République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Saad-Dahleb BLIDA/Département d'Informatique

Mémoire du projet de fin d'étude
Pour l'obtention du diplôme
d'Ingénieur d'Etat en Informatique

Sujet :

Simulation et Comparaison des Protocoles MAC dans les Réseaux Mobiles Ad hoc

Thème proposé et encadré par :

Monsieur DJENOURI Djamel

Suivi par :

Mademoiselle BOUSTIA Narimane

Etudié par :
BOUZENADA Mohamed Fateh
KHIAT Faten

Promotion 2003/2004

DEDICACES (BOUZENADA)

Aux êtres les plus chers au monde. A mes parents.

A mes frères et sœurs.

A Leila qui m'a beaucoup aidé.

A mon Encadreur qui ma fait aimer le domaine de la recherche.

A tous mes professeurs.

A tous mes amis.

DEDICACES (KHIAT)

Aujourd'hui est un grand jour, il marque un tournant qui n'est pas des moindre, le cap qui sépare la vie d'étudiant de la vie d'adulte responsable et j'espère actif. Aujourd'hui nous sortirons de cette assemblé diplômé prêt à entamer la vie des grands comme l'on précédemment fait nos parents, plein d'espoirs pour une vie remplie de sucés, en espérant une vie meilleur que celle de nos chère parents, nous rendant plus fort pour nous et nos proches.

Je profite de l'occasion pour dédier ce modeste travail à ma chère mère qui ma largement soutenu et écouter durant mon cursus, à mon défunt père qui est à l'origine du choix de mes études, et qui à ma grande peine n'est pas la pour apprécier le résultat. Je dédie aussi ce travail à mes deux frères en leur souhaitant un parcourt plus stimulant et plus de réussite. Ainsi qu'à toutes ma famille oncles, tantes, cousins et cousines.

Je profite aussi de l'occasion pour exprimer des remerciements à tous ceux qui m'on soutenue surtout ces deus dernières années et en particulier mes amis : Djamel, Mohamed, Karim, Amel, Selma, et Nadjet.

Sans oublier tous mes camarades en leur souhaitant plein de réussite et un très bon courage.

Mercie.

REMERCIEMENTS

Monsieur DJENOURI Djamel envers qui nous éprouvons un grand respect a été l'acteur principal dans la réalisation de ce travail. Nous tenons à le remercier du fond du cœur.

Comme nous exprimons toute notre gratitude envers Mademoiselle BOUSTIA qui n'a ménagé aucun effort dans notre suivi.

Nos remerciements vont également à tous les Professeurs qui nous ont encadrés le long de notre cursus universitaire et à l'équipe administrative du CERIST.

Nous profitons de l'occasion pour remercier Monsieur Jean-Sébastien DUPUY, Ingénieur chez MICROSOFT France qui nous a aidés à résoudre un BUG que nous avons rencontré dans VC++.

Table des matières

Liste des Figures

Liste des Tableaux

Introduction :

Les technologies actuelles en matière de réalisation de composants électroniques, et en particulier de microprocesseurs, permettent de développer des équipements de taille et de poids de plus en plus réduits. Cela a permis l'apparition d'unités informatiques mobiles de plus en plus puissantes, tels que les ordinateurs portables (Laptops) et les assistants personnels (PDA[1]). De plus, les technologies réseaux ayant pris beaucoup d'ampleur ces dernières années, et avec l'augmentation des besoins des utilisateurs, ces unités mobiles commencent à être munies de moyens de communication, ce qui leur permet de communiquer entre eux, en formant un réseau mobile. En comparaison avec l'ancien environnement (l'environnement statique), le nouvel environnement résultant appelé l'environnement mobile permet aux unités de calcul une libre mobilité et ne pose aucune restriction sur la localisation des usagers. À part ces avantages, la mobilité engendre quelques inconvénients propres à l'environnement mobile, notamment une fréquente déconnexion, un débit de communication et des ressources modestes.

Les réseaux mobiles sans fil peuvent être classés en deux catégories. Les réseaux avec infrastructures, utilisent généralement le modèle de communication cellulaire, une importante logistique et infrastructure fixe est nécessaire pour le déploiement d'un tel réseau, le réseau GSM en constitue un exemple. La deuxième catégorie de réseaux mobiles est les réseaux sans infrastructures, qui sont appelés les réseaux ad hoc. Leur caractéristique principale est qu'ils ne requièrent aucune infrastructure fixe ou administration centrale, et toutes les unités du réseau sont mobiles. Aucune supposition ou limitation n'est faite sur la taille du réseau, ce qui implique la possibilité que le réseau ait une taille énorme.

Le medium sans fil est partagé entre tous les utilisateurs du réseau, pour assurer un partage équitable entre les mobiles, et pour éviter les accès multiples (collisions), une stratégie ou protocole de contrôle d'accès au canal (MAC) est nécessaire. Plusieurs protocoles MAC on été proposés pour les communication sans fil, ces protocoles sont adapté dans l'environnement ad hoc.

---

[1] Personnel Digital Assistant

Notre travail entre dans le cadre de l'étude des protocoles MAC dans les réseaux ad hoc. Il consiste à comparer un certain nombre de protocoles et d'évaluer leurs performances en recourant à la simulation. Nous étudierons les protocoles les plus connus, à savoir CSMA, 802.11, et MACA qui sont implémentés dans le Simulateur GloMoSim. Ces protocoles sont tous basés sur une approche orientée émetteur (Sender-Initiated), c'est-à-dire qu'il revient à l'émetteur d'initier la communication en demandant la permission de transmettre au récepteur. Dans notre travail, nous avons implémenté un nouveau protocole basé sur l'approche inverse, dite approche orientée récepteur (Receiver-Initiated), dans laquelle le récepteur scrute ses voisins en leur demandant s'ils ont des paquets de données à transmettre. Nous avons choisi ce protocole pour pouvoir réaliser, en plus de la comparaison des protocoles, une comparaison des deux approches. Notre travail se compose de quatre chapitres.

Le premier chapitre, plutôt introductif, sera consacré à l'étude des réseaux mobiles ad hoc. Avant de les définir, nous présenterons les deux catégories d'environnements mobiles pour aboutir aux environnements sans infrastructure ou ad hoc, dont nous évaluerons les propriétés en les comparant à celles des réseaux cellulaires (réseaux avec infrastructures). Nous y présenterons également la façon de modéliser un réseau Ad hoc, ses principales caractéristiques ainsi que ses différentes applications. Concernant les challenges, problèmes à étudier sérieusement avant la mise en place de ce genre de réseau, nous nous contenterons de les citer pour en reprendre deux dans le détail, à savoir, le problème de routage, celui-ci ayant constitué le problème majeur à cause de la topologie dynamique du réseau, et le problème de contrôle d'accès au canal, objet de notre recherche.

Le deuxième chapitre, aura pour objet l'étude des protocoles MAC. Il contiendra une classification détaillée de ces protocoles, basée sur plusieurs critères; un exposé des problèmes d'accès au canal; et enfin, une études proprement dite, abordant chaque protocole à part.

Le troisième chapitre, nous le consacrerons à l'environnement de simulation où nous réaliserons notre étude. Nous introduirons dans sa première partie des concepts généraux ayant trait à la simulation. Nous présenterons, dans la seconde, le langage de simulation par lequel est développé le simulateur, à savoir le langage PARSEC que

nous avons utilisé pour implémenter le protocole MACA-BI ainsi que les métriques de simulation utilisées. La troisième, enfin, nous la consacrerons au simulateur GloMoSim.

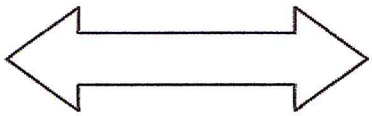
Le quatrième chapitre comprendra les analyses et les résultats de notre simulation.

CHAPITRE I : LES RESEAUX MOBILES AD HOC

I.1 Introduction

Les réseaux sans fil ont connu durant la dernière décennie une utilisation croissante auprès du grand public. En effet, leur utilisation favorise l'accès distant aux différents types d'information (vidéo, audio,...) à partir d'équipements variés et notamment mobiles. Ces équipements ont des caractéristiques particulières (puissance de traitement limitée, capacité de stockage modeste et ressource d'énergie autonome), et accèdent au réseau à travers des interfaces de communication sans fil.

Ainsi l'évolution de la communication sans fil a poussé les chercheurs à tenter de réaliser la plus grande ambition des réseaux :



"L'accès à l'information n'importe où et n'importe quand".

Accéder à l'information à n'importe quel moment est déjà un acquis, reste alors la contrainte du lieu à surmonter, la création des réseaux cellulaires va justement dans cette direction, mais elle comporte certaines restrictions. La nécessité et la dépendance d'une infrastructure en sont les plus importantes. Les chercheurs ont pensé alors à surmonter ces limites, ce qui a donné naissance au réseau ad hoc ; étant un réseau totalement mobile et ne présentant aucune limite d'infrastructure liée à la localisation géographique des nœuds.

Toutes sortes de question se posent dés lors à l'égard de ce réseau ad hoc :
Qu'est ce qu'un réseau ad hoc? Quel est le profil du trafic dans ce type de réseau ? En quoi un réseau ad hoc est différent du réseau cellulaire ? Comment modélise t-ons un réseau ad hoc? Quelles en sont les caractéristiques ? Ou trouve t-on les réseaux ad hoc? Quels sont les défis à relever pour réaliser un réseau ad hoc? Quels sont les problèmes que l'on rencontrera afin de le mettre en place?

Dans ce chapitre nous essayons de répondre à toutes ces questions afin de donner une idée détaillée sur le réseau ad hoc.

I.2 Le modèle des environnements mobiles

Un environnement mobile est un système composé de sites mobiles, il permet à ses utilisateurs d'accéder à l'information indépendamment de leurs positions géographiques. Les réseaux mobiles ou sans fil peuvent être classés principalement en deux classes : les réseaux avec infrastructure et les réseaux sans infrastructure.

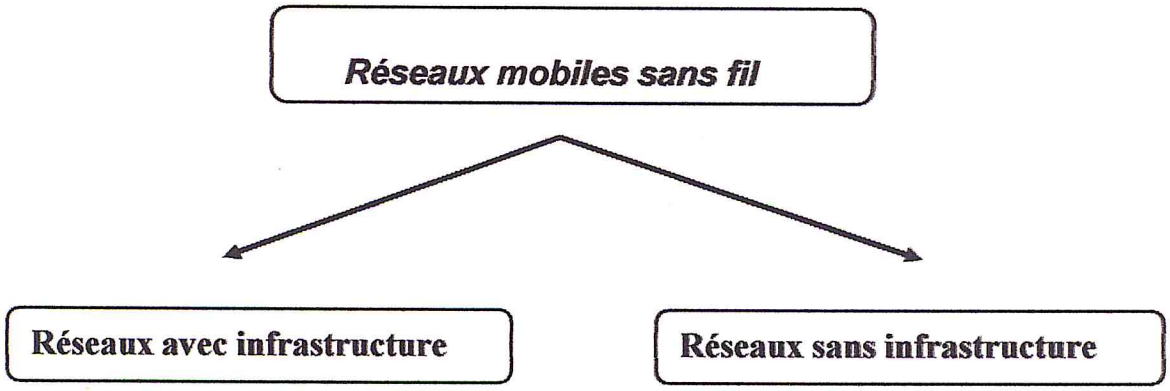


Figure I.1 : La décomposition des réseaux mobiles

Les réseaux avec infrastructure (cellulaires) sont composés à leur tour de deux types de sites.

1- Les sites fixes nommés aussi stations de base (SB), forment un réseau fixe, généralement filaire, qui représente la colonne vertébrale du réseau. Ces sites sont munis d'une interface de communication sans fil, établissant une liaison directe avec les sites mobiles localisés dans une zone géographique limitée appelée cellule.

2- Les sites mobiles sont les nœuds qui composent le réseau. Ils communiquent entre eux en faisant appel aux SB qui établissent un chemin de communication sans fil entre ces nœuds ou entre un nœud et une autre SB.

A chaque station de base correspond une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages. Alors que les sites fixes sont interconnectés entre eux à travers un réseau de communication filaire, généralement

fiable et d'un débit élevé, les liaisons sans fil ont une bande passante limitée qui réduit sévèrement le volume des informations échangées [Duc92].

Dans ce modèle, une unité mobile ne peut être, à un instant donné, directement connectée qu'à une seule station de base. . Elle peut communiquer avec les autres sites à travers la station à laquelle elle est directement rattachée. L'autonomie réduite de sa ressource d'énergie lui occasionne de fréquentes déconnexions du réseau; sa reconnexion peut alors se faire dans un environnement nouveau voire dans une nouvelle localisation. Le réseau cellulaire GSM reste l'un des exemples les plus expressifs de cette classe.

Les particularités des réseaux cellulaires (sites fixes, sites mobiles et le lien entre eux) sont absentes dans les réseaux sans infrastructure (ad hoc). Dans ce cas, deux stations situées dans des zones géographiques différentes nécessitent toujours un ou plusieurs intermédiaires qui ne sont autres que des sites mobiles, ces sites mobiles joueront et le rôle d'hôtes et de l'infrastructure en question. Le réseau ad hoc se base principalement sur cette idée.

Voici une représentation des réseaux infrastructurés et non infrastructurés :

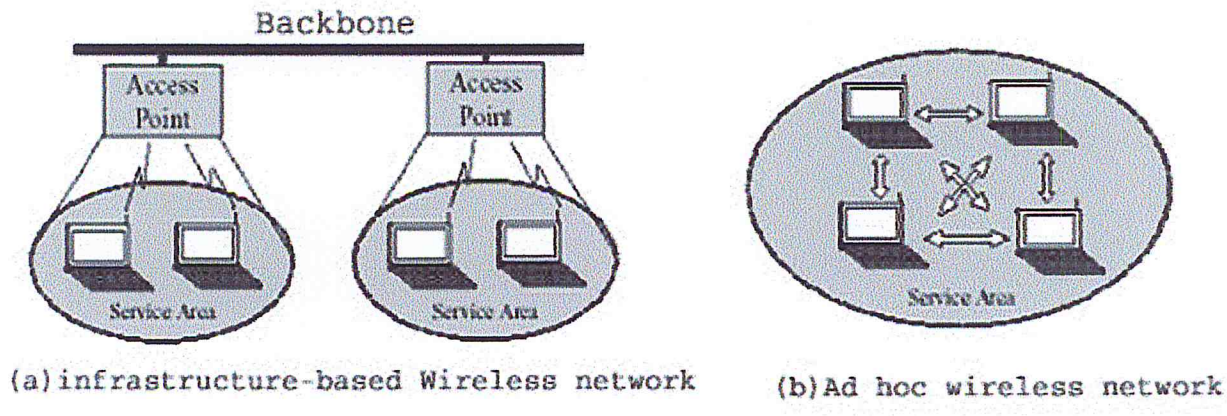


Figure I.2 : réseaux sans fil avec et sans infrastructure

I.3 Définition du réseau ad hoc

Le concept des *réseaux mobiles ad hoc* essaie d'étendre les notions de la mobilité à toutes les composantes de l'environnement. Contrairement aux réseaux basés sur la communication cellulaire, aucune administration centralisée n'est disponible. Ce sont les hôtes mobiles eux-mêmes qui forment, d'une manière *ad hoc*, une infrastructure du réseau. Aucune supposition ou limitation n'est faite sur la taille du réseau ad hoc, celui-ci pouvant théoriquement contenir des centaines ou des milliers d'unités mobiles. Mais qu'est ce qu'un réseau *ad hoc*? Nous trouvons dans la littérature différentes définitions, nous en avons retenu quelque unes :

Un réseau ad hoc est une collection d'unités mobiles, qui peuvent dynamiquement communiquer (les unités communiquent alors qu'elles sont en mouvement) n'importe ou et n'importe quand sans besoin d'une infrastructure préexistante. Autrement dit, c'est un système autonome dans lequel les hôtes mobiles connectés par des liaisons sans fil sont libre de se déplacer aléatoirement, et souvent agissent comme des routeurs en même temps. [Sun01]

Les réseaux ad hoc sont des réseaux hertziens qui se déploient facilement voire automatiquement entre personnes souhaitant communiquer entre elles, sans qu'il y est besoin de développer toute une infrastructure pour y parvenir.
Fonctionnant sur le principe du peer to peer, il suffit que deux terminaux soient à proximité pour qu'ils puissent communiquer entre eux si besoin. [site1]

Un réseau mobile ad hoc appelé aussi MANET (Mobile ad hoc NETworks) est une collection de terminaux mobiles (nœuds) équipés d'interfaces de communication sans fil. Cela forme un réseau autonome sans l'aide d'une infrastructure fixe. Un nœud peut transmettre directement des paquets de données à d'autre nœuds se trouvant sur sa porté, si le nœud destinataire est hors de porté de transmission, le réseau utilise le routage multi-saut (multi-hop) pour assurer l'acheminement des paquets.[Poz02]

Un réseau mobile ad hoc est un réseau sans fil, sans station de base ni aucune infrastructure filaire backbone. Les nœuds utilisent la transmission de paquets en mode peer to peer, et un routage Multi-Saut, pour communiquer entre eux. La topologie du réseau est dynamique ; cela est dû aux mouvements fréquents des nœuds. Par conséquent, des protocoles de routage dynamique sont nécessaires pour

l'établissement et le maintien des chemins entre la paire de nœuds communiquant [Nas00].

En définitif, un réseau ad hoc ou MANET est un réseau mobile sans fil, dont les unités forment la structure et l'infrastructure du réseau. Ce réseau exploite les ondes radio de courte portée pour établir les liaisons de communication (Short Distance Wireless).

Il se distingue des réseaux câblés et des réseaux cellulaires par sa mobilité (tous les sites du réseau sont mobiles), sa topologie dynamique, et le routage multi-saut de ses paquets.

