service et contrôle de topologie dans les réseaux Ad hoc*", Thèse de doctorat de l'Ecole nationale supérieure des télécommunications (ENST), Paris, France. Soutenue en 2005.

[9] G. PEI, M. GERLA, X. HONG et C.C. CHIANG, "*A wireless hierarchical routing protocol with group mobility*", In IEEE WCNC'99, septembre, 1999.

[10] D. TREZENTOS, "*Standard pour réseaux sans fil: IEEE 802.11* ", In Traité Télécoms, volume TE 7 375, pages 1-12. Techniques de l'ingénieur, 2002.

[11] J. Van der MEERSCEN, "*Hybridation entre les modes Ad hoc et infrastructure dans les réseaux de type Wi-Fi*", Rapport de DEA, 2006.

[12] W. BENHAMMADI, M. SCHOOBROODT, "*Fonctionnement de la norme de base IEEE 802.11 : Architecture et modes de fonctionnement*", 2006.

[13] D. DHOUTAUT, "*Etude du standard IEEE 802.11 dans le cadre des réseaux Ad hoc : de la simulation a l'expérimentation*", Thèse de Doctorat, Institut National des Sciences Appliquées de Lyon, Soutenue le 11 Décembre, 2003.

[14] K. AL AGHA, G. PUJOLLE, et G. VIVIER, "*Réseaux de mobiles et Réseaux sans fil*", No. ISBN 2-212-11018-9, Eyrolles, 2001.

[15] M. BRAHMA, "*Etude de la QoS dans les Réseaux Ad hoc : Intégration du Concept de l'Ingénierie du Trafic*", Thèse de doctorat, Université de Haute Alsace, UFR des Sciences et Techniques, Soutenue le 13 décembre 2006 Numéro d'ordre : 06MULH0844.

[16] IEEE Computer Society, "*Mobile Ad hoc networks*", Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte, NC 28223-0001.

[17] A. KSENTINI, "*Qualité de Service (QoS) dans les réseaux locaux sans fil basés sur la technologie IEEE 802.11*", Thèse de Doctorat, Université de Cergy-Pontoise, Soutenue le 08 Décembre, 2005.

[18] A. VAN DEN BOSSCHE, "*Proposition d'une nouvelle méthode d'accus déterministe pour un réseau personnel sans fil a fortes contraintes temporelles*", 2007.

[19] M. El MASRI, "*Contribution a la qualité de service dans les réseaux d'accus sans fil*", Thèse de Doctorat, Université de Toulouse, Soutenue le 9 juillet, 2009.

[20] M. TERRE, "*WiFi: Le Standard 802.11Couche physique et couche MAC*", 2007.

[21] P. VAN VIET, "*Etudes des modules sans fil dans NS2*", Rapport final ; Institut de la Francophonie pour l'Informatique (IFI), Hanoi,

[22] A. NASIPURI, "*IEEE Standard for Information technology, Telecommunications and information exchange between systems Local and metropolitan area networks specific requirements*", Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-2007.

[23] R. MOAWAD, "*QoS dans les WPAN, WLAN et WMAN*", 2004.

[24] L. GANNOUNE , S. ROBERT et D. RODELLAR, *A Survey of QoS Techniques and Enhancements for IEEE 802.11 Wireless LANs*", Ecole d'Ingénieurs du Canton de Vaud (EIVD), Switzerland, 2003.

[25] D. ESPES, "*Protocoles de routage réactifs pour l'optimisation de bande passante et la garantie de délai dans les réseaux Ad hoc mobiles*", Thèse de Doctorat ; Université Toulouse III - Paul Sabatier, Soutenue le 27 novembre, 2008.

[26] C. CHENG, R. RILEY, S. KUMAR et J.J. GARCIA-LUNA-ACEVES, "*A Loop-Free Bellman-Ford Routing Protocol without Bouncing Effect*", ACM SIGCOMM'89, Sept 1989.

[27] "*http://malm.tuxfamily.org/doc/qr-chap1-algo.htm*", Date de consultation, Mars 2007.

[28] F. THEOLEYRE, F. VALOIS, "*Routage Hybride sur Structure Virtuelle dans les Réseaux Mobiles Ad hoc*", CFIP'05 : Colloque francophone sur l'ingénierie des protocoles, 29 Mars - 1er Avril, Bordeaux, France, 2005.

[29] M. ACHIR, "*Technologies basse consommation pour les réseaux Ad hoc*", Thèse pour obtenir le grade de docteur, INPG : Institut National Polytechnique De Grenoble, LETI-DCIS-SASTI du CEA Grenoble, Soutenue le 06 juillet 2005.

[30] "*http://www.inria.fr/valorisation/standardisation/sansfil/index.fr.html*",Date de consultation, Mars 2007.

[31] Y.C. HU, D.B. JONSON, "*Implicit source routes for on-demand Ad hoc network routing*", In MobiHoc '01 : Proceedings of the 2nd ACM international symposium on Mobile Ad hoc networking and computing, pages 1-10, New York, NY, USA, ACM Press, 2001.

[32] V. PARK, M. CORSON, "*Temporally-ordered routing algorithm (TORA)*", Internet Draft draft-ietf-MANET-tora-spec-04.txt, Internet Engineering Task Force, July 2001.

[33] "*http://ethesis.inp-toulouse.fr/archive/00000277/*", Date de consultation, Décembre 2006

[34] M. ABOLHASAN, T. WYSOCKI, et E. DUTKIEWICZ, "*A review of routing protocols for mobile Ad hoc networks*", Ad hoc Network, vol. 2, pages 1-22, 2004.

[35] E.M, BELDING-ROYER. et C.K. TOH, "*A review of current routing protocols for ad-hoc mobile wireless networks*", IEEE Personal Communications Magazine, pages 46-55, 1999.

[36] STOJMENOVIC, "*Position based routing in Ad hoc networks*", IEEE Communications Magazine, vol. 40, no. 7, pages 128-134, 2002.

[37] G. CHELIUS, "*Architectures et communications dans les réseaux spontannés sans fil*", Thèse de Doctorat en Informatique, INSA de Lyon, Soutenue le 26 Avril, 2004.

[38] M. MAUVE, J. WIDMER et H. HARTENSTEIN, "*A survey on position-based routing in mobile Ad hoc networks*", IEEE Network Magazine. v15 i6, pages 30-39, 2001.

[39] B. MOLO, "R*outage dans les réseaux mobiles Ad hoc*", Maitrise en informatique pour Grade de Maitre ıs sciences, Faculté des sciences et de génie, université de Laval, QUÉBEC, Soutenue le Juin, 2007.

[40] T. CLAUSEN, P. JACQUET, "*Optimized Link State Routing Protocol (OLSR)*", Request for Comments (Draft Standard) 3626. Internet Engineering Task Force, Oct. 2003.

[41] A. QAYYUM, L. VIENNOT, et A. LANOUITI, "*Multipoint relaying: an efficient technique for flooding in mobile wireless Networks*", Rapport de recherche 3898, Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt, 16 pages France, 2000.

[42] C. E. PERKINS, P. BHAGWAT, "*Ad hoc networking, Chapitre DSDV: routing over a multihop wireless network of mobile computers*", pages 53-74, Addison-Wesley, 2001.

[43] C.A. AL-KHWILDI, Y. CASEY, H. ALDELOU, et H.S. AL-RAWESHIDY, "*A performance comparison of multi on-demand routing in wireless Ad hoc Networks*", In IEEE International Conference on Wireless And Mobile Computing, volume 3, pages 9-16, USA, August 22-24. 2005.

[44] C.S.R. MURTHY, B.S. MANOJ, "*Ad hoc Wireless : Architectures and Protocols*", Prentice Hall Communications Engineering and Emerging Technologies Series, Prentice Hall, 1ıre edition, 05, 2004.