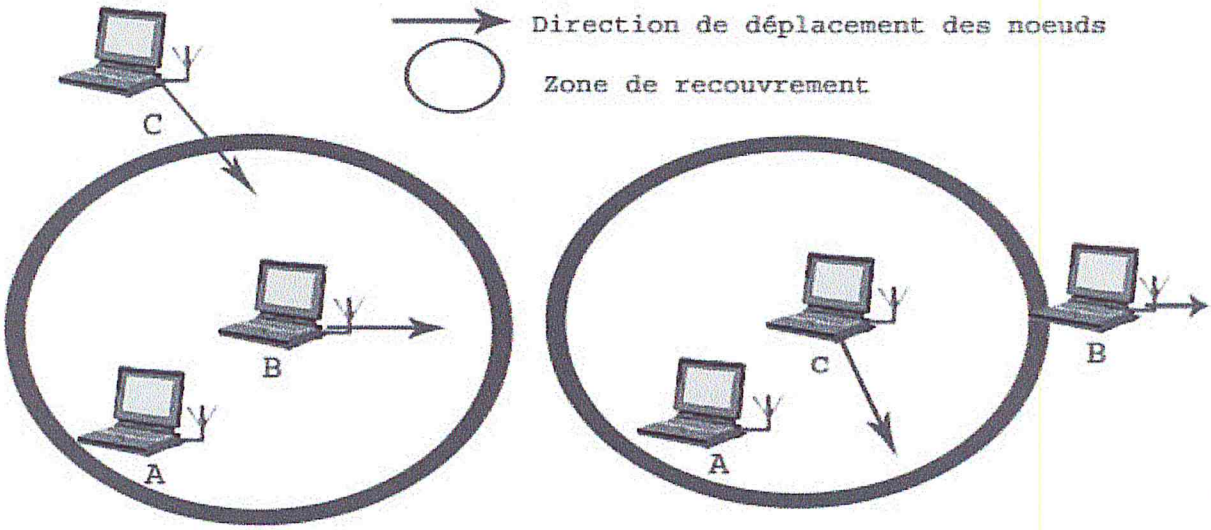


Figure I.3 : Représentation d'un réseau ad hoc

I.4 Formes du trafic dans un réseau ad hoc

La communication dans un réseau sans fil ad hoc prend plusieurs formes de trafic différentes [Sun01] :

- Peer-to-Peer : la communication entre deux noeuds étant en un saut, le trafic du réseau est souvent consistant.
- Remote-to-Remote : la communication est établie entre deux noeuds au delà d'un saut mais elle est maintenue sur un chemin stable entre eux. Ceci peut être le résultat de la communication entre plusieurs noeuds situés dans une même zone géographique se déplaçant en groupe.

- Trafic Dynamique : ceci se produit lorsque plusieurs noeuds sont dynamiques et se déplacent indépendamment les un des autres. Les chemins doivent être reconstitues, il en résulte une faible connectivite.

I.5 Réseau Cellulaire et Réseau ad hoc

Pour mieux évaluer les propriétés d'un réseau ad hoc, il est utile de les comparer avec celles du réseau GSM (réseau cellulaire équipé de stations de base fixes (SB) et de terminaux mobiles (MT)).

- ✓ Dans un réseau cellulaire, il est nécessaire d'organiser la zone de couverture et la capacité requise pour y parvenir à priori ; par contre dans un réseau ad hoc la zone de couverture est entièrement dépendante du nombre de nœuds présent et de leur distribution. Sachant que la zone de couverture et la distribution sont deux critères dynamiques par nature, il n'est donc pas possible de les organiser à priori.
- ✓ Les ressources telles que la largeur de bande passante, la capacité de stockage, et la durée de vie des batteries sont limitées pour tous les nœuds dans le réseau ad hoc ; différemment aux SB utilisées dans les réseaux cellulaires.
- ✓ La sécurité du réseau est un important soucie pour les réseaux ad hoc à cause de la mobilité des nœuds, la nature des liaisons sans fil et surtout l'absence d'un contrôle centralisé, cette dernière difficulté n'existe pas dans les réseaux cellulaires ou il y a des SB.

I.6 Modélisation d'un réseau ad hoc

Un réseau ad hoc dont les nœuds ont le même domaine de puissance peut être modéliser par un graphe non orienté $G_t = (V_t, E_t)$ où V_t représente l'ensemble des nœuds (i.e. les unités ou les hôtes mobiles) du réseau et E_t modélise l'ensemble des connexions qui existent entre ces nœuds (voir la figure I.4). Si $e = (u, v) \in E_t$, cela veut dire que les nœuds u et v sont en mesure de communiquer directement à l'instant t.

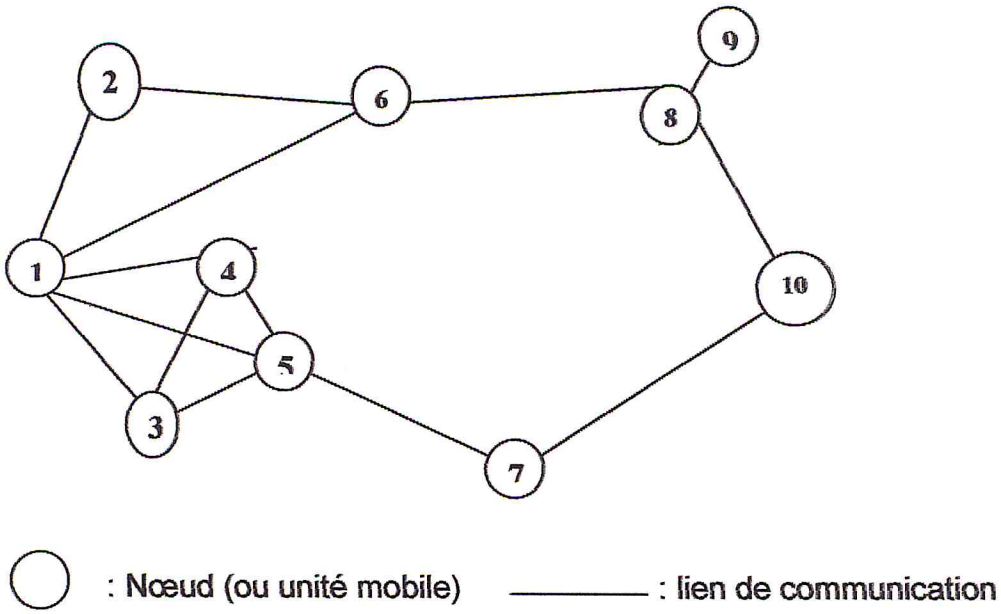


Figure I.4 [Dja01] : La modélisation d'un réseau ad hoc

La topologie du réseau peut changer à tout moment (voir la figure I.5), elle est donc dynamique et imprévisible ce qui fait que la déconnexion des unités soit très fréquente.

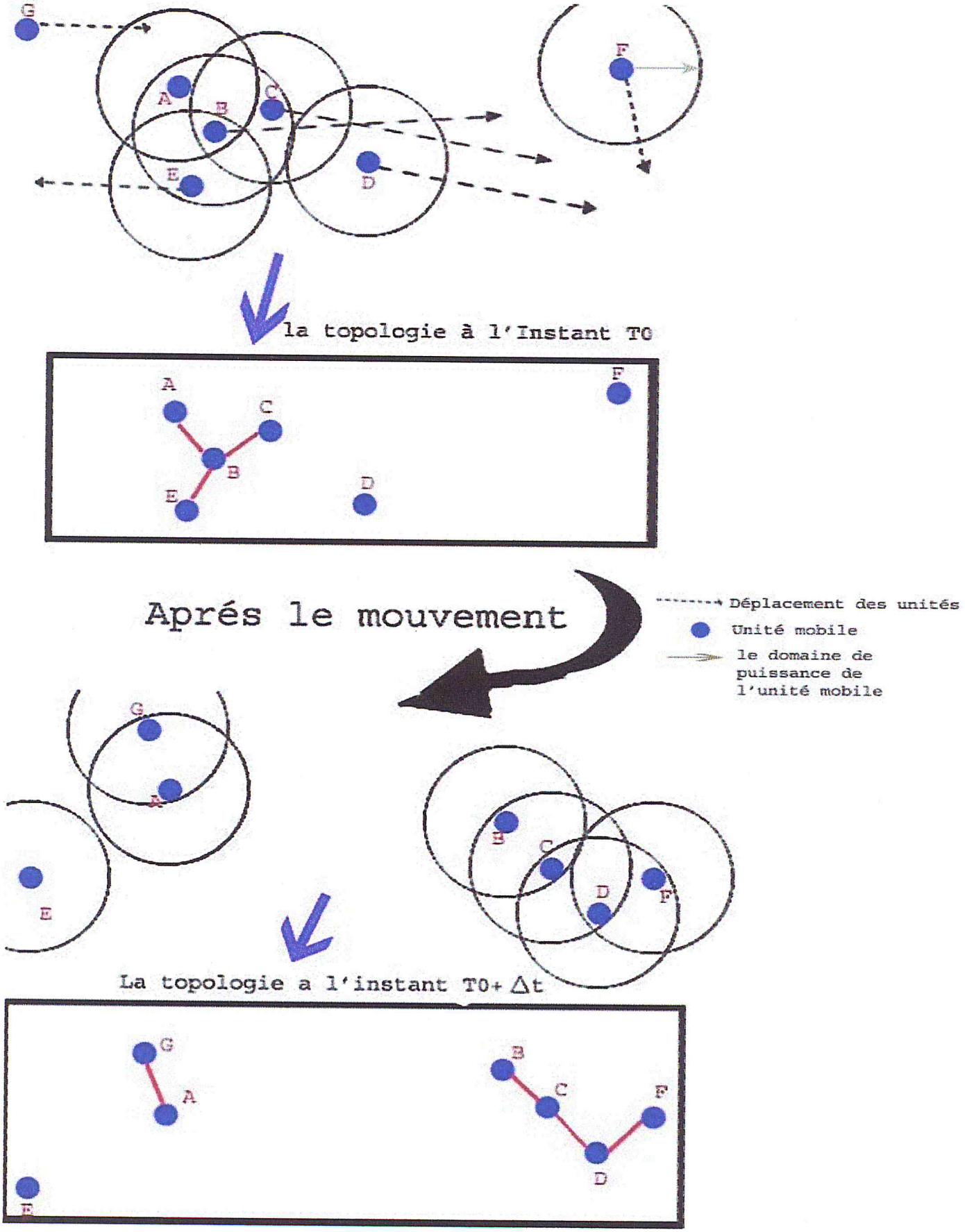


Figure I.5 Mouvement des nœuds dans un réseau ad hoc

I.7 Caractéristiques des réseaux ad hoc

Un réseau ad hoc est caractérisé par ce qui suit :

1) Une topologie dynamique

Les unités mobiles du réseau se déplacent de façon libre et arbitraire. Par conséquent, la topologie du réseau peut changer, à des instants imprévisibles, d'une manière rapide et aléatoire. Les liens de la topologie peuvent être unis ou bidirectionnels, selon les domaines de puissance des nœuds, s'ils sont identiques les liens seront bidirectionnels, et ils sont unidirectionnels dans le cas contraire.

2) L'absence d'infrastructure

Pour résoudre ce problème, chaque terminal mobile est un nœud autonome, ayant les fonctions d'un hôte et d'un routeur. Autrement dit, en plus de sa capacité de traitement en tant que hôte, le nœud mobile doit aussi effectuer les fonctions de l'infrastructure à savoir, le routage.

3) Opérations distribuées

L'absence d'infrastructures préexistante et de tout genre d'administration centralisée, fait que le contrôle et la gestion du réseau sont distribués entre les différents terminaux du réseau. Ces derniers doivent collaborer entre eux pour accomplir les différentes fonctions du réseau (sécurité, routage et accès aux canaux,... etc).

4) Liaisons à débits variables et à bande passante limitée

Une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé, ce partage fait que la bande passante réservée à un hôte est modeste. En plus, le débit réel des communications sans fil, après avoir déduit les effets des accès multiples, du bruit, des interférences, etc, est souvent très inférieur aux taux de transfert maximum de l'interface de communication.

5) Ressources d'énergies limitées

Les hôtes mobiles sont alimentés par des ressources d'énergie autonomes, tel que les batteries ou les autres ressources consommables. Le paramètre d'énergie doit être pris en considération dans tout contrôle effectué par le système.

Le tableau 1 [For94] présente les composantes d'un PDA (Personal Digital Assistant) et leur consommation d'énergie.

Composant	Consommation (Watts)
Système de base (2MB,25MHZ CPU)	3.650
Système de base (2MB,10MHZ CPU)	3.15
Système de base(2MB,5MHZ CPU)	2.8
Ecran backlight	1.425
Moteur de disque dur	1.1
Coprocesseur	0.65
Disquette	0.5
Ecran LCD	0.315
Clavier	0.49
Port parallèle	0.35
Port série	0.30

Tableau I.1 : la consommation d'énergie des composantes d'un PDA

La consommation est importante quand les unités transmettent ou reçoivent les messages à travers une connexion sans fil.

Des études réalisées sur plusieurs interfaces de communications commerciales (wavelan, Metricom et IR) ont montré que leur consommation dans l'état oisif est entre 150 et 170 mW.

DEC Roamabout consomme 5.76 W durant la transmission, 2.88 W durant la réception, 0.35 W dans l'état oisif. Dans plusieurs interfaces de communication commerciales, la quantité d'énergie consommée durant l'émission est égale au double à celle consommée durant la réception.

6) Sécurité physique limitée

Les réseaux mobiles ad hoc sont plus affectés par le paramètre de sécurité, et ils sont plus menacés par les attaques que les réseaux filaires classiques. Cela se justifie par les contraintes et les limitations physiques de ces réseaux.

7) Limitation des Terminaux

Les nœuds d'un réseau ad hoc sont des unités mobiles caractérisées par une capacité de traitement modeste, une mémoire de faible capacité, et une faible puissance de stockage. De telles unités ont besoin d'algorithmes et de mécanismes optimisés pour pouvoir accomplir les traitements et les différentes fonctions de communication.

I.8 Les applications des réseaux ad hoc

Durant ces dernières années, les réseaux Ad hoc ont suscité une attention considérable dans divers domaines notamment le domaine militaire.

Généralement, les réseaux ad hoc sont utilisés dans des applications où le déploiement d'une infrastructure réseau filaire est trop contraignant, soit parce qu'il est difficile à mettre en place, soit parce que la durée d'installation du réseau ne justifie pas le câblage et l'infrastructure à demeurer.

Parmi ces applications nous citons :

1) Le domaine militaire

Les réseaux Ad hoc permettent aux équipements militaires (troupes, véhicules) de se déplacer et de communiquer entre eux dans les champs de bataille en maintenant les informations entre les soldats et les chefs de troupes [Ros00].

2) Opérations de secours

Les opérations de secours se passent dans des régions désastreuses (incendie, inondation, tremblement de terre) où une infrastructure de réseau filaire ou station de base n'existe pas. La communication entre les équipes de secours étant nécessaire [Ros00], une installation d'un réseau Ad Hoc peut résoudre le problème dans ce cas.

3) Contrôle d'environnement

Des petits véhicules équipés de caméras et des détecteurs de son peuvent être utilisés dans une région déterminée afin de collecter un ensemble d'informations et de l'envoyer à travers un réseau ad hoc à une station de traitement, nous pouvons avec cette méthode prévoir la pollution de l'eau par exemple. [Roy99] [Fro00]. Ce type d'application est aussi appelé réseaux de capteurs (Sensor Networks).

4) Au niveau de salles fermés (Indor)

Les réseaux Ad hoc permettent de créer temporairement des réseaux multimédia entre des différentes unités mobiles (Palmtop, PDA, ...) pour l'échange et le partage d'informations entre les participants à une conférence, Réunion, Salle de TP,... etc

Une autre application appropriée a ce niveau peut s'effectuer dans les HOME NETWORKS où les différents équipements échangent des informations (son,

vidéo, alarme). Une des applications possibles dans ce contexte est un réseau de robots qui effectuent plusieurs travaux ménagers (nettoyage, assurance de la sécurité) [Fro00]

Les réseaux ad hoc peuvent être utilisés aussi dans divers environnements enfermé : un stade, un Aéroport,... etc.

5) PAN (Personal Area Network)

Un réseau ad hoc peut être utiliser pour étendre la connectivité des réseaux personnels (de faible porté), dont le but est de constitué des connexions entre des appareils distants de quelques mètres, généralement pour relier des périphériques (imprimante, PDA, appareil photo, ...) à des ordinateurs.

Un réseau ad hoc peut aussi étendre l'accès à Internet ou à d'autre réseaux, à savoir les LAN sans fil (WLAN), le GPRS (réseaux de téléphonie mobile de la 2,5 génération), et l'UMTS (réseaux de téléphonie mobile de la troisième génération). [Sun01]

I.9 Les Challenges des réseaux Ad hoc

Les caractéristiques des réseaux ad hoc introduisent plusieurs challenges qui doivent être étudiés soigneusement avant toute réalisation commerciale. Cela inclut :

1) Le Routage

Etant donné que la topologie des réseaux ad hoc change constamment, l'acheminement des données entre les nœuds du réseau constitue un challenge. Les chemins entre une pair de nœuds quelconque peuvent contenir des sauts multiples, ce qui est plus complexe que la communication à travers un seul saut.

Le routage Multicast constitue un autre challenge pour les réseaux ad hoc, parce que l'arbre multicast n'est pas statique, ceci étant dû au mouvement aléatoire des nœuds dans le réseau. [Sun01]

2) Sécurité et Fiabilité

En plus des vulnérabilités engendrées par les liaisons sans fil, un réseau ad hoc a ses propres problèmes de sécurité dont la cause pourrait être les paquets délivrés par des voisins non qualifiés entre autres causes. Ajoutons a cela la nécessité de nouveaux schémas distribués d'authentification et de distributions de clés et de

certificats due a l'absence d'un contrôle centralisé (le contrôle et la gestion du réseau sont distribuer entre les différents nœuds du réseau). [Sun01]

Les caractéristiques des liaisons sans fil introduisent aussi des problèmes de fiabilité, causé par : la limitation de la porté de transmission sans fil, la nature de diffusion du média sans fil (le problème des nœuds cachés[2] par exemple), les paquets perdus à cause de la mobilité et les erreurs de transmission de données.