[45] D.B. JOHNSON, D.A. MALTZ, "*Dynamic Source Routing in Ad hoc Wireless Networks*", In Imielinski and Korth, editors, Mobile Computing, volume 353, Kluwer Academic Publishers, 1996.

[46] D. JOHNSON, Y. HU, et D. MALTZ, "*The Dynamic Source Routing Protocol (DSR) for Mobile Ad hoc Networks for IPv4*", RFC 4728, February, 2007.

[47] V. UNTZ, "*Les réseaux sans fil spontanés pour l'Internet Ambiant*, Thèse de Doctorat, INP Grenoble, Soutenue le 5 décembre, 2007.

[48] M.K. MARINA, S.R. DAS, "*Performance of route caching strategies in dynamic source routing*", In ICDCSW'01 : Proceeding of the 21st International Conference on distributed Computing Systems, page 425, IEEE Computer Society, April 16-19, 2001.

[49] S. BOUDJIT, "*Autoconfiguration and Security Schemes for OLSR Protocol for Mobile Ad hoc Networks*", Thèse de doctorat, Université Pais Sud, Faculté des Sciences d'Orsay, Soutenue le 25 Septembre 2006, N° : 8388.

[50] G. PEI, M. GERLA, X. HONG, et C.C. CHIANG, "*A wirless hierarchical routing protocol with group mobility*", In Proceeding of IEEE Wirless Communications and Networking Conference, WCNC'99, number 1, pages 1538-15-42, New Orleans, LA, USA, septembre, 1999.

[51] J.H. ZYGMUNT, M.R. PEARLMAN, "*Zone routing protocol for ad-hoc networks*", Internet Draft, draft-ietf-MANET-zrp-02.txt, work in progress, 1999.

[52] J.H. ZYGMUNT, M.R. PEARLMAN, et P. SAMAR, "*The Intrazone Routing Protocol (IARP) for Ad hoc networks*", draft-ietf-MANET-zone-iarp-01.txt, Draft IETF, Juin 2001.

[53] J.H. ZYGMUNT, M.R. PEARLMAN, et P. SAMAR, "*The Intrezone Routing Protocol (IERP) for Ad hoc networks*", draft-ietf-MANET-zone-ierp-01.txt, Draft IETF, Juin 2001.

[54] Livre chez Eyrolles, "*802.11 et Les Réeseaux Sans Fil*", http://www.eyrolles.com, publiи en aout, 2002 .

[55] M. JIANG, J. LI et Y.C. TAY, "*Cluster Based Routing Protocol (CBRP)*", Internet draft 1, IETF-MANET Working Group, [on line] 1999.

[56] K. BEYDOUN, "*Conception d'un protocole de routage hiurarchique pour les réseaux de capteurs*", Thèse de Doctorat, U.F.R des sciences et techniques de l'université de Franche-Comte, Soutenue le 16 décembre, 2009.

[57] S. LINDSEY, C.S. RAGHAVENDRA, "*PEGASIS: Power-efficient gathering in sensor information systems*", IEEE Aerospace Conference Proceedings, Vol. 3, pages 3-11,30, 2002.

[58] N. DAUJEARD, J. CARSIQUE, R. LADJADJ, et A. LALLEMAND, "*Le routage dans les réseaux mobiles Ad hoc*", Année 2002-2003.

[59] C.-C. CHIANG, H.-K. WU, W. LIU, et M. GERLA, "*Routing in clustered multihop, mobile wireless, networks with fading channel*", IEEE SICON, pages 197-211, 1997.

[60] Participant : LIFL CReSTIC, "*Rapport sur l'état de l'art des algorithmes de routage*", Projet RISC (Résaux Hétérogиnes Intélligents pour Situation de Crise) Rapport RISC-SP3.1-#6.ed0 ANR (Agence Nationale de la Recherche), 30/05/2008.

[61] "*The Global Positioning System FAQ*", URL: http://www.gpsy.com/gpsinfo/gps-faq.txt, July, 1997.

[62] M. HAUSPIE, "*Contributions a l'étude des gestionnaires de services distribués dans les réseaux Ad hoc*", Thèse doctorat, Université des Sciences et Technologies de Lille, Soutenue le 14 janvier, 2005.

[63] Y.-B. KO, N.H. VAIDYA, "*Location-Aided Routing (LAR) in mobile Ad hoc Networks*", In Proceedings of ACM/IEEE MOBICOM'98, pages 66-75, Dallas, Texas, October 1998.

[64] C. BASILE, M-O. KILLIJIAN, D.POWELL, "*A Survey of Dependability Issues in Mobile Wireless Networks*", Technical Report, LAAS CNRS Toulouse, France 2003.

[65] D.ELORRIETA,"*Protocoles de routage pour l'interconnexion des réseaux Ad-Hoc et UMTS*", Mem.DEA 2007.

[66] N. BACCOUR SELLAMI, "*Conception d'une nouvelle strategie de routage dynamique pour les reseaux mobiles Ad hoc*", mémoire de Mastere en Informatique Industrielle, l'École Nationale d'Ingénieurs de Sfax , Tunisie, Soutenue le 27 juin 2006, N° d'ordre: 2006-AII .

[67] S. BASAGNI, I. CHLAMTAC, V.R. SYROTIUK et B.A. WOODWARD, "*A distance routing effect algorithm for mobility (Dream)* ", Proceeding. 4th ACM/IEEE, Int'l Conf. on Mobile Computing and networking (MobiCom'98), pages 76-84, New York, NY,USA, ACM Press, 1998.

[68] P. PRIMET, "*Contribution au Support réseau des applications réparties Qualité de Service : pour un réseau sensible aux flux*", Thèse d'Habilitation a Diriger des Recherches, a UCB Lyon, Laboratoire RESAM, ENS Lyon ,France, Soutenue le 20/04/2002.

[69] O. VILLIN, "*Gestion de la qualité de service de bout en bout dans les systumes répartis: approche gestion de ressources*", Thèse de Doctorat, Université d'Évry Val d'Essonne, 29 Avril, 2002 .

[70] "*Recommendation E800: Terms and Definitions Related to QoS and Network Performance Including Dependability*", TeIecommunication Standardization Sector of ITU-T. Aout 1994.

[71] E. CRAWLEY, R. NAIR, B. RAJAGOPALAN, et H. SANDICK, "*A Framework for QoS-based Routing in the Internet*", IETF RFC2386 .

[72] K. AL AGHA, "*Qualité de Service dans les Réseaux Ad hoc et Wi-Max*, Laboratoire De Recherche En Informatique Université de Paris-Sud, 2004.

[73] L. TOUMI, "*Algorithmes et mécanismes pour la qualité de service dans les réseaux hétérogunes*", Thèse de Doctorat a L'INPG, Soutenue le 20 décembre, 2002.

[74] S. KAMMOUN, "*Implémentation de la QoS sur un protocole de routage (multicast) Ad hoc*", Mémoire d'ingénieur, Ecole supérieur de communication de Tunis, Promotion 2005/2006.

[75] N. YOUNES, "*La Qualité de service des services multimédia sur les réseaux Ad hoc sans fil à multi sauts*", Mémoire pour Maitrise, Ecole Supérieure, université du Québec, Date : 7 Aout 2009 .

[76] R, BRADEN. D, CLARK et S, SHENKER, "*Integrated Services in the Internet Architecture : an Overview*", Internet Request For Comments RFC 1633, June, 1994.

[77] I. NIANG, D. SERET, "*Dimensionnement de DiffServ basé sur des métriques de performance*" , CARI02, Yaoundé, 2002.