3) Consommation d'énergie

La plupart des protocoles réseaux qui existe ne tiennent pas compte de la consommation d'énergie, car ils supposent la présence de stations statiques et de routeurs alimentés par le secteur. Cependant, les équipements mobiles actuels sont principalement alimentés par des ressources d'énergie autonomes (batterie) de capacité limitée (de nos jours, l'autonomie d'une batterie Lithium-Ion est seulement de 2 à 3 heures). Cette contrainte doit être prise en considération lors de la conception des protocoles pour les réseaux mobiles, en particulier, pour les réseaux ad hoc, où chaque équipement mobile doit accomplir en plus de ses fonctions comme hôte, des fonctions de routeur (acheminement des paquets), ce qui engendre une consommation d'énergie plus importante. [Sun01]

4) L'accès au canal

Le média sans fil est un média partagé, ce qui veut dire que n'importe quel nœud à n'importe quel moment peut accéder au canal. Ainsi deux ou plusieurs nœuds peuvent transmettre en même temps, ce qui engendre des collisions. D'où la nécessité d'un protocole MAC (Media Access Control), son rôle étant de contrôler l'accès au canal, d'éviter les transmissions simultanées et d'assurer la fiabilité de transmission.

Contrairement au réseaux cellulaires où le contrôle d'accès au canal est centralisé, étant adapté au niveau des Contrôleurs Centraux (protocoles TDMA et FDMA), dans les réseaux ad hoc, le contrôle d'accès est implémenté au niveau des nœuds et il est basé sur la compétition; c'est-à-dire que chaque nœud doit essayer de devancer les autres pour accéder au canal, et en même temps d'éviter les éventuelles collisions avec ses voisins.

[2] Ce problème sera explicité dans le chapitre suivant.

La mobilité des noeuds, le problème des nœuds cachés, et le problème des nœuds exposés[3] doivent être pris en compte lors de la conception des protocoles MAC dans les réseaux ad hoc. [Toh02]

I.10 Le problème de routage dans les réseaux Ad hoc

Le routage est une méthode d'acheminement d'informations à la bonne destination à travers un réseau de connexion donné. Dans un environnement ad hoc, on ne peut se baser sur un routage du même type que celui utilisé par le protocole IP. Ce dernier est hiérarchisé, basé sur une connaissance de routeurs fixes qui font le lien entre les différents sous réseaux. Ici, on ne dispose pas de balises ou de machines auxquelles on peut se référer pour faire le lien entre deux objets. Un protocole gérant la forte mobilité du réseau est donc nécessaire.

La Figure I.6 illustre un graphe aléatoire modélisant un réseau ad hoc quelconque :

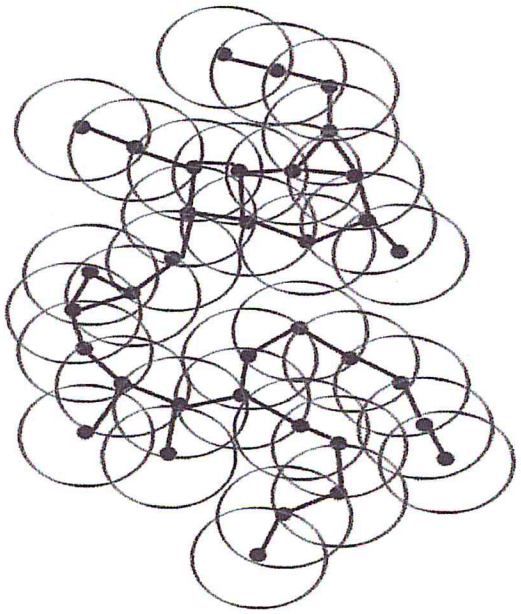


Figure I.6 : maillage aléatoire [Tor02]

Le problème du routage revient à calculer le « meilleur » chemin qui permet de joindre deux nœuds quelconques.

Il s'agit d'un problème de plus court chemin dans un graphe pour le quel il existe plusieurs algorithmes performants (Bellman- Ford et Djikstra sont les plus connus), mais la vrai difficulté est ailleurs. Chaque nœud doit avoir sa table de routage

[3] Ce problème sera expliqué dans le chapitre suivant.

à jour, il faut donc maintenir ces tables. Etant donné la mobilité qui cause le changement fréquent de la topologie, et la limitation des terminaux, cette tache reste assez complexe et cause une surcharge non négligeable dans le réseau. Cet surcharge augmente avec la taille du réseau et la mobilité des nœuds.

Les différents algorithmes de routage proposés ont pour objectif la résolution de ce problème.

Une difficulté dans le choix du critère permettant de dire qu'une route est « la meilleur » est l'affectation d'une métrique « un poids » à chaque arête du graphe, où la longueur d'un chemin est la somme des poids des branches qui le constituent. Plusieurs métriques sont possibles, en particulier :

- Le nombre de branche du chemin.
- La qualité de chaque branche, afin de privilégier les liens de bonnes qualités, sur lesquels on peut y faire passé plus d'informations. De plus si un mécanisme de retransmission est prévu sur chaque lien, choisir un bon lien minimise le nombre de retransmission et donc le délai de transfert.

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en deux principales catégories : les protocoles proactifs et les protocoles réactifs. Une autre catégorie consiste à hybrider les deux premières. [Ris03]

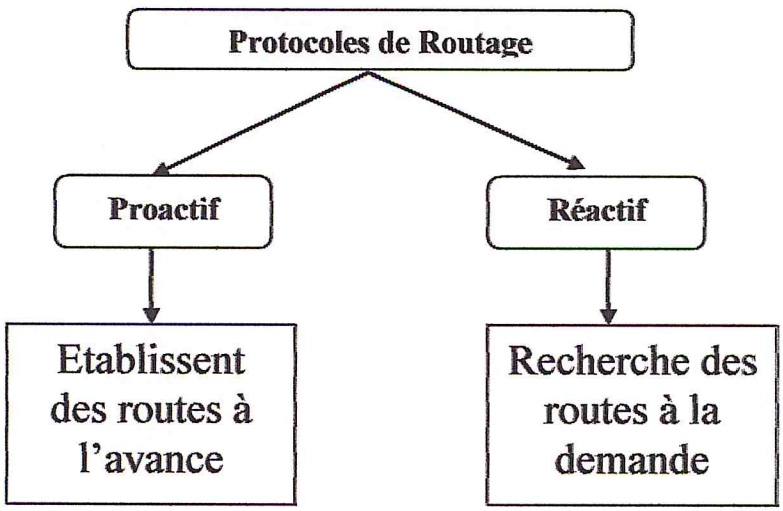


Figure I.7 : Classification des protocoles de routage

Les protocoles Proactif :

Les protocoles de routage proactifs se basent sur la même philosophie des protocoles de routage utilisées dans les réseaux filaires conventionnels. Ses deux principales méthodes sont :

- Etat de lien (Link State).
- Vecteur des Distance (Distance Vector).

Les deux méthodes exigent une mise à jour périodique des données de routage qui doivent être diffusées par les différents nœuds du réseau.

On peut citer quelques exemples de protocoles Proactifs:

- DSDV (Dynamic Destination Sequenced Distance Vector Routing Protocol).
- OLSR (Optimized Link State Routing Protocol).
- FSR (Fisheye State Routing Protovol).

Le problème avec ces protocoles est la surcharge. Les nœuds diffusent et maintiennent les informations de routage sans prendre en considération les besoins du réseau.

Les protocoles Réactif :

Contrairement aux protocoles proactifs, les protocoles réactifs créent la route au besoin. Ainsi la route est établie et maintenue par une procédure de maintenance des routes jusqu'à ce que le destinataire devienne inaccessible, ou que la route ne soit plus utilisée depuis longtemps.

Voici quelques exemples de protocoles réactifs :

- DSR (Dynamic Source Routing Protocol).
- AODV (Ad hoc On Demande Distante Routing Protocol).
- TORA (Temporally-Ordered Routing Algorithm).

Aucune mise à jours périodique n'est nécessaire pour ces protocoles. Les routes sont crées et maintenues seulement au besoin.

Les protocoles Hybrides :

Ces protocoles essaient d'incorporer les aspects des protocoles proactifs et réactifs. Ils sont généralement utilisés pour gérer le routage hiérarchique. Dans ce

mode de routage les nœuds sont regroupés dans des clusters, chacun contenant un nœud tête (head). Le Head du cluster agit comme une passerelle pour les autres clusters, comme l'illustre la Figure I.8 :

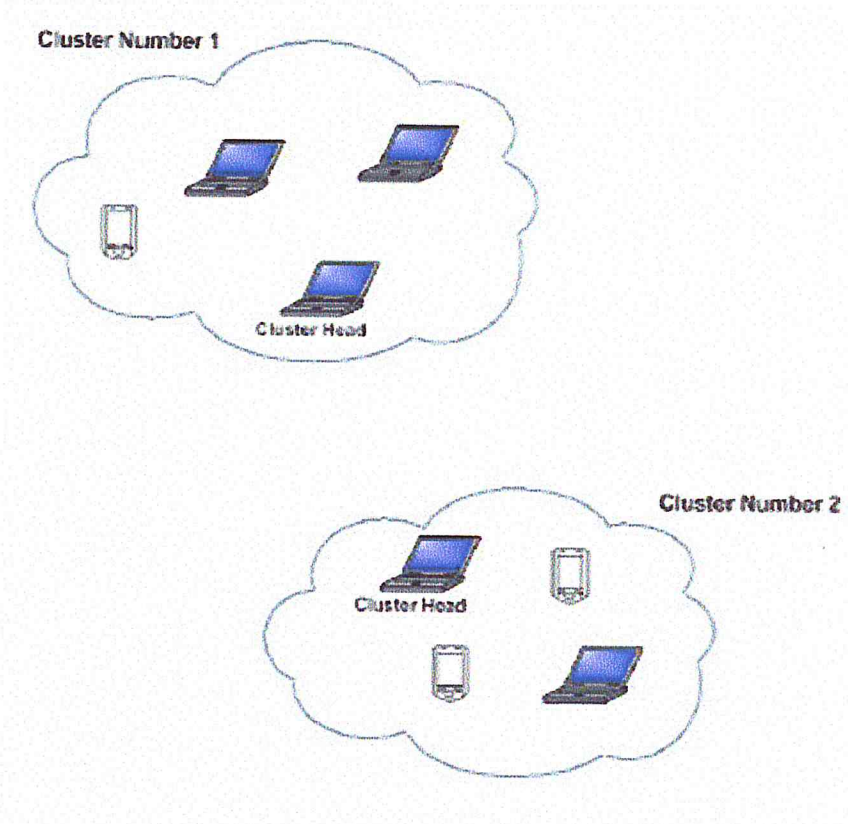


Figure I.8 [Pau03]: La structure hiérarchique

L'avantage de la structure hiérarchique est la possibilité d'utiliser un protocole réactif dans le cluster ce qui est très efficace dans les réseaux à faible portée. Pour la communication inter-clusters, un protocole proactif peut être implémenté. Il assure une meilleur porté au réseau.

ZRP (Zone Routing Protocol) est un exemple de protocole de routage Hybride. Pour plus de détails sur les protocoles de routage voir référence [Ris03]

I.11 Le problème de la couche MAC

Le médium sans fil se distingue du médium câblé par le fait qu'il est un medium partager et largement accessible, et par conséquent les transmissions peuvent interférer entre elles.

Considérons la situation dans la figure suivante:

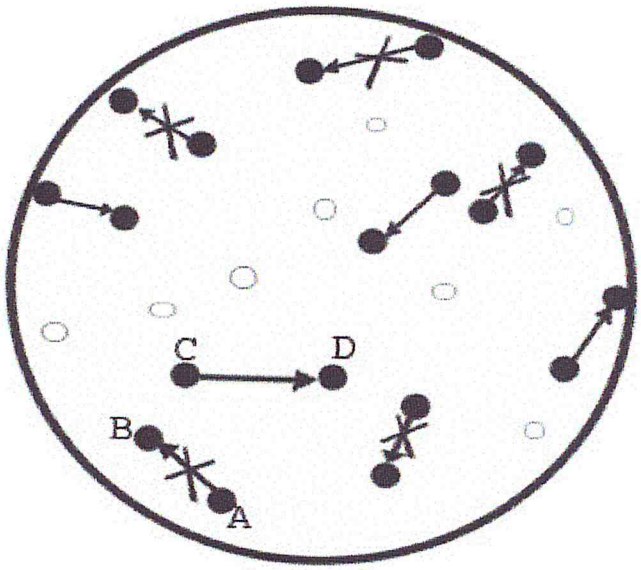


Figure I.9 [Kaw01]: Le problème d'accès au canal

Le nœud A ne peut transmettre à B en même temps que le nœud C transmet à D si B se trouve dans le domaine de puissance (porté) de C. Ainsi, seulement un certain ensemble de transmission simultané peuvent être effectué dans le réseau sans fil, comme il est indiqué dans la figure ci-dessus.

Le problème de control d'accès au canal consiste à assurer la transmission directe de manière que les paquets atteignent leurs destinations voisines prévues (d'un seul saut) sans interférence.

Le challenge principal dans la conception d'un protocole MAC pour un réseau ad hoc, est de réduire l'impact du problème des nœuds cachés, quand le nœud source ne peut pas entendre la transmission d'une autre source distante[4], et d'assurer l'équité (fairness) entre les nœuds partagent les canaux.

[4] Ce problème va être exposé dans le chapitre suivant.

I.12 Conclusion

Avant de conclure ce chapitre, nous jugeons nécessaire de signaler que les fondements des réseaux ad hoc ne constituent nullement un nouveau concept. En tant que réseaux dynamiques sans fil, ils ont été déployés à priori dans le domaine militaire et civil à partir des années 70 (paquet radio network). Néanmoins, la particularités de ces réseaux est la libération de toute forme d'infrastructure (infrastructureless) et par conséquence l'utilisations des hotes mobiles comme relais et alors des chemins multi-sauts. Depuis, plusieurs travaux de recherches leur ont été consacrés.

Un nouveau groupe de travail a été formé au sein de l'IETF (Internet Engineering Task Force), dont le but est d'étudier et de développer les supports de routage standard de l'Internet pour les réseaux mobiles et les segments IP sans fil autonomes, et de développer des structures pour implanter les protocoles basés sur IP dans les réseaux ad hoc. De plus, plusieurs conférences internationaux et séminaires organisés notamment par IEEE et ACM ont eu pour objet d'étude ces réseaux. MobiHoc (The ACM Symposium on Mobile Ad Hoc Networking & Computing) constitue l'une des plus importantes conférences d'ACM SIGMOBILE (Special Interesr Group on Mobility of systems) dans le domaine.

Notons que les recherches dans ce type de réseaux ne cessent d'attirer l'attention de divers secteurs, académique, industriel et gouvernemental notamment.

Néanmoins, de par les complexités suscitées des réseaux ad hoc (la Mobilité de tout les sites du réseau, la topologie dynamique, le routage Multi-saut, le problème d'accès au canal, ...), plusieurs problèmes demeurant encore ouverts à la recherche.

Notre premier chapitre ayant traité de généralités (définition des réseaux ad hoc, leurs caractéristiques, leurs applications,...), les chapitres suivant seront consacré justement a l'un de ces problème, à savoir le problème de la couche MAC. Dans le chapitre suivant, nous présenterons en détail les différents protocoles d'accès au canal.

CHAPITRE II : LES PROTOCOLES DE CONTROLE D'ACCES AU CANAL

II.1 Introduction

Les réseaux ad hoc imposent des exigences spécifiques dans la conception des protocoles de communication pour les différentes couches du réseau. Ces protocoles doivent satisfaire un certain nombre de critères, à savoir :

- Assurer un haut débit et un faible temps de latence moyen[5].
- Supporter le trafics hétérogènes (données, voix, et vidéo).
- préserver de l'ordre des paquets.
- Supporter la priorité du trafic.
- Assurer l'équité (fairness) entre les nœuds.

La conception d'un protocole de communication qui satisfait ces critères reste plus difficile à atteindre dans un réseau ad hoc, à cause de la mobilité et de l'absence de toute sorte d'infrastructure fixe. Le contrôle et l'administration du réseau doivent être distribués entre les différentes stations, ainsi une conception rigoureuse des protocoles de contrôle d'accès au canal doit être envisagée.

Un protocole d'accès au canal (MAC) est un ensemble de règles et de procédures de rôles multiples, il est en charge d'éviter les collisions, d'assurer le partages de la bande passante, et de résoudre certains problèmes spécifiques aux transmissions hertziennes, à savoir le problème des terminaux cachés, et des terminaux exposés que nous étudierons par la suite.

La conception d'un bon protocole MAC présente comme challenges la mobilité ainsi que l'instabilité du canal de transmission qui vari avec le temps. La mobilité affecte le protocole MAC dans la mesure où, quand l'ensemble des utilisateurs se concurrence pour accéder au canal, la capacité dans le medium continue à changer, ce qui rend difficile l'allocation équitable des canaux. Ceci d'une part, d'autre part la caractéristique de la variation des conditions des canaux avec le temps (timevarying

[5] Le temps de latence moyen définit le temps moyen que prend le paquet pour atteindre sa destination.

chanals), tels que le fading[6] et les interférences perturbent l'allocation du canal rendent plus difficile le contrôle d'accès au canal.

Plusieurs protocoles MAC on été proposés et conçus. Dans ce qui suit, nous présenteront les protocoles d'accès au canal désignés aux réseaux sans fil les plus connus. Nous décrirons pour chacun les principales caractéristiques et fonctionnalités.

II.2 Classification des protocoles MAC

Les protocoles d'accès au canal des réseaux sans fils peuvent être classés de différentes façons. Selon le mode d'échange nous aurons des protocoles synchrone et d'autre asynchrones. Par ailleurs, si nous considérons l'initiateur de la communication, nous distinguons des protocoles où la communication débute ou par l'émetteur ou par le récepteur. La dernière classification prend en considération s'il y a compétition entre les stations pour accéder au canal, elle distinct donc des protocoles basés sur la compétition (contention-based) et d'autre sans compétition.