[78] C. CHASSOT, "*Contribution aux protocoles et aux architectures de communication de bout en bout pour la QdS dans l'internet*", Mémoire d'Habilitation a Diriger des Recherches, Institut National Polytechnique de Toulouse, France, Soutenue au LAAS-CNRS, le 12 décembre, 2005.

[79] J.WROCLAWSKI, "*Specification of the Controlled-Load network element service*, RFC 2211, September, 1997.

[80] S. SHENKER, C. PARTRIDGE, et R. GUÉRIN, "*Specification of Guaranteed quality of service*", RFC 2212, September, 1997.

[81] R. BRADEN, L. ZHANG, S. BERSON, S. HERZOG, S. JAMIN, "*Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*", RFC 2205, September, 1997.

[82] S. MARTIN, "*Maitrise de la dimension temporelle de la qualité de service dans les réseaux*", Thèse de doctorat en sciences de l'Université PARIS XII, 6 juillet, 2004 .

[83] L.ZHANG, et al, "*RSVP : A New Resource reservation Protocol*", IEEE Network Magazine, pp. 8-18, September, 1993 .

[84] J. WROCLAWSKI, "*The use of RSVP with IETF Integrated Services*", RFC 2210, September, 1997.

[85] S. BLAKE, D. BLACK, M. CARLSON, E. DAVIES, WANG, et W. WEISS, "*An Architecture for Differentiated Services*", RFC 2475, December 1998.

[86] J. HEINANEN, F. BAKER, W. WEISS, J. WROCLAWSKI, "*Assured Forwarding PHB Group*", RFC 2597, IETF : The Internet Engineering Task Force, http ://www.ietf.org, June, 1999.

[87] V. JACOBSON, K. NICHOLS, et K. PODURI, "*An Expedited Forwarding PHB Group RFC 2598*", June,1999 .

[88] K . NICHOL, S. BLAKE, F. BAKER, et D. BLACK, "*Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*", RFC 2474, Décembre, 1998.

[89] C. CHAUDET, "*Routage QoS et réseaux ad-hoc : état de l'art*" , Technical report, LIP-ENS Lyon, Oct. 2002 .

[90] H. BADIS, "*Étude et conception d'algorithmes pour les réseaux mobiles et Ad hoc*", Thèse de doctorat ; Université Paris-Sud, Soutenue: Décembre, 2005.

[91] H. XIAO, K. G. SEAH, A. LO, et K.C. CHUA, "*A flexible quality of service model for mobile Ad hoc Networks*", Vehicular Technology Conference Proceedings, 2000, VTC, 2000-Spring Tokyo, IEEE : 51st, Volume : 1, Page(s) : 445 -449.

[92] G. AHN, A.T. CAMPBELL, A. VERES et L. SUN, "*SWAN : Service Differentiation in Stateless Wireless Ad hoc Networks*", in IEEE INFOCOM02, vol. 1, (New york, USA), pp. 457-466, May, 2002 .

[93] C. SARR, S.KHALFALLAH et I. Guéerin LASSOUS, "*Gestion dynamique de la bande passante dans les réseaux Ad hoc multi-sauts*", JDIR'09: 10émes Journées Doctorales en Informatique et Réseaux, 2009.

[94] H. BADIS, "*CEQMM: A Complete and Efficient Quality of service Model for MANETs*", The Third ACM International Workshop on Performance Evaluation of Wireless Ad hoc PE-WASUN, Spain, Oct, 2006 .

[95] H. BADIS, K. AL AGHA, *QOLSR, "QoS routing for Ad hoc Wireless Networks Using OLSR*", European Transactions on Telecommunications, Vol. 15, No. 4, pp. 427-442, 2005 .

[96] Y. YANG, R. KRAVETS, "*Distributed QoS Guarantees for Realtime Traffic in Ad hoc Networks*", In IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON), pages 118- 127, Oct, 2004.

[97] K. CHEN, S.H. SHAH, et K. NAHRSTEDT, "*Cross Layer Design for Data Accessibility in Mobile Ad hoc Networks*", in Wireless Communications, vol. 21, pages: 49-75, New york, USA, 2002 .