II.2.1 Classification selon le mode d'échange

Il s'agit là, de distinguer les protocoles MAC selon le mode d'échange.

II.2.1.1 Les protocoles MAC Synchrones

Dans les protocoles MAC synchrones, tous les nœuds sont synchronisés sur la même horloge. Ceci se réalise par le biais d'une diffusion régulière d'une trame balise (Beacon Frame) [7]par une entité maître (master). Tous les noeuds écoutent ce beacon et synchronisent leurs horloges avec celle du maître [Toh02].

II.2.1.2 Les protocoles MAC asynchrones

Dans cette classe de protocoles, les nœuds ne sont pas nécessairement synchronisés sur la même horloge. Un mécanisme de contrôle distribué est donc utilisé pour coordonner l'accès au canal. De ce fait, l'accès au canal est basé sur la compétition [Toh02].

II.2.2 Classification selon l'initiateur de la communication

Nous distinguons dans cette classification deux types de protocoles. Des protocoles où la communication est initiée par l'émetteur, et d'autres où elle l'est par

[6] Fading : c'est l'affaiblissement du signal transmis.
[7] La trame balise est une trame envoyée périodiquement par l'entité maître contenant les informations de synchronisation.

le récepteur. Le choix de la stratégie d'initiation est dépendant des types d'applications que le réseau doit supporter.

II.2.2.1 Protocoles MAC avec initiation au récepteur (Receiver-Initiated)

Comme illustré dans la Figure II.1, c'est le récepteur (nœud B) qui contacte l'émetteur (nœud A) en premier, pour l'informer qu'il est prêt à recevoir des données. C'est une forme de polling.

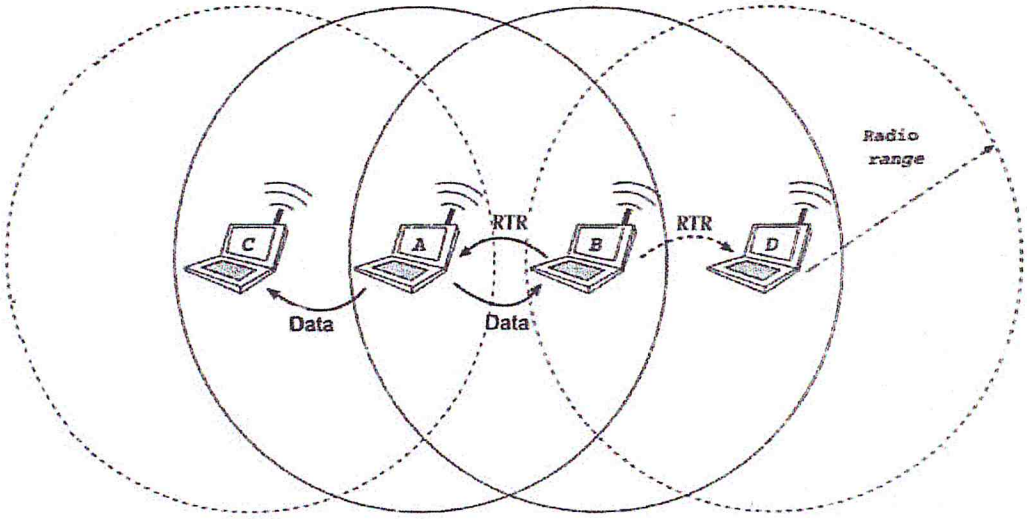


Figure II.1 [Toh02] : Receiver-Initiated MAC protocols

C'est aussi une forme passive d'initiation, puisque l'émetteur n'a pas a initier la demande d'envoie. De plus, comparé au mécanisme RTS/CTS[8], un seul paquet de contrôle est utilisé. Le premier protocole conçu dans cette classe est le protocole MACA-BI (Multiple Access Collision Avoidance – By Invitation) que nous étudierons plus tard dans ce chapitre.

II.2.2.2 Protocoles MAC avec initiation a l'émetteur (Sender-Initiated)

Dans cette classe de protocoles, c'est à l'émetteur d'initier la communication en informant le destinataire qu'il a des données à transmettre. Un exemple qui rentre dans cette classe est le protocole MACA que nous étudierons par la suite.

[8] Ce mécanisme sera expliqué par la suite.

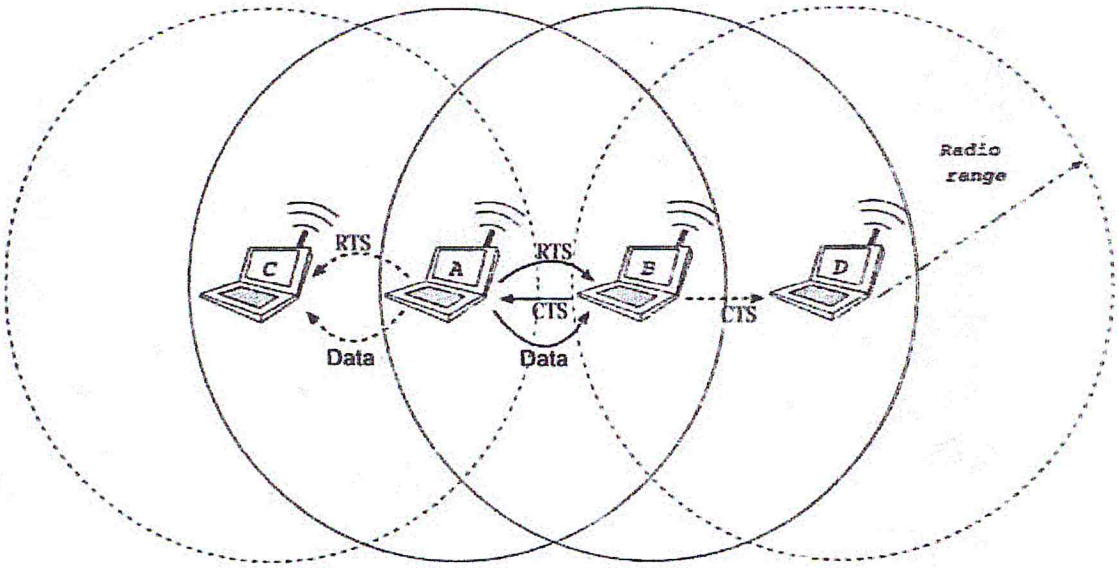


Figure II.2 [Toh02] : Sender-Initiated MAC protocols

Comme le montre la figure II-2, le nœud A envoie une demande de transmission RTS au noeud B (le récepteur). Si B est prêt à recevoir les données de A il répond par un CTS, ainsi A peut procéder à transmettre ses données. Cette approche est nommée aussi approche orientée émetteur, elle est plus intuitive et plus adaptée pour un réseau de trafics imprédictible.

II.2.3 Classification selon la compétition

En plus des deux premières classifications, les protocoles MAC peuvent être distingués aussi par l'existence ou pas de la compétition entre les stations qui veulent accéder au medium.

II.2.3.1 Les protocoles basé sur la compétition

Dans ce type de protocoles, les stations doivent se concurrencer pour avoir le contrôle du canal de transmission chaque fois qu'elles veulent transmettre un paquet. Développé à l'université de Hawaii en 1970, le protocole ALOHA [Abr70] en est un exemple. Dans ce protocole, une entité désirant émettre des données le fait immédiatement. Il y aura bien entendu des collisions, mais en écoutant ce qu'il envoie, l'émetteur pourra savoir si le paquet a été détruit par une autre émission auquel cas il attendra un temps aléatoire pour éviter que deux paquets en collision le reste indéfiniment, ensuite il émettra à nouveau ses données. Quand le réseau est chargé, les chances de collision augmente et donc les performances d'ALOHA se réduisent. Le protocole CSMA (Carrier Sens Multiple Access) [Kle75] essaie de

remédier à ce problème. Dans CSMA quand un nœud veut émettre, il ne transmet pas directement mais il écoute le canal avant de transmettre. S'il est libre, il émet ses données, s'il est occupé, deux cas sont possibles. Soit le nœud écoute le canal et transmet dès qu'il se libère[9], soit il attend un temps aléatoire avant de recommencer le processus depuis le début[10].

C'est les protocoles adoptés pour les réseaux ad hoc.

II.2.3.2 Les protocoles non-basés sur la compétition

Etant donné l'instabilité des protocoles basés sur la compétition dans un réseau chargé, plusieurs protocoles non- basés sur la compétition on été proposés, cela inclut :

- Les protocoles d'assignation fixe : ils incluent les méthodes de multiplexage fréquentielle FDMA (Frequency Division Multiple Access), de multiplexage temporelle TDMA (Time Division Multiple Access), et du multiplexage par code CDMA.

Dans un réseau basé sur le protocole TDMA, le domaine de temps est divisé en plusieurs intervalles dénommés *slot time* avec un intervalle pour chaque utilisateur. Par contre FDMA consiste à diviser la bande de fréquence en plusieurs divisions, avec une division pour chaque utilisateur. Ces deux approches conduisent à une allocation rigide de la bande de fréquence ou de temps, c'est-à-dire que chaque utilisateur aura une fréquence ou un slot time réservé, même s'il ne transmet pas continuellement, ce qui cause un gaspillage de la bande de fréquence. Une autre limite de ces approches est la difficulté de gérer l'assignation des slot time ou des fréquences dans un environnement dépourvu d'un contrôle centrale. Le protocole CDMA, ou accès multiple par division de codes, autorise l'allocation de la totalité de la bande de fréquences, de manière simultanée, à tous les utilisateurs d'une même cellule. Pour ce faire, un code binaire spécifique est octroyé à chaque utilisateur. L'utilisateur se sert de son code pour transmettre l'information qu'il désire communiquer en format binaire d'une manière orthogonal, c'est-à-dire sans interférence entre les signaux des

[9] Dans ce cas, le protocole est nommé CSMA persistant (persistent CSMA).
[10] Il est nommé dans ce cas CSMA non-persistant (Non-persistent CSMA).

autres communications. Cependant, CDMA est difficile à implémenter dans un environnement ad hoc, à cause de la mobilité des noeuds et la séquence requise pour garder trace des modèles de saut de fréquence et/ou la gamme des codes pour les noeuds dans un voisinage qui varie avec le temps

- Le polling et le passage de jeton (token passing) : le polling est une technique d'interrogation-réponse dans laquelle un maître interroge régulièrement les esclaves pour leur donner le moyen de communiquer. Le Polling réduit le temps perdu au silence des stations dans TDMA, mais il introduit un délai supplémentaire dans la scrutation de toutes les stations. Le problème est le même dans le cas des protocoles a passage de jeton. Le jeton doit passer par toutes les stations de l'anneau virtuel, y compris les stations qui n'ont pas de données à transmettre.

- Les protocoles avec réservation dynamique : plusieurs protocoles basés sur la réservation dynamique ont été proposés pour minimiser les problèmes constatés dans les protocoles d'assignation fixe, du polling et du token passing. Dans cette catégorie de protocoles, les stations doivent se concurrencer pour réserver soit le droit de transmettre un seul paquet, soit le droit de devenir membre de l'ensemble des utilisateurs permit d'utiliser le canal.

II.3 Les problèmes d'accès au canal dans les réseaux ad hoc

Dans cette partie, nous allons étudier quelques problèmes spécifiques à la transmission sans fil, à savoir le problème du terminal caché, et du terminal exposé.

II.3.1 Le problème des Terminaux Cachés

C'est un problème rencontré dans les protocoles basés sur la compétition. Il se définit comme suit : Si l'émetteur écoute le canal et le trouve libre, cela ne veut pas dire qu'il est libre dans la zone du récepteur, la Figure II.3 illustre le phénomène. Le terminal A écoute le canal et le trouve libre. Il émet son message au terminal B et tout au long du transfert, il ne détecte pas de collision. Pourtant, pendant ce temps le

terminal C a émis un message vers B qui reçoit donc deux paquets en même temps et ne peut alors en traiter aucun[11].

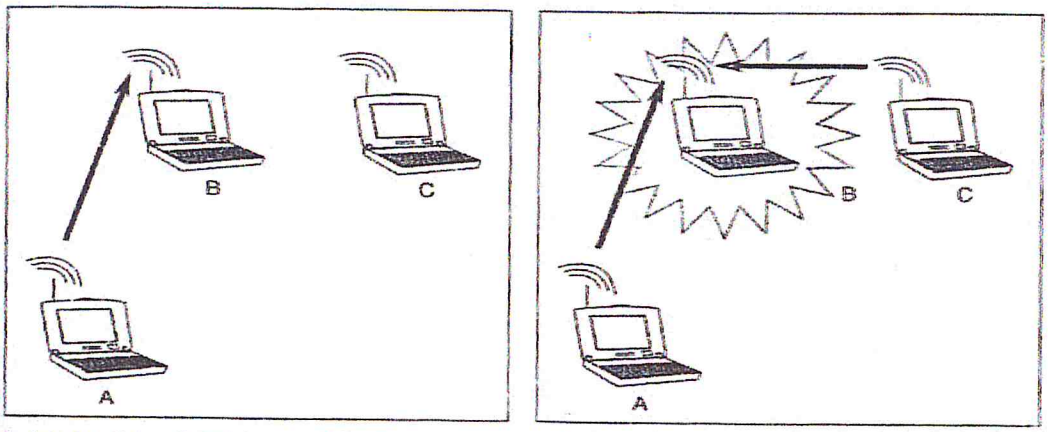


Figure II.3 [Toh02] : Problème de noeud caché

Dans ce cas, le noeud C est caché par rapport à A, donc C ne peut pas écouter la transmission de A à B, de ce fait rien ne l'empêche de transmettre à B, ce qui génère une collision au niveau du nœud B.

Pour éviter la collision, tout les nœuds voisins au nœud récepteur doivent être informés que le medium est occupé. Pour cela, un mécanisme de réservation du canal est utilisé. Ce mécanisme manipule des paquets de contrôles (de taille réduite), dont on distingue deux types, le RTS (Request To Send) et le CTS (Clear To Send). Quand un noeud veut transmettre, il émet un RTS au destinataire, le nœud destinataire répond à l'émetteur par un CTS pour permettre la transmission. Etant donnée la nature de diffusion du medium sans fil, tous les nœuds voisins au nœud émetteur et au nœud récepteur seront informés par RTS (ou CTS) que le canal est occupé. De ce fait, ces noeuds stoppent leur transmission pendant une durée suffisante pour ne pas perturber la communication entre les deux noeuds communicants, ce qui permet d'éviter les collisions au niveau de l'émetteur et au niveau du récepteur. La Figure II.4 illustre le mécanisme RTC/CTS :

[11] Cela suppose que deux émissions simultanées détruisent les deux paquets, ce qui est souvent le cas. Avec certaines technologies, il est possible avec une certaine probabilité que le paquet émis avec la plus forte puissance soit exploitable.

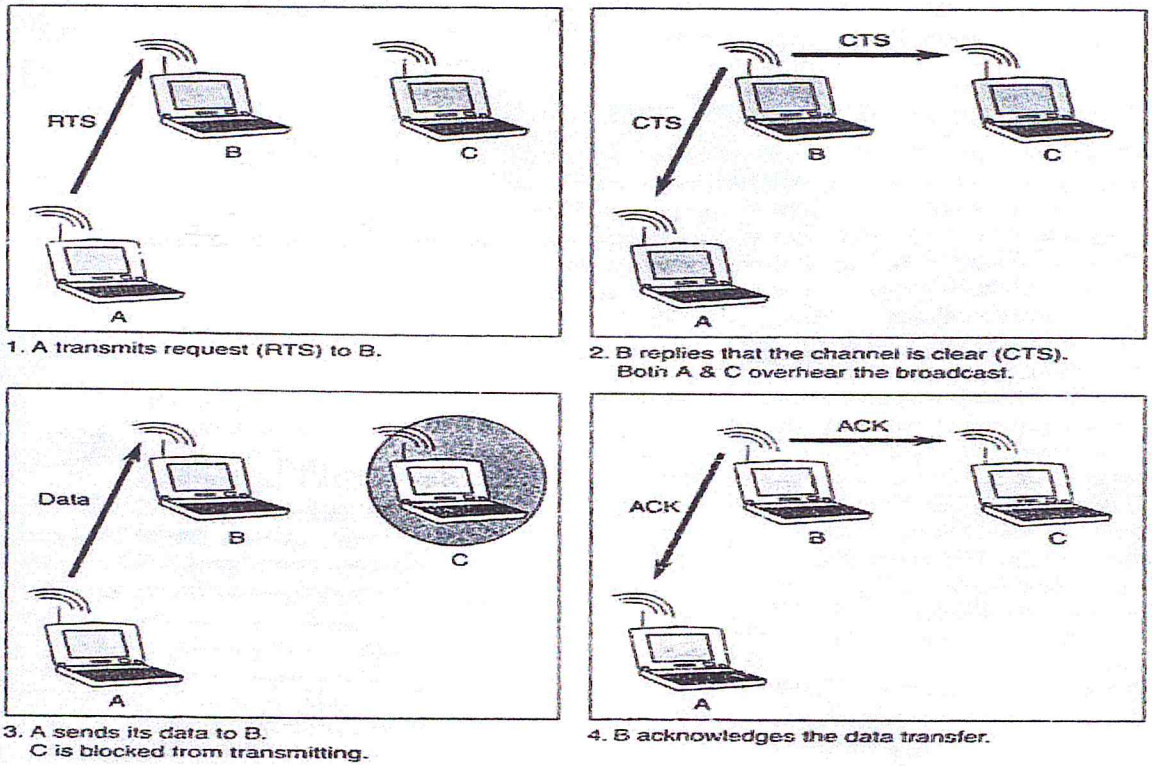


Figure II.4 [Toh02] : Mécanisme RTS/CTS pour résoudre le problème de nœud caché

L'utilisation d'un accusé de réception (ACK) est indispensable, car l'écoute du canal par l'émetteur ne suffit pas à dire si les données ont été transmises correctement.