[98] N. NIKAEIN, C. BONNET, Y. MORET, et I.A. RAI, "*2LQoS- Two-Layered Quality-of-ServiceModel for Routing in Mobile Ad hoc Networks*", Institut Eurécom, 06904 Sophia Antipolis, France .

[99] P. KARN, "*MACA : new channel acces method for packet ratio*", in proceeding of ARRL/CRRL Amateur radio 9th Computer Networking Conference 1990 .

[100] V. BHARGHAVAN et al, *MACAW: A media acces protocol for wirless LANs*", in proceeding ACM Special Group Communication conference (SIGCOMM'94), August, London, UK, 1994.

[101] "*IEEE 802.11e draft/D4.1, Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)* " , Feb. 2003.

[102] I. AAD, C. CASTELLUCCIA, "*Differentiation mechanisms for IEEE 802.11*", In Proceedings of IEEE INFOCOM'01, Anchorage, Alaska, Avril 22- 26, 2001.

[103] J. DENG, R.S. CHANG, "*A priority scheme for IEEE 802.11 DCF access method*", IEICE Transactions in Communications, vol 82-B, n 1, janvier 1999.

[104] C. MRABET, "*Amélioration du service Alternative Best Effort (ABE-α and WABE) pour le transport de flux multimédias sur les réseaux Ad hoc*", Diplôme des Etudes Approfondies, Ecole Nationale des Sciences de l'Informatique, Université de la Manouba, 12 Juillet, 2004.

[105] N. H. VAIDYA, P. BAHL, et S. GUPTA, "*Distributed fair scheduling in wireless LAN*", In Sixth Annual International Conference on Mobile Computing and Networking, Boston, USA August 2000.

[106] Y. DRABU, "*A Survey of QoS techniques in 802.11*", Department of Computer Science, Kent State University, Available at: http://trident.mcs.kent.edu/ ydrabu/research /wmac/mac.pdf .

[107] J. L. SOBRINO, A. S. KRISHNAKUMAR, "*Quality of Service in ad-hoc carrier sense multiple access Networks*", IEEE Journal on Selected Areas in Communications, 17(8) pages:1353-1368, August 1999 .

[108] A.BRANCHS, X. PEREZ, "*Providing Throughput Guarantees in the IEEE 802.11 Wirless LAN*", in proceeding of IEEEE WCNC'02, Orlando, USA, March 2002 .

[109] C.R. LIN, M. GERLA, "*MACA/PR : An Asynchronous Multimedia Multihop Wireless Network*", In Proceeding of IEEE INFOCOM'97, April 1997 .

[110] "*http://www.ieee802.org/1/pages/802.1D.html*", date consultation: Mai 2010.

[111] S-B. LEE, G-S AHN, X. ZHANG, et A.T. CAMPBELL, "*INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad hoc Networks*", Journal of Parallel and Distributed Computing (Academic Press), Special issue on Wireless and Mobile Computing and Communications, Vol. 60 No. 4, pp.374-406, April, 2000 .

[112] C. CHAUDET, I. GUÉRIN LASSOUS, "*BRuIT : Bandwidth Reservation under InTerferences influence*", European Wireless 2002 (EW2002), Florence, Italy, février, 2002 .

[113] P. MOHAPATRA, J. LI, et C. GUI, "*QoS in mobile Ad hoc Networks*", University of California .

[114] L. HANZO, R. TAFAZOLLI, "*A survey of QoS routing solutions for mobile Ad hoc Networks*", IEEE Communication. Surv. and Tutor., vol. 9, no. 2, pages: 50-70, 2007 .

[115] P. BRECKER, "*QoS Routing Protocols for Mobile Ad-hoc Networks: A Survey*", Technical Report 368/08; No. D2.5.1 , Fraunhofer IESE and TU Kaiserslautern, Date: 6. August 2007 .