II.3.2 Les points faibles du mécanisme RTS/CTS

Le mécanisme RTS/CTS n'est pas une solution complète au problème de nœuds cachés. Nous présenterons ici deux scénarios montrant les faiblesses de ce mécanisme [Toh02].

Le premier scénario est illustré par la Figure II.5, le nœud B répond par un CTS au RTS envoyé par A. Pendant ce temps, D transmet un RTS à C. Cela produit une collision au niveau du nœud C. Puisque D n'a pas reçu le CTS de C, il retransmet donc son RTS. Le nœud A, après avoir reçu le CTS de B, et sans aucune conscience de la collision produite au niveau du nœud C, procède à la transmission de données à B. Une collision est produite alors entre les données envoyées par A et le CTS envoyé de C à D.

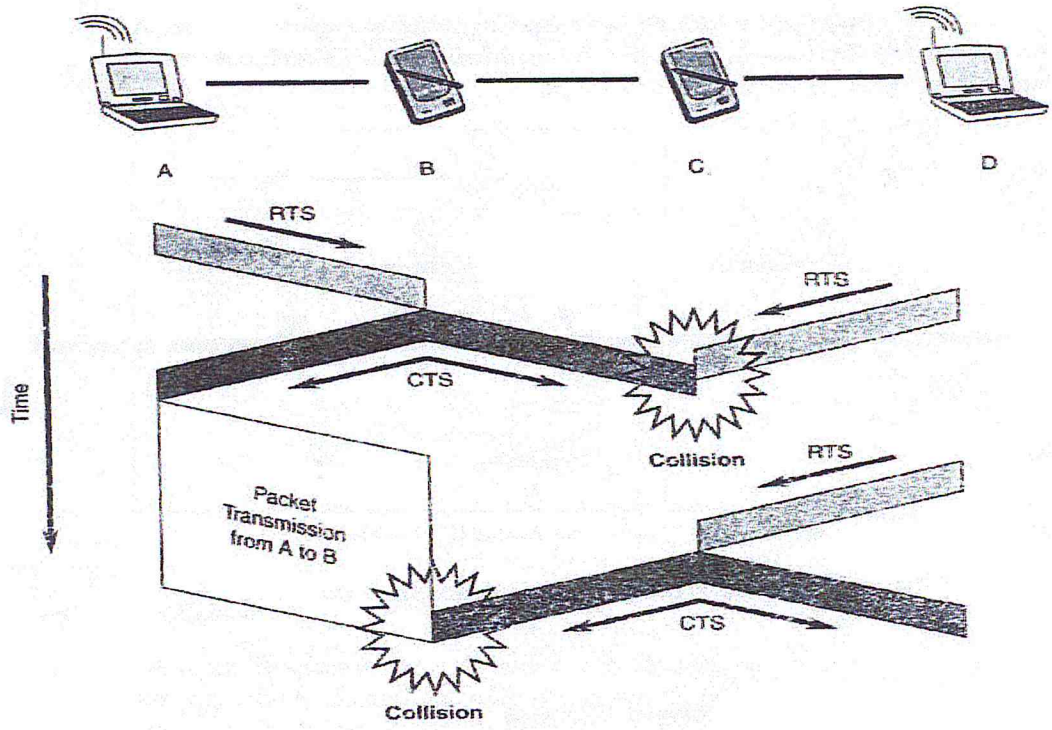


Figure II.5 [Toh02] : L'incomplétude du mécanisme RTS/CTS

Le deuxième scénario parait lorsque plusieurs CTS sont envoyés aux nœuds voisins différents, ce qui produit des collisions. Comme illustré dans la figure II.6, deux nœuds ont envoyé des paquets RTS à des stations différentes dans des moments différents. Le nœud A a envoyé un RTS à B, au moment où B retourne le CTS à A, C désire transmettre à D, il lui envoie donc un RTS. Le nœud C ne peut pas entendre le CTS envoyé par B (puisque il est entrain de transmettre son RTS à D). Donc il est inconscient de la communication entre A et B. Le nœud D retourne un CTS à C. De ce fait, les deux nœuds A et C rentrent en collision quand ils commencent de transmettre leurs données.

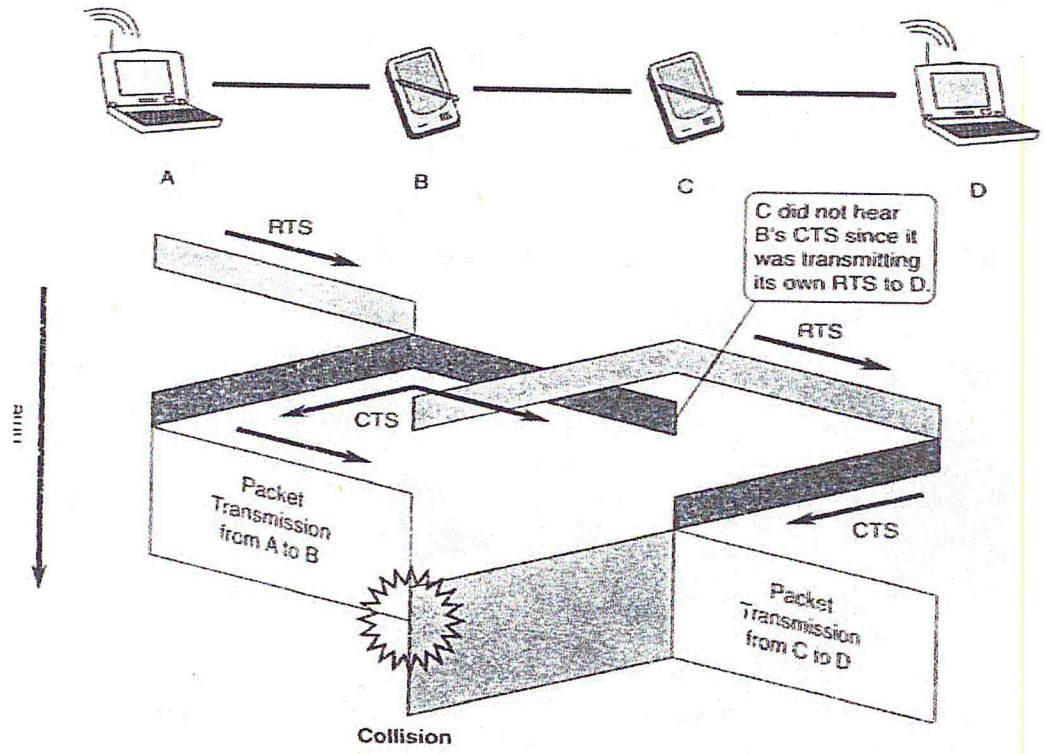


Figure II.6 [Toh02] : Un autre problème du mécanisme RTS/CTS

II.3.3 Problème des terminaux exposés

On peut également considérer le problème inverse du noeud caché dit du « noeud exposé ». Celui-ci est illustré par la Figure II.7. Le noeud B veut envoyer des données au noeud A. Il écoute le canal et détecte que le noeud C émet un message pour D. Il décide alors de reporter son émission alors que cette communication pouvait quand même avoir lieu, ce qui cause un gaspillage de la bande passante.
Un noeud exposé (B dans ce cas) est donc un noeud qui se trouve dans la portée de communication de l'émetteur (le nœud C) et hors de la portée du destinataire (D).

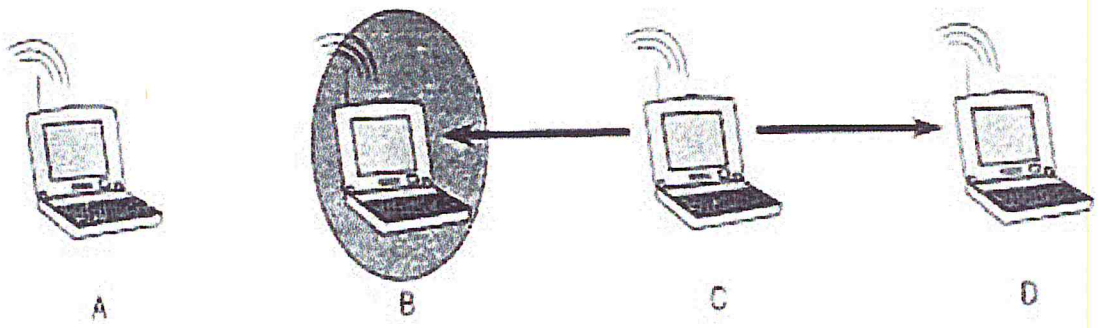
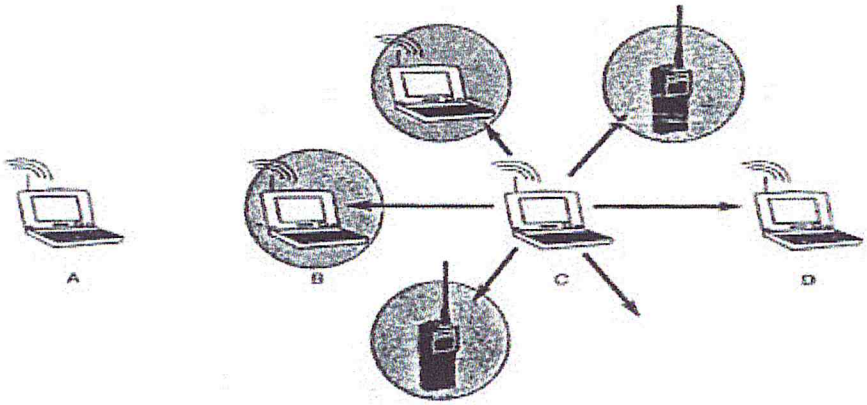
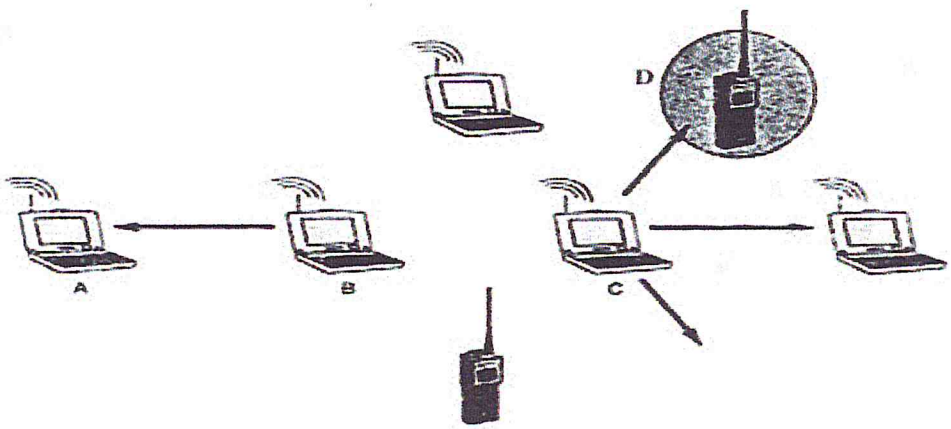


Figure II.7 [Toh02] : Problème de nœud exposé

La solution de ce problème consiste à séparer les canaux de contrôle et les canaux de données. Ou à utiliser des antennes dirigées [Toh02], comme c'est illustré sur la FigureII.8. Dans ce second cas, le terminal C qui utilise une antenne Omnidirectionnelle peut rendre plusieurs terminaux exposés (Figure II.8.a), et donc il les empêche de communiquer, même si ces communications peuvent être effectuées sans causer d'interférences avec la transmission de C. Ce problème est atténué, si le nœud C utilise une antenne dirigée, comme le montre la figure II.8.b. Le nœud C peut continuer de communiquer avec D sans influencé la communication entre A et B.



a- Une antenne omnidirectionnelle est utilisé – tout les voisins sont exposé



b : Une antenne dirigé remédier a ce problème- B n'est pas bloqué

Figure II.8 [Toh02] : Utilisation d'antenne dirigée pour résoudre le problème des nœuds exposés.

II.4 Les protocoles MAC

Dans cette partie, nous présenterons les différents protocoles MAC, le protocole CSMA ayant été abordé précédemment dans ce chapitre (cf. Les protocoles basés sur la compétition.), nous commençons alors par le protocole IEEE 802.11.

II.4.1 Le protocole IEEE 802.11 [Puj01]

En 2001, l'IEEE 802.11 est devenue le premier standard international pour les réseaux locaux sans fil, il a pour but de fournir une connectivité sans fil à des stations fixes ou mobiles qui demandent un déploiement rapide au sein d'une zone local en utilisant différentes bandes de fréquences.

Comme tous les standards de l'IEEE, l'IEEE 802.11 couvre les deux premières couches du modèle de référence OSI (Open Systems Interconnexion). L'une des caractéristiques essentielles du standard est qu'il définit une couche MAC commune à toutes les couches physique. Ainsi, de futures couches physiques peuvent être ajoutées au standard sans qu'il soit nécessaire de changer la couche MAC.

Comme illustre la figure II.9 :

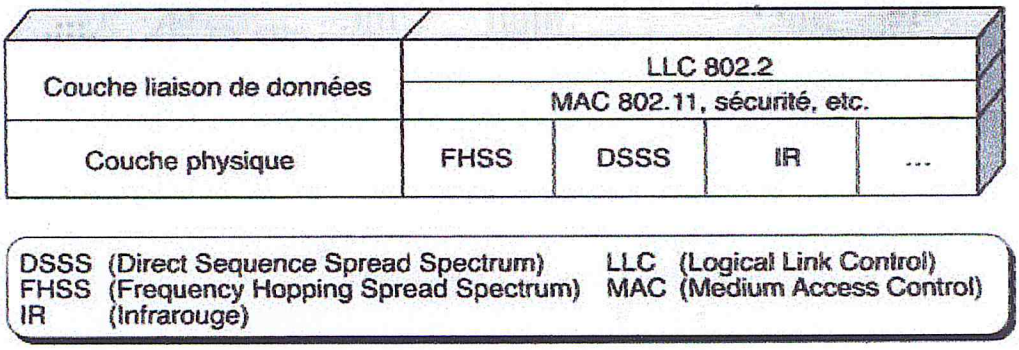


Figure II.9 : Le modèle IEEE 802.11

La couche physique a pour rôle de transporter correctement la suite de signaux 0 ou 1 que l'émetteur souhaite envoyer au récepteur. Elle est divisée en deux sous couches, PLCP (Physical Layer Convergence Protocol) et PMD (Physical Medium Dependent). La sous-couche PMD s'occupe de l'encodage des données. De son coté, la sous couche PLCP s'occupe de l'écoute du support et fournit un CCA (Clear Channel Assessment), qui est le signal utilisé par la couche MAC pour savoir si le support est occupé ou non.

La couche liaison du protocole IEEE 802.11 est composé essentiellement de sous couches, LCC (Logical Link Control) et MAC. Cette couche LCC utilise les

mêmes propriétés que la couche LCC 802.2. Il est de ce fait possible de relier un WLAN à tout autre réseau local appartenant à un standard de l'IEEE. La couche MAC, quand à elle, est spécifique de l'IEEE 802.11.

Le rôle de la couche MAC 802.11 est assez similaire à celui de la couche MAC 802.3 du réseau Ethernet : les terminaux écoutent le canal avant d'émettre. Si le canal est libre le terminal émet, sinon il se met en attente. Cependant, la couche MAC 802.11 intègre un grand nombre de fonctionnalités que l'on ne trouve pas dans la version filaire.

Les fonctionnalités nécessaires pour réaliser un accès sur une interface sans fil sont les suivantes :

- Procédure d'allocation du support.
- Adressage des paquets.
- Formatage des trames.
- Contrôle d'erreurs CRC (Cyclic Redundancy Check).
- Fragmentation et réassemblage.

L'une des particularités du standard est qu'il définit deux méthodes d'accès fondamentalement différentes au niveau de la couche MAC :

- La première DCF (Distributed Coordination Function), qui correspond à une méthode d'accès assez similaire au réseau traditionnel supportant le best effort[12]. La DCF a été conçu pour prendre en charge le transport de données asynchrone, dans lequel tous les utilisateurs qui veulent transmettre des données ont une chance égale d'accéder au support.
- La seconde méthode d'accès est la PCF (Point Coordination Function). Fondée sur l'interrogation à tour de rôle des terminaux, ou polling, contrôlée par le point d'accès, la méthode PCF est conçu essentiellement pour la transmission des données sensibles, qui demandent une gestion du délai utilisée pour les application du type temps réel telles que la voix ou la vidéo, mais elle est toutefois centralisée.

[12] Le best-effort est un mode de qualité de service d'un réseau n'offrant pas de garantie de débit ni de délai.

Un réseau ad hoc utilise uniquement le DCF. Tandis qu'un réseau classique IEEE 802.11, avec des points d'accès, utilise à la fois le DCF et le PCF comme méthode d'accès.

Pour notre travail nous nous intéresserons au 802.11 DCF, puisque la méthode d'accès PCF n'est pas adaptable pour un réseau ad hoc.

IEEE 802.11 DCF

La DCF est la méthode d'accès générale utilisée pour permettre des transferts de données asynchrones en best effort. Elle est basé sur le protocole CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance) plus un mécanisme d'acquittement positif.

Le protocole CSMA/CA

L'EEE 802.11 utilise un protocole légèrement modifié par rapport au CSMA/CD[13] d'Ethernet, appelé CSMA/CA. Le CSMA/CA évite les collisions en utilisant des trames d'acquittement, ACK (ACKnowledgement), un ACK est envoyé par la station destinataire pour confirmer que les données sont correctement reçues.

L'accès au support est contrôlé par l'utilisation d'espace intertrame, ou IFS (Inter-Frame Spacing), qui correspond à l'intervalle de temps entre la transmission de deux trames. Les intervalles IFS sont des périodes d'inactivité sur le support de transmission. Les valeurs des différents IFS sont calculées par la couche physique. Le standard définit trois types d'IFS

- SIFS (Short IFS). Le plus petit des IFS. utilisé pour séparer les transmissions au sein d'un même dialogue (envoi de données, ACK. etc.). Il y a toujours une seule station pour transmettre â cet instant, ayant donc la priorité sur toutes les autres stations. La valeur du SIFS est de $28\,\mu s$.
- PIFS (PCF IFS). Utilisé par le point d'accès pour accéder avec priorité sur le support par rapport aux stations du réseau. le PIFS correspond à la valeur du

[13] CSMA/CD (Carrier Sense Multiple Access/ Collision **Détection**) n'est pas adaptable pour un environnement sans fil, du fait que les liaisons radio ne sont pas en full-duplex, les stations ne sont pas capable d'écouter le support et de transmettre en même temps. La détection de collisions est donc impossible.

SIFS, auquel on ajoute un temps, ou timeslot, défini dans l'algorithme de backoff, la valeur est 78 μs.

- DIFS (DCF IFS). Utilisé lorsqu'une station veut commencer une nouvelle transmission. Le DIFS correspond à la valeur du PIFS, à laquelle on ajoute un temps de 128 μs.

Un terminal peut écouter l'activité de toutes les stations voisines. Ainsi, lorsqu'une station envoie une trame, les autres stations l'entendent et pour éviter une collision, mettent à jour un timer, appelé NAV (Network Allocation Vector), permettant de retarder toutes les transmissions prévues. Le NAV est calculé en utilisant l'information située dans le champ durée de vie, ou TTL (Time To Live), contenu dans les trames qui ont été envoyées. Les autres stations n'ont la possibilité de transmettre qu'après la fin du NAV.

Lors d'un dialogue entre deux stations, le NAV est calculé par rapport au champ durée de vie des différentes trames qui sont envoyées (données. ACK, etc.). Le NAV correspond à un temporisateur qui détermine l'instant auquel la trame peut être transmise avec succès. Une station source voulant transmettre des données écoute le support (en parlera de l'écoute du support par la suite). Si aucune activité n'est détectée pendant une période de temps correspondante à un DIFS, la station source transmet ses données immédiatement. Si le support est encore occupé, la station continue de l'écouter jusqu'à ce qu'il soit libre. Quand le support devient disponible, la station retarde encore sa transmission en utilisant l'algorithme de backoff avant de transmettre ces données.

Si les données envoyées ont été correctement reçues, la station destinataire attend pendant un temps équivalent à un SIFS et émet un ACK pour confirmer la bonne réception des données. Si l'ACK n'est pas détecté par la station source ou si les données ne sont pas reçues correctement ou encore si l'ACK n'est pas reçu correctement, on suppose qu'une collision s'est produite et la trame est retransmise. Comme illustré à la figure II.10.

Lorsque la station source transmet ses données, les autres stations mettent à jour leur NAV, en incluant le temps nécessaire pour la transmission de la trame de données, ainsi que le SIFS et l'ACK.

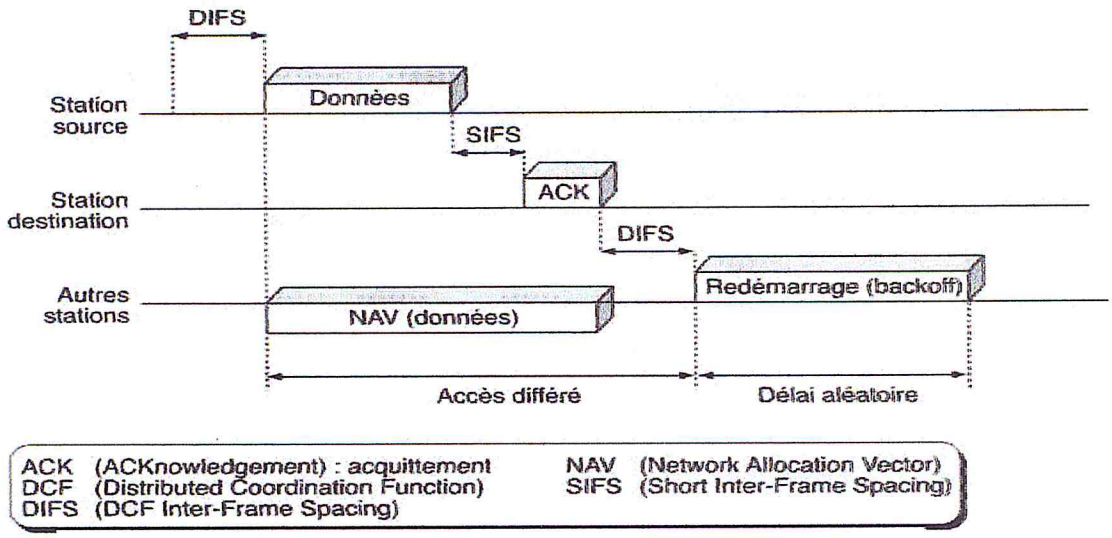


Figure II-10 : La transmission de données dans 802.11

L'algorithme de Backoff permet de résoudre le problème de l'accès au support lorsque plusieurs stations veulent transmettre des données en même temps. Dans l'IEEE 802.11, le temps est découpé en tranche, qui correspondent chacune à un timeslot. Contrairement au timeslot utilisé dans l'aloha, qui correspond à la durée minimale de transmission d'une trame, le timeslot utilisé en 802.11 est un peu plus petit que cette durée minimal. Il est utilisé pour définir les intervalles IFS ainsi que les temporisateurs pour les différentes stations. Son implémentation est différente pour chaque couche physique.

Initialement, une station calcule la valeur aléatoire d'un temporisateur, appelé timer backoff compris entre 0 et 7 et correspondant à un certain nombre de timeslots. Lorsque le support est libre, les stations décrémentent leur temporisateur jusqu'à ce que le support soit occupé ou que le temporisateur atteigne la valeur 0. Si le temporisateur n'a pas atteint la valeur 0 et que le support soit de nouveau occupé, la station bloque le temporisateur. Dès que le temporisateur atteint la valeur 0, la station transmet sa trame. Si deux ou plusieurs stations atteignent la valeur 0 au même instant, une collision se produit, et chaque station doit régénérer un nouveau temporisateur, compris cette fois entre 0 et 15.

Pour chaque tentative de retransmission, le temporisateur croît de la façon suivante :

$$[2^{2+i} x ranf()] x timeslot$$

i correspondant au nombre de tentatives consécutives d'une station pour l'envoi d'une trame, et ranf () est une variable aléatoire uniforme comprise entre 0 et 1.

Grâce à cet algorithme, les stations ont la même probabilité d'accéder au support, car chaque station doit, après chaque transmission, réutiliser le même algorithme. Son seul inconvénient est quelle ne garantie pas un délai minimal, ce qui complique la prise en charge d'applications temps réel telles que la voix ou la vidéo.

L'écoute du support

Dans l'IEEE 802.11, l'écoute du support se fait à la fois au niveau de la couche physique avec le PCS (Physical Carrier Sense) et au niveau de la couche MAC avec le VCS (Virtual Carrier Sense). Le PCS détecte la présence d'autres stations 802.11 en analysant toutes les trames passant sur le support hertzien et en détectant l'activité sur le support grâce à la puissance relative du signal des autres stations.

Le VCS[14] est un mécanisme de réservation basé sur l'envoi de trames RTS/CTS (Request to Send/Clear te Send) entre une station source et une station destination avant tout envoi de données. Une station source qui veut transmettre des données envoie un RTS, toutes les stations voisines entendant le RTS lisent le champ de durée du RTS et mettent à jour leurs NAV. La station destination ayant reçu le RTS répond, après avoir attendu pendant un SIFS, en envoyant un CTS. Les autres stations entendant le CTS lisent le champ de durée du CTS et mettent à nouveau à jour leur NAV. Après la réception du CTS par la station source, cette dernière sera assurée que le support est stable et réservé pour sa transmission. Comme illustrer à la Figure II.11.

Cela permet à la station source de transmettre ces données ainsi que de recevoir l'ACK sans collision. Comme les trames RTS/CTS réservent le support pour la transmission d'une station, ce mécanisme est habituellement utilisé pour envoyer de

[14] En plus du PCS, IEEE 802.11 utilise l'écoute virtuelle du canal (VCS) dont le but est de réduire la probabilité que deux stations entre en collision à cause du problème du terminal caché. Cependant, aucun mécanisme n'est prévu par l'IEEE802.11 pour réduire le problème de terminal exposé.

grosses trames pour lesquelles une retransmission serait trop coûteuse en terme de bande passante.

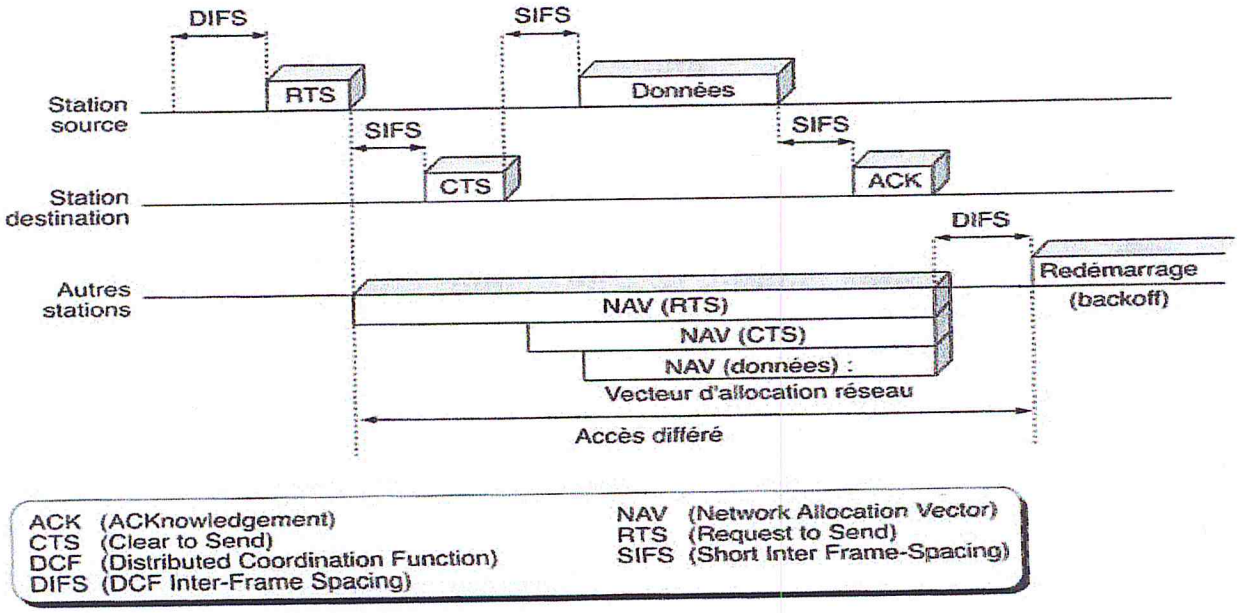


Figure II.11 : Transmission de données en utilisant les trames RTS/CTS

Les stations peuvent choisir d'utiliser toujours le mécanisme RTS/CTS, ou de ne l'utiliser que lorsque la trame à envoyer excède une variable RTS_Threshold ou encore de ne jamais l'utiliser.

Le mécanisme de réservation RTS/CTS permet de résoudre le problème du terminal caché, cependant, il ne permet pas d'éviter les collisions, car des RTS peuvent être envoyés simultanément par deux stations voisines. Toutefois, une collision de RTS ou de CTS ne gaspille pas autant de bande passante qu'une collision de données, étant donné que les trames RTS (ou CTS) sont relativement petites.

En conclusion, le CSMA/CA permet de partager l'accès. Le mécanisme d'acquittement traite en outre les problèmes liés aux interférences et, en règle générale, à tous les problèmes liés à l'environnement sans fil de manière efficace. Le mécanisme de réservation RTS/CTS évite les problèmes de station cachée. Toutefois tous ces mécanismes entraînent l'ajout au trames 802.11 des entêtes, que les trames Ethernet ne possèdent pas. C'est l'une des raisons pour lesquelles les réseaux 802.11 montrent toujours des performances plus faibles que les réseaux locaux Ethernet.

Les trames IEEE 802.11

Les paquets IP doivent être transmis sur le support hertzien, et ils doivent être placés dans des trames de données. De plus, pour contrôler et pour gérer la liaison, il est nécessaire d'avoir des trames spécifiques. Il existe trois types de trames :

- les trames de données, utilisées pour la transmission de données utilisateur ;
- les trames de contrôle, utilisées pour contrôler l'accès au support (RTS, CTS, ACK)
- les trames de gestion, utilisées pour les associations ou les désassociations d'une station avec un point d'accès, ainsi que pour la synchronisation et l'authentification.

Structure des trames

Toutes les trames IEEE 802.11 sont composées de la manière illustrée sur la figure II.12 :

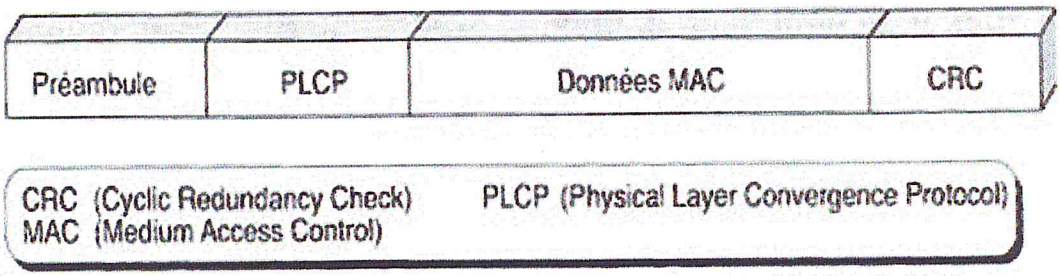


Figure II.12 : Structure d'une trame IEEE 802.11

Le PLCP contient des informations logiques utilisées par la couche physique pour décoder la trame, tandis que le préambule est dépendant de la couche physique et contient deux séquences :

- une séquence de synchronisation utilisée par le circuit physique pour sélectionner l'antenne à laquelle se raccorder.
- Une séquence SFD (Start Frame Delimiter), utilisée pour définir le début de la trame.

La zone "données MAC" transporte le protocole de niveau sous-jacent, comme illustré sur la figure II.13 :

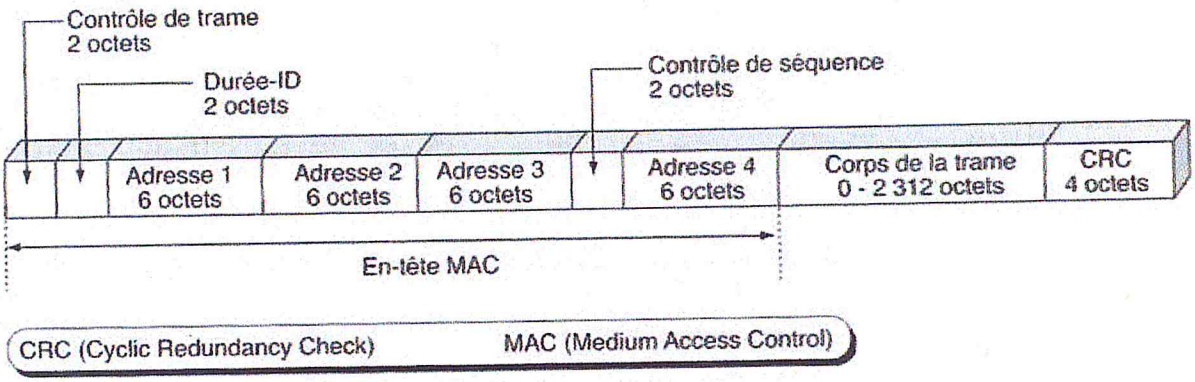


Figure II.13 : La zone MAC

Le champ *contrôle de trame* permet d'émettre des informations pour le contrôle de la trame.

Le champ *Durée-ID* : par rapport au type de trame, ce champ peut avoir deux sens différents :

- pour les trames de polling en mode d'économie d'énergie, c'est l'identifiant de la station.
- Pour les autres trames, c'est la valeur de durée de vie utilisée pour calculer le NAV.

Les champs *adresses* : une trame peut contenir jusqu'à quatre adresses :

- Adresse 1 correspond a l'adresse de la station destination.
- Adresse 2 correspond à l'adresse de la station source.
- Adresse 3 est l'adresse de la station source originale dans le cas ou la trame vient du système de distribution[15]. Dans le cas ou la trame est adressée au point d'accès, cette adresse correspond à l'adresse du terminal destination.
- Adresse 4 est utilisé lorsque la trame est transmise d'un point d'accès à un autre point d'accès lorsqu'on utilise un système de distribution mobile (Wireless DS).

Le champ *Contrôle de Séquence*, est utilisé pour spécifier l'ordre des fragments d'une trame fragmentée et pour reconnaître des paquets qui ont été dupliqués.

CRC : le CRC est sur 32 bits.

[15] Un système de distribution, ou DS (Distribution System), est un réseau généralement filaire à haut débit, qui consiste a interconnecté les différentes cellules dans le cas d'un réseau avec infrastructure.

Les trames de contrôle

- La trame RTS est illustrée par la figure II.14.