[116] P. SINHA, R. SIVAKUMAR et V. BHARGHANAN, "*CEDAR: a Core-Extraction Distributed Ad-Hoc Routing Algorithm*", IEEE Journal on Selected Areas in Communications, Vol. 17, No. 8, August, 1999 .

[117] H. BADIS, A. Munaretto, K.A. AGHA, "*Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR)* ", INRIA, LIP6 France Internet-Draft, 14 October, 2004 .

[118] A. MUNARETTO, H. BADIS, K. AL AGHA et G. PUJOLLE, "*QOLSR : Routage avec QoS dans OLSR*", IEEE MWCN: International Workshop On Mobile and Wireless Communications Networks , Banyuls-sur-mer, France, May 2003.

[119] S. CHEN; K. NAHRSTEDT, "*Distributed quality-of-service routing in Ad hoc networks*", IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad hoc Networks, 17(8):1488-1505, august, 1999.

[120] C. CHAUDET, "*Qualité de service et réseaux ad-hoc : un état de l'art*", Rapport de recherche, INRIA, ISSNN 0249-6399 Numéro 4325 - 12 novembre, 2001.

[121] Q. XUE, A. GANZ, "*Ad hoc QoS on-demand routing (AQOR) in mobile Ad hoc networks*", Journal of Parallel and Distributed Computing, 63(2):154-165, 2003.

[122] Y.-B. Ko, N.H. VAIDYA, "*Location-Aided Routing (LAR) in mobile Ad hoc networks. Wireless Networks*", 6(4):307-321, 2000.

[123] S.B. UDAY, K. KARIBASAPPA et V.D. MYTRI, "*Entropy based QOS Routing for MANET*", International Journal on Recent Trends in Engineering & Technology, Vol. 05, No. 01, Mar 2011.

[124] S.JAMALI, B.SAFARZADEH et H.ALIMOHAMMADI, "*SQR-AODV: A stable QoS-aware reliable on-demand distance vector routing protocol for mobile ad hoc networks*", Scientific Research and Essays, Vol. 6(14), pp. 3015-3026, 18 July, 2011, Available at http://www.academicjournals.org/.

[125] S. SOUNDARARAJAN, R.S. BHUVANESWARAN, "*Multipath Routing Backbone for Improving QoS in Mobile Ad hoc Networks*", European Journal of Scientific Research, Vol.53 No.2, pp.222-230, http://www.eurojournals.com/ejsr.htm, date consultation 09/2011.

[126] M.K. MARINA, S.R. DAS, "*Ad hoc on-demand multipath distance vector routing*", ACM SIGMOBILE Mobile Computing and Communications Review, Vol.6, No.3, July 2002.

[127] Y. WANG, P. REN, et G. WU, "*Throughput-aimed MAC Protocol with QoS Provision for Cognitive Ad Hoc Networks*", IEICE Transaction Communication, Vol.E93-B, N°.6, JUNE 2010.

[128] N. SARMA, S. NANDI, "*A Multipath QoS Routing with Route Stability for Mobile Ad Hoc Networks*", Journal: IETE Technical Review ISSN 0256-4602, Vol 27, issue 5 pp 380-397, 11-aug -2010.

[129] P.K. SURI, M. K.SONI et P. TOMAR, "*Cluster Based QoS Routing Protocol for MANET*", International Journal of Computer Theory and Engineering, Vol. 2, No. 5, October, 2010; pp. 1793-8201.

[130] V. RISHIWAL, S. VERMA, et S.K. BAJPAI, "*QoS Based Power Aware Routing in MANETs*", International Journal of Computer Theory and Engineering, Vol. 1, No. 1, pp.1793-8201, April 2009.

[131] S. UPADHAYAYA, C. GANDH, "*Qos routing using link and node stability in mobile Ad hoc Networks*", Journal of Theoretical and Applied Information Technology, Vol. 8. No.2; 31October 2009.

[132] F. QIN, Y. LIU, "*Multipath Based QoS Routing in MANET*", Journal of Networks, Vol. 4, N° 8, October 2009.

[133] N. SARMA, S. NANDI, "*A Cross-layer QoS Mapping Framework for Mobile Ad Hoc Networks*", Journal: IETE Technical Review, Volume: 25; Issue: 6; Start page: 346; 2008.

[134] I-Sheng LIU, F. TAKAWIRA et H. JUN XU, "*A Hybrid Token-CDMA MAC Protocol for Wireless Ad Hoc Networks*", IEEE Transaction on Mobile Computing, Vol. 7 No. 5, pp. 557-569; May 2008.

[135] C. PERKINS, E. BELDING-ROYER, S. DAS, "*Ad hoc On-Demand Distance Vector (AODV) Routing*", Request For Comments 3561, Network Working Group, http://www.ietf.org/rfc, Juillet 2003

[136] H. BOUKOUNA, "*Adaptation d'un protocole de découverte de services pour les réseaux Ad hoc*", 2008.

[137] D. ELORRIETA, "*Protocoles de routage pour l'interconnexion des réseaux Ad-Hoc et UMTS*", Université libre de Bruxelles, 2007.

[138] F. AMEZA, "*Le routage dans les réseaux Ad hoc (OLSR et AODV)*", Université de BJAIA 2007.

[139] A. BENOIT, "*Notes de cours master1 (Chapitre 3 : Réseaux sans fil)*", ENS Lyon, 2006.

[140] N. MANSOURI, "*Protocole de routage multichemin avec équilibrage de charge dans les réseaux mobiles Ad hoc*", Ecole Supérieur Des Communication De Tunis 2007.

[141] M.N OUNES, K. LAGGOUN, "Etude et simulation sous ns du protocole AODV dans les réseaux mobiles", mémoire d'ingénieur, Université de Batna, Promotion Juin 2007.

[142] M.SEDRATI, C.BOULKAMH, M.BENMOHAMMED, "*AODV-SR: une variante AODV pour l'Amélioration de la Qualité de Service dans le Routage* ", CRATT'2009 : Colloque de recherche Appliquée et de Transfert de Technologie ISET Radès 11-12 Novembre 2009.

Site: http://www.isetr.rnu.tn/cratt/Programme/programme.html

[143] M.SEDRATI, A.BILAMI, M.BENMOHAMMED, " *M-AODV: AODV variant to Improve Quality of Service in MANETs* ", IJCSI: International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011, Pages 429-436. ISSN (Online): 1694-0814, www.IJCSI.org.

[144] M.A. CHENNA, M. N. BARECHE, "Amélioration de la QdS avec le protocole AODV", mémoire d'ingénieur, Université de Batna, Promotion Juin 2009.

[145] C. YAWUT, "*Adaptation a la mobilité dans les réseaux Ad hoc*", Thèse de Doctorat, Université de Toulouse Soutenue le 28/09/2009.

[146] *M.SEDRATI*, A.BILAMI, R.MAAMRI, M.BENMOHAMMED, "*Contention Window Optimization for Distributed Coordination Function (DCF) to improve Quality of Service at MAC layer*", International Conference on Digital Information and Communication Technology and its Applications (DICTAP2011); Bourgogne, France 21-23 juin 2011, Proccedings (Eds.): DICTAP 2011, Part I, CCIS 166, pp. 704–713, 2011, *Springer-Verlag* Berlin Heidelberg 2011, *Lecture Notes in Computer Science.*

[147] A. ABDELAZIZ, W. MEDDOUR, "Amélioration de la QdS au niveau de la couche MAC 802.11", mémoire d'ingénieur, Université de Batna, Promotion Juin 2009.

[148] K. FALL, K. VARADHAN, "*The NS manual* ", *The VINT project*, 2003

[149] "*Tuorial AWK* ", *Par Nyal*, 10 Janvier 2005. http://nyal.developpez.com/tutoriel/gawk/gawk.pdf.