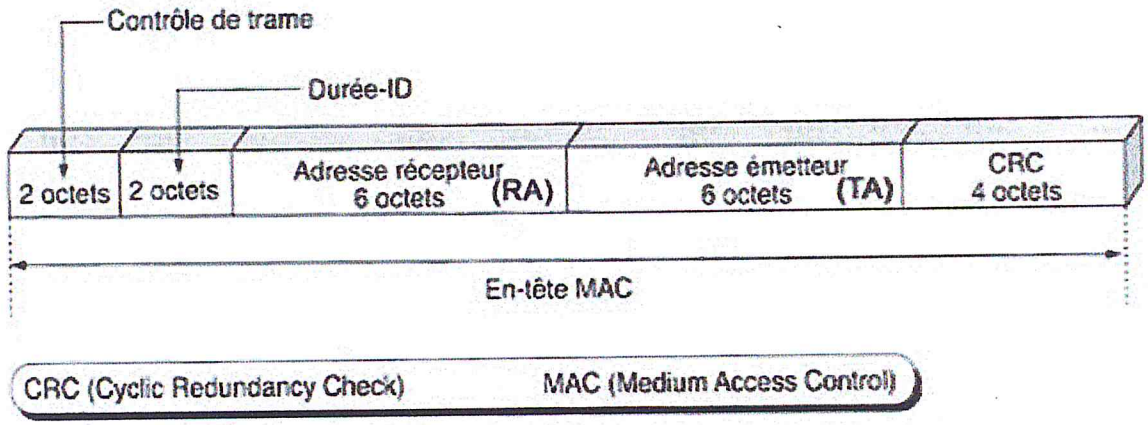


Figure II.14 : La trame RTS

- **RA :** Correspond à l'adresse de la station destination.
- **TA :** Correspond à l'adresse de la station source qui émet la trame RTS.
- **Durée- ID** : Champ durée de vie correspond au temps nécessaire à la transmission de la trame RTS, auquel on ajoute le temps de transmission d'une trame CTS et le temps de transmission d'une trame ACK ainsi que trois SIFS.

- La trame CTS est illustrée à la figure II.15 :

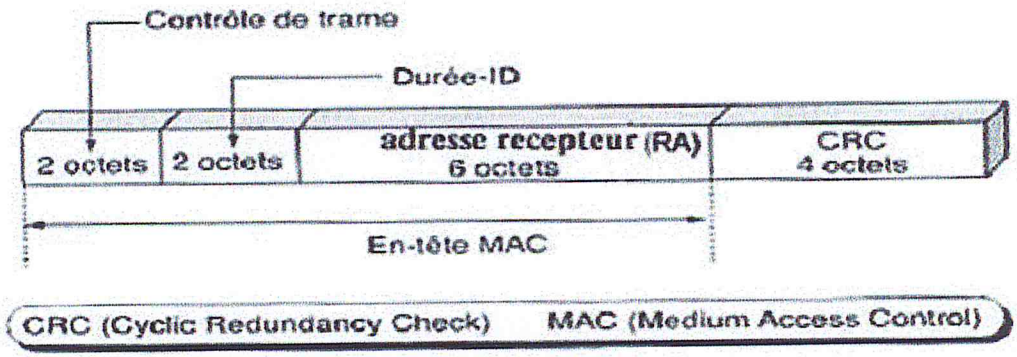


Figure II.15: La trame CTS

- **RA :** Correspond à l'adresse de la station source qui provient du champ TA de trame RTS.
- **Durée-ID :** Champ durée de vie correspondant à la valeur du durée de vie de la trame RTS moins le temps de transmission de la trame CTS et d'un SIFS.

- Enfin, la trame ACK, illustrée à la figure II.16

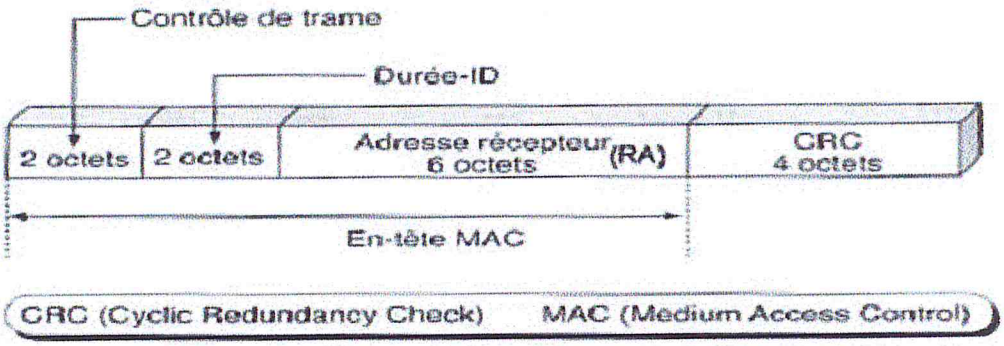


Figure II.16 : La trame ACK

Elle comporte les champs suivants :

- **RA :** correspond à l'adresse du terminal qui doit recevoir l'acquittement (elle est extraite du champ adresse 2 de la trame à acquitter).
- **Durée-ID** : si le bit More Fragment de la trame qui doit être acquitter possède la valeur 0 (le fragment en cours est le dernier fragment), le champ durée de vie a pour valeur 0. sinon, il correspond au champ durée de vie de la trame précédemment reçue moins le temps de transmission de la trame ACK et un SIFS.

II.4.2 : Le protocole MACA [Kar90]

Proposé par Karn en 1990, le protocole MACA (Multiple Access with Collision Avoidance) propose une solution au problème du terminal caché et du terminal exposé. Il a été inspiré du protocole CSMA\CA. Avec la présence des terminaux cachés dans le réseau, si l'émetteur écoute le canal et le trouve libre, cela ne veut pas dire qu'il puisse transmettre sans risque de collision. Inversement, avec la présence de terminaux exposés dans le réseau, si l'émetteur écoute le canal et le trouve occupé, cela ne veut pas dire qu'il ne peut pas transmettre. Donc l'écoute du support (partie CS de CSMA/CA) est inutile. Karn a pensé ignorer cette partie, ce qui a donné un nouveau protocole MA\CA ou MACA. Comparé au protocole 802.11 DCF, MACA n'utilise pas le PCS (Physical Carrier Sense).

MACA évite la collision en utilisant un mécanisme de réservation du support (RTS/CTS) différent de celui utilisé par 802.11 DCF. Dans MACA, toutes les stations qui reçoivent un RTS adressé à d'autres stations, stoppent leurs transmissions pendant un temps suffisant pour que l'émetteur reçoit le CTS, et non pas jusqu'à la fin de la transmission de données comme c'est le cas dans le

protocole 802.11. Et de la même façon, si une station reçoit un CTS qui ne lui est pas adressé, elle inhibe ses transmissions pendant un temps suffisant pour recevoir le paquet de données. De ce fait, on gagne en terme de bande passante. (Voir la Figure II.17.a et II.17.b) :

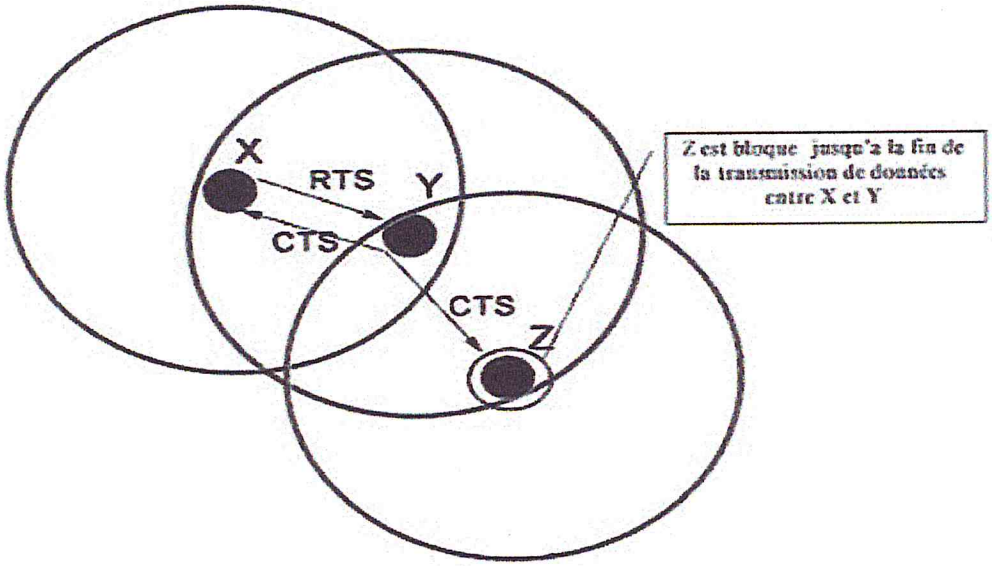


Figure II.17a : Premier exemple montrons le mécanisme RTS\CTS dans MACA

Le nœud Z ne peut entendre la transmission de X à Y, mais il peut entendre le CTS de Y. Le nœud Z doit donc inhiber ses transmissions, à la réception du CTS de Y, jusqu'à ce que la donnée de X soit complètement reçue par Y. Cependant, comment Z sait combien de temps il doit attendre après avoir reçu le CTS de Y ? L'émetteur ou l'initiateur du dialogue (X dans l'exemple) inclus dans le RTS la quantité de donnée qu'il veut transmettre, le récepteur (Y dans l'exemple) écho cette information dans son CTS. Ainsi, n'importe quelle station recevant un CTS peut savoir combien de temps elle doit attendre pour éviter la collision avec le paquet de données.

Les liens entre chaque paires de nœuds sont bidirectionnel (toutes les stations ont des puissances de transmission et des niveaux de bruit comparable). Quand une station reçoit un CTS qui ne lui est pas adressé (comme illustré dans l'exemple précédent), cela signifie que si la station émet, sa transmission sera probablement interférée avec la réception des données par le destinataire (celui qui a envoyé le CTS). MACA interdit cette transmission alors que CSMA le permet. Ainsi MACA pallie au problème des nœuds cachés. Contrairement à cela, si une station reçoit un

RTS adressé à d'autres stations, mais elle ne reçoit pas le CTS correspondant, cela signifie que le destinataire prévue du CTS est hors de la portée de communication de cette station. Voir la figure suivante :

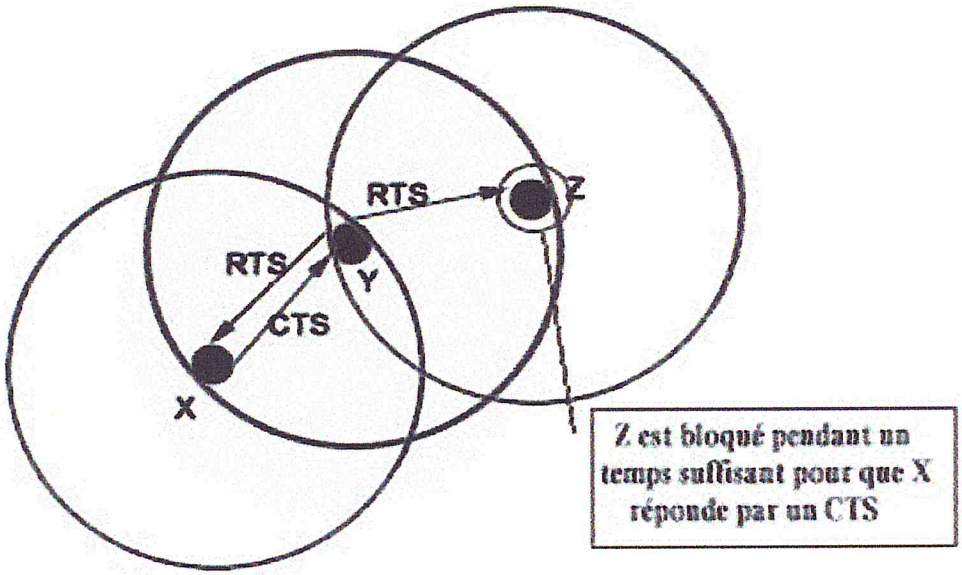


Figure II.17b : Deuxième exemple montrons le mécanisme RTS/CTS dans MACA

Le nœud Z reçoit le RTS de Y mais ne reçoit pas le CTS, dans ce cas, Z peut transmettre sans crainte d'interférer avec la transmission de données de Y à X (même s'il peut entendre cette transmission). MACA permet cette transmission alors que CSMA l'interdit. De cette façon, MACA remédie au problème du terminal exposé. Mais peut aussi causer un autre problème si le correspondant de Z est à la portée de X.

Le mécanisme RTS/CTS utilisé par MACA ne résout par totalement le problème de collision, plusieurs stations peuvent envoyer des paquets RTS simultanément, ce qui généré des collisions. Pour minimiser ce problème, MACA utilise un algorithme de Backoff similaire à celui qu'utilise CSMA. Quand une collision se produit entre deux ou plusieurs RTS, chaque station attend pendant un intervalle de temps choisi aléatoirement (un temporisateur) avant de réessayer d'envoyer. Après chaque tentative, la taille de l'intervalle est doublée. Cet intervalle doit être un multiple du « slot time » qui est la durée d'un RTS dans le cas du protocole MACA.

Comparé au protocole CSMA, MACA réduit les chances des collisions des paquets de données. Cependant, des collisions entre les paquets RTS peuvent se produire[16], mais cela est moins coûteux en bande passante par rapport aux collisions entre paquets de données, étant donné la taille réduite des paquets de contrôle (RTS, CTS). Dans le cas ou la taille de ces derniers est comparable à celle des paquets de données, MACA permet aux stations de se passé du dialogue RTS/CTS et de procéder directement à l'envoie de leurs paquets de données.

Plusieurs modifications de MACA ont été proposées. Fullmer et Garcia-Luna-Aceves ont proposé d'étendre MACA en lui ajoutant l'écoute du support avant la transmission. Le protocole résultant est nommé FAMA-NTR [Ful95]. D'autres modifications se sont portées sur MACA, supprimant les paquets RTS, principalement pour transmettre des messages multi-paquets ou supporter les flux temps réels. Pour la transmission des messages multi-paquets, Fullmer et Garcia-Luna-Aceves on proposé dans [Gar96] de remplacer tous les RTS excepté celui du premier paquet par plusieurs fanions dans le header des paquets de données afin d'augmenter l'utilisation du canal lors de la transmission. Et pour les application temps réels (multimédia), Lin et Gerla on proposé le protocole MACA/PR [Lin97] qui consiste à n'utiliser l'échange RTS/CTS que pour la transmission du premier paquet du flux.

D'autres extensions de MACA ont intégré d'autres étapes au dialogue RTS/CTS, le but étant de pallier aux erreurs de transmission. Par exemple dans [Sob96], les auteurs ont introduit un mini paquet pour inviter l'émetteur de retransmettre son dernier paquet en cas de perte (Acquittement négatif). Dans un autre cas, les trois étapes de MACA ont été étendu à cinq (protocole MACAW) [Bha94] avec des acquittements pour pouvoir détecter les trames perdues[17] types Stop & Wait. MACAW utilise aussi l'écoute du support pour éviter de perturber les RTS et CTS qui sont entrain d'être échangés. Mis à part que, chaque étape

[16] A cause de l'absence de l'écoute physique du canal (PCS).
[17] Dans MACA, il fallait attendre que les couches supérieures s'en rendre compte.

supplémentaire dans le dialogue introduit un temps TX-RX turn-around time[18], ainsi que des bits préambule (pour la synchronisation), des bits de contrôle et des bits checksum, ce qui réduit clairement le débit. Pour remédier à ce problème, un autre protocole MACA basé sur l'approche par invitation utilisant uniquement deux étapes à été proposé, ce protocole est nommé MACA-BI [Tal97].

II.4.3 Le protocole MACA-BI (MACA By Invitation) [Tal97]

Proposé en 1997, MACA-BI est un protocole MAC basé sur le principe d'invitation. Quand un nœud veut émettre des données, au lieu de demander la permission de transmission en émettant un RTS, il attend une invitation sous forme d'un paquet de contrôle de la part du destinataire prévu. Tel que nous avons vu précédemment, le protocole MACA est un protocole à trois étapes, c'est-à-dire qu'il passe par trois étapes (RTS-CTS-Données) pour la transmission d'une donnée. MACA-BI, par contre, est un protocole à deux étapes. Le RTS est supprimé, et le CTS est renommé RTR (Ready to Receive) indiquant que le nœud est prêt pour recevoir un certain nombre de paquets. Les deux étapes de MACA-BI sont illustrées par la figure II.19 :

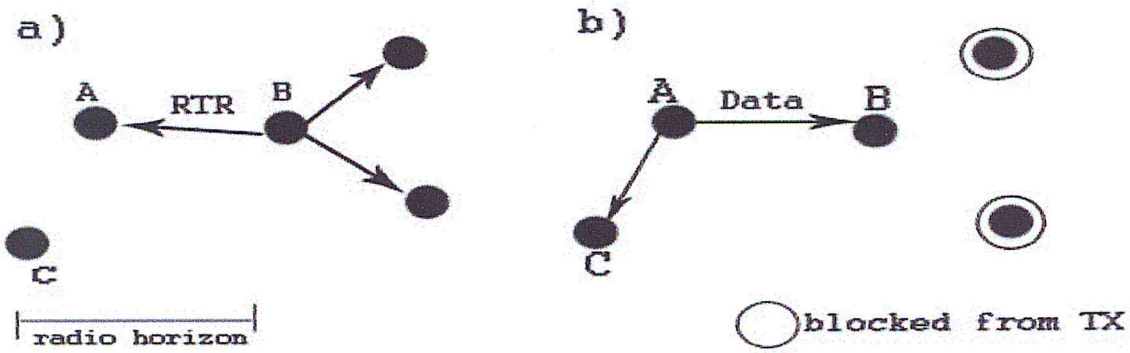


Figure II.19: Les deux cycles de MACA-BI

Notons que le nœud B n'a pas une connaissance exacte du buffer[19] du nœud A, il doit donc estimer le contenu du buffer ainsi que le taux d'arrivée moyen. Pour cela, chaque paquet de données doit contenir un certain nombre d'informations dites le Backlog (le nombre de paquets dans le buffer et leurs tailles). A partir du backlog et

[18] Le Turn-around time est la période de temps depuis la transmission jusqu'à la réception.
[19] Le buffer est un espace mémoire (une file d'attente) contenant l'ensemble des paquets qui attendent d'être transmise.

de l'historique antérieur, le nœud B peut décider combien de paquets il doit invité. Le nœud A répond donc par les paquets demandés ainsi que son nouveau backlog.

Cependant, l'approche par Invitation n'est efficace que dans le cas où le trafic est prédictible, chose qui permet à un nœud de prédire lequel de ses voisins à des paquets à émettre, ainsi que leurs nombre.

Pour accroître les performances de MACA-BI dans le cas d'un trafic non stationnaire, les nœuds ont la possibilité de transmettre un RTS explicite au cas où la capacité du buffer ou le délai ont dépassé un certain seuil. Ainsi, le destinataire prévu répond par un RTR pour permettre la transmission.

Le nombre réduit de cycles utilisé dans MACA-BI introduit un certain nombre d'avantages, à savoir :

- un turn-around time réduit.
- Les fonctionnalités de MACA sont préservées dans MACA-BI.
- MACA-BI est moins vulnérable à la corruption[20] des paquets de contrôle étant donnée qu'il utilise un nombre réduit de ces paquets.
- Le mécanisme par invitation (reveiver-driven) de MACA-BI pourvois automatiquement la régulation du trafic, le contrôle de flux, et le contrôle de congestion.

Comme MACA, le protocole MACA-BI empêche les collisions directes entre paquets de données (Data-Collision free protocol). Pour montrer cela, considérons le réseau à quatre nœuds de topologie linéaire[21] illustré par la Figure II.20 :

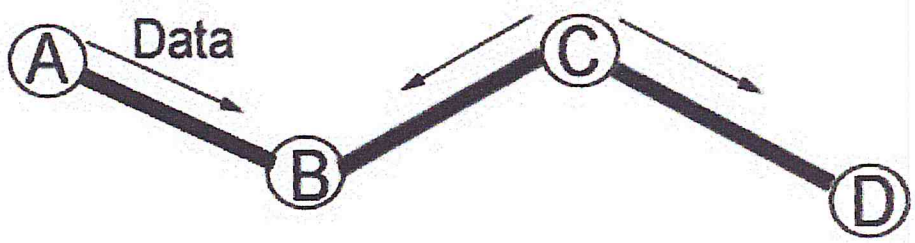


Figure II.20 : La collision entre paquets de données dans MACA-BI

[20] La corruption d'une trame est la transformation de la suite binaire transmise.
[21] Prouver cette propriété dans une topologie linéaire est suffisant pour déduire sa validité dans n'importe quelle topologie, puisqu'une topologie linéaire engendre un maximum de situations de « terminal caché ».

Une collision de données se produit lorsque A transmet des données à B en même temps que C transmet des données à B ou à D. Nous allons montrer qu'une telle collision est impossible dans MACA-BI :

- C transmet un paquet de données à B. cela est impossible du fait que le nœud B ne peut inviter qu'un seul nœud à la fois (Soit A ou C dans notre cas).
- C transmet un paquet de données à D, cela peut se passer seulement dans le cas ou C n'a pas entendu le RTR de B vers A. Nous avons alors deux cas :
  - B transmet un RTR à A au moment ou C a déjà transmis (un paquet RTR ou un paquet de donnée), cela est impossible puisque la transmission de C est entendu par B, ce qui l'empêche d'émettre un RTR à A.
  - B transmet un RTR à A pendant que C a reçu un RTR de D. la transmission de C n'est pas possible parce que le RTR de D va entré en collision avec le RTR de B au niveau du nœud C. Ainsi la transmission de donnée de C vers D est-elle empêché.

A partir de cela, nous concluons que la collision entre paquets de données ne peut se produire dans MACA-BI si aucune mobilité des nœuds n'est introduite. Toutefois, les collisions entre paquets de contrôle peuvent être engendrées que se soient des collisions directes[22] ou indirectes[23]. Ce genre de collisions peut aussi conduire a des collision entre paquets de contrôle et paquet de données, ce qui réduit clairement les performance du protocole. Une analyse détailler dans [Sob96] à montrer qu'aucun protocole MACA n'a pu éviter ce type de collisions. L'utilisation des Acquittements explicites (ACKs) est la seule solution pour pallier à ce type de perte de données.

Dans [Tal97] F. Talucci et al ont évalué les performances de MACA-BI dans deux cas, le premier cas concernant un réseau à saut unique (single-hop) et le deuxième un réseau multi-sauts.

Dans le premier cas, les auteurs ont développé dans un premier lieu un modèle analytique en utilisant les mêmes approches et hypothèses que dans [Ful95]. Dans

[22] Une collision directe se produit entre des nœuds qui se trouvant réciproquement dans leurs portées de communication. Une telle collision est engendrée à cause de l'échec de l'écoute du canal (carrier sens failure) dû au temps de propagation non nul.

[23] Une collision indirecte est une collision qui se produit entre des nœuds cachés l'un par rapport a l'autre et qui émettent à un voisin commun. Une telle collision est produite à cause du problème de terminal caché.

leurs analyse, les auteurs ont considérés un réseau sans fils à saut unique, totalement connecté (maillage complet), ce qui exclut par définition le problème du terminal caché. Dans leur expérience les auteurs ont assuré un nombre infini de nœuds générant un trafic poissonnier, avec un intervalle de temps δ entre paquets. Ils ont assuré aussi dans leur expérience que le canal est préparé pour une station quand elle a un paquet de données prêt a transmettre. Ils ont utiliser des paquets de données de 296 bytes, des paquets de contrôle de 20 bytes, un délai de propagation de 54 μs, et une vitesse du canal de 1Mbps. Le turn-aroun time est négligés. Ils ont étudiés le débit normaliser du protocoles MACA-BI par rapport aux protocoles CSMA non-persistent, FAMA-NTR[24] et MACA. A travers leurs expérience les auteurs ont concluez que MACA-BI fonctionne très bien dans un réseau à saut unique totalement connecté, et qu'il est comparable au autres protocoles MACA.

Dans le deuxième cas, les auteurs ont évalué les performances de MACA-BI dans un réseau multi-sauts contenant neuf nœuds. Pour cela, ils ont sélectionné trois topologies. La première est un anneau double avec étoile, elle ressemble presque à un réseau à saut unique totalement connecté, avec un minimum de situation du terminal caché. La deuxième topologie est une grille 3x3, elle montre un nombre important de terminaux cachés, et la troisième est une étoile où tous les voisins du nœud central sont cachés l'un par rapport à l'autre. Tous les liens ont une capacité de 1 Mbps. Dans leurs études, les auteurs ont comparé MACA-BI à deux autres protocoles, à savoir, CSMA non-persistant, et FAMA-NTR. Les auteurs ont assuré une parfaite prédiction de l'occupation des buffers, le nœud ayant le plus de paquets dans son buffer est invité à transmettre. Les auteurs ont mesuré le débit et le délai des paquets par rapport à la charge du réseau dans les trois topologies, et ils ont parvenus aux résultats suivants : les performance de CSMA dépassent celles de FAMA-NTR et MACA-BI dans la première topologie, ceci étant dû à l'effet minimal du problème de terminal caché dans cette topologie. Par conséquent CSMA est parfaitement adéquat dans ce cas. Des performances comparables ont été remarquées pour FAMA-NTR et MACA-BI. Comme le problème de terminal caché est devenu plus fréquent dans les deux autres topologies, les performances de MACA-BI on

[24] FAMA-NTR est un protocole basé sur MACA, auquel est ajoutée l'écoute non-persistant du canal.

clairement dépassé celles de CSMA. Seulement, ce qui a surpris les auteurs dans ce cas consiste en les faibles performances de FAMA-NTR. Les auteurs expliquent ce résultat par le débit non équilibré dans le réseau d'une part et d'autre part par le partage inéquitable du medium entre les différents transmetteurs, dans les deux cas l'approche orientée émetteur (Sender oriented) utiliser par FAMA-NTR est la cause de ce problème. Par contre, l'approche orientée récepteur (receiver oriented) qu'utilise MACA-BI peut mieux arbitrer la transmission entre les émetteurs concurrents. Une autre cause de la dégradation des performances de FAMA-NTR est le "turn-around time additionnel" pour chaque paquet et la transmission d'un seul paquet par cycle (dialogue RTS/CTS). Dans MACA-BI, plusieurs paquets peuvent être transmet par cycle.

Cependant, dans leurs études, les auteurs n'ont pas pris en considération l'effet de plusieurs critères importants sur les protocoles, notamment la mobilité qui constitue l'un des problèmes majeurs des réseaux ad hoc.

II.5 Conclusion

Dû à la mobilité et l'absence de tout sorte de contrôle central, l'accès au canal dans les réseaux ad hoc doit être effectué de façon distribués, c'est-à-dire qu'il n'y a pas d'entité centrale pour coordonner l'accès mais que chaque nœud est responsable d'assurer son accès au canal en évitant au maximum la monopolisation de ce dernier et les conflits avec ses voisins, l'approche compétition est alors plus adéquate aux réseaux ad hoc. Des protocoles MAC gérant la forte mobilité ainsi que l'accès distribuer sont donc nécessaires. Plusieurs protocoles MAC on été proposés. Dans la première partie de ce chapitre, nous avons présenté une classification détaillée des protocoles selon plusieurs critères. Nous avons étudié ensuite quelques problèmes spécifiques à la transmission hertzienne, notamment le problème du terminal caché et celui du terminal exposé. Dans la troisième partie nous avons étudiés les protocoles MAC les plus connus. Le tableau II.1 récapitule et compare les protocoles MAC étudiés :

Tableau II.1 : Récapitulatif et Comparatif des protocoles MAC étudiés

	CSMA	802.11 DCF	MACA	MACA-BI
- Initiation de transmission	- transmission initiée par l'émetteur.	- transmission initiée par l'émetteur.	- transmission initiée par l'émetteur.	- transmission initiée par le récepteur.
- l'écoute physique du canal PCS	Oui	Oui	Non	Non
- mécanisme RTS/CTS (VCS).	- pas de mécanisme RTS/CTS. La station qui veut transmettre écoute le canal. S'il est libre, elle émet directement sa trame, si le canal est occupé, elle attend un temps aléatoire (Backoff) puis elle répète la procédure depuis le début.	- utilise le mécanisme de réservation RTS/CTS quand la taille de la trame dépasse un certain seuil (seulement pour les grosses trames, auxquelles une retransmission est très coûteuse en bande passante). - utilise l'écoute non persistent[25]	- utilise le mécanisme de réservation RTS/CTS. Quand la taille des paquets de contrôle (RTS, CTS) est comparable a celle des paquets de données, les stations peuvent ne pas utiliser ce mécanisme.	- pas de mécanisme RTS/CTS. Le RTS est supprimé et le CTS est renommé RTR. Ce dernier est envoyé par le destinataire pour indiquer qu'il est prêt à recevoir un certain nombre de paquets.
- les divergences dans le mécanisme RTS/CTS.		- le nœud qui reçoit un RTS ou CTS qui ne lui est pas adressé, doit initialiser le NAV, pour qu'il reste inactif jusqu'à la fin de la transmission de données pour éviter les collisions avec les trames ACKs.	- quand un nœud entend un RTS ou CTS, il ne s'inhibe pas de transmettre jusqu'à la fin de la transmission de données mais il s'inhibe : • Pendant un temps suffisant pour que l'émetteur reçoit le CTS, dans le cas ou le nœud à reçu un RTS. • Pendant un temps suffisant pour recevoir	- les nœuds peuvent transmettre un RTS explicite dans le cas où la capacité du buffer ou le délais dépasse un certain seuil.

[25] Quand la station trouve le canal occupé, au lieu d'écouter jusqu'à ce qu'il se libère puis commencer à émettre, elle attend un temps aléatoire puis elle écoute le canal de nouveau. C'est ce que l'on appelle l'écoute non-persistent du canal.

			le paquet de données, dans le cas où le nœud a reçu un CTS.	
- les collisions	- des collisions entre paquets de données peuvent se produire à cause du problème de nœud caché.	- le mécanisme RTS/CTS résout le problème du nœud caché, mais il ne résout pas complètement celui de la collision. Plusieurs RTS peuvent être envoyés simultanément par plusieurs stations. En plus des collisions entre paquets de contrôle, des collisions entre paquets de données et paquets de contrôle peuvent aussi se produire.	- le mécanisme RTS/CTS permet d'éviter le problème de nœud caché, mais les collisions des paquets RTS restent toujours possibles. Des collisions entre paquets de données et paquets de contrôle sont aussi possibles.	- Comme pour MACA, et 802.11, les collisions directes entre paquets de données ne peuvent se produire. Cependant, les collisions entre paquets de contrôles, ou entre paquets de données et paquets de contrôles sont possibles.
- mécanisme de Backoff	- les stations utilisent un mécanisme Backoff dans le cas ou le canal est occupé.	- 802.11 utilise un mécanisme de Backoff pour réduire le problème de collision des RTS.	- un mécanisme backoff similaire à celui de CSMA et 802.11 est utilisé.	- utilise un mécanisme de backoff similaire a celui de MACA.
- mécanisme d'acquittem-ent	- pas de mécanisme d'acquittement au niveau liaison.	- utilise un mécanisme d'acquittement (ACK) au niveau de la couche liaison, ce qui permet a cette dernière de détecter les paquets perdus.	- pas d'ACK. En cas de perte de données, la couche liaison est incapable de le détecter. Il faut attendre que les couches supérieures s'en rendent compte.	- pas de mécanisme d'acquittement.
- les étapes de transmission	- l'écoute du canal puis l'envoi des données directement s'il est libre.	RTS-CTS-Donnée-ACK	RTS-CTS-Donnée	RTR-Donnée

CHAPITRE III : ENVIRONNEMENT DE SIMULATION

III.1 Introduction

La multitude des solutions proposées à chaque protocole dans chaque couche pour les réseaux sans fil, câblés, et les réseaux satellites ont mené à une explosion dans les choix possibles pour de tels réseaux. La dimension de ces réseaux fait de l'expérimentation et de la mesure antérieure au déploiement des actions impossibles. Cependant, les risques de déployer ces nouvelles technologies dans les situations critiques exigent l'assurance de leur bon fonctionnement, d'où la nécessité d'une technique qui permet de comprendre et de tester le fonctionnement du système et de prédire son évolution, ses propriétés et ses performances. La simulation est l'outil qui assure ces fonctions. Dans notre travail, nous avons eu recours à la simulation pour évaluer les performances des protocoles MAC dans les réseaux mobiles ad hoc. Pour cela nous avons utilisé le simulateur GloMoSim. Nous commençons dans ce chapitre par introduire des notions générales sur la simulation. Nous présenterons ensuite le langage de programmation utilisé pour développer GloMoSim, à savoir le langage PARSEC. La dernière partie de ce chapitre sera consacrée à la présentation du simulateur.

III.2 Introduction à la simulation

La simulation par ordinateur est apparue en même temps que l'informatique pour les besoins du projet Manhattan pendant la seconde Guerre mondiale, afin de modéliser le processus de dénotation nucléaire. Depuis, elle a évolué parallèlement à l'informatique.

De nos jours, la simulation connaît un essor considérable. Ceci est dû aussi bien à l'intérêt théorique que présente la modélisation des systèmes simulés, qu'aux besoins croissants de simuler par ordinateur des réalisations de plus en plus complexe.

On constate en effet que les conditions d'expérimentation sont de nos jours toujours plus coûteuses et moins flexibles. La simulation consiste à représenter la réalité simulée sur ordinateur, à donner à celle ci des moyens de perceptions réalisés artificiellement, puis à simuler son fonctionnement. C'est sur ce dernier point que la

simulation joue un rôle important en mettant à la disposition de l'utilisateur un environnement d'expérimentation dont on peut faire varier les paramètres.

Les conceptions et les techniques de programmation on fait de leur coté d'important progrès, en particulier dans les domaines de l'abstraction et du parallélisme. C'est ainsi que les langages de programmations d'aujourd'hui offrent des moyens performants, ce qui rend possible la réalisation d'un simulateur dans un environnement de programmation existant [ERA 96].

III.2.1 Limite de l'expérimentation directe

Une expérimentation directe effectuée sur le terrain peut se révéler coûteuse, irrationnelle ou même impossible. Il serait par exemple irrationnel de fermer arbitrairement l'un des guichets d'une banque et de laisser le public s'accumuler dans le hall pour le seul intérêt d'observer le phénomène. Il serait de même inconcevable dans notre étude de mettre en œuvre un réseau ad hoc, de déplacer les nœuds et changer les paramètres pour comparer les protocoles de la couche MAC.

C'est à cause des difficultés liée à l'expérimentation directe, dans le but aussi de pouvoir examiner facilement et rapidement des variantes de système étudié, que l'on cherche à réaliser un modèle de ce système dont on peut analyser numériquement le comportement et sur la base de cette analyse, inférer le comportement du système réel lui-même.

III.2.2 Notations et définitions

Dans cette partie, nous allons introduire quelques concepts principaux de la simulation:

Système : On appelle système un ensemble d'objets ou d'entités en interaction.

Système réel : On appelle communément système réel le phénomène que l'on veut observer. On y distingue d'une part les centres d'activités ou taches d'activités, c'est-à-dire les endroits où il se passe quelque chose, où un objet est crée, traité, transformé ou détruit, d'autre part les objets qui circulent dans le système et qui sont manipulés par différentes taches actives.

Modèle : Un modèle signifie la représentation de système réel ou imaginaire dans le but d'expliquer et prédire certains aspects de son comportement.

Etat : On appelle état d'un système (ou plus exactement d'un modèle de système), l'ensemble minimal de variables permettant de le décrire et de décrire son évolution future

Evénement : On appelle événement un phénomène capable de changer l'état du système.

Système discret: Un système discret est un système dans le quel les variables décrivant un état ne changent de valeur qu'en un nombre fini de point sur l'axe du temps. La station de service est un exemple de système discret, puisque les variables qui y figurent ne changent de valeur qu'à des instants précis. Par exemple le nombre de client dans la station ne se modifie qu'à l'arrivée ou au départ d'un client.

Système continu : Un système continue est un système dans lequel le temps s'écoule de façon continue et ou les variables peuvent changer de valeur à tout instant. La simulation du vol d'un avion est un système continu en ce sens que les cordonnées et la vitesse de l'avion sont des fonctions qui prennent une valeur en tout point de l'axe du temps. Cela ne veut pas dire bien sur que les fonctions elles mêmes sont continues. Le terme continu se rapporte à la perception du temps dans la simulation.

Simulation : La simulation est une technique de modélisation du monde réel. Elle permet de représenter le fonctionnement d'un système composé de différents centres d'activité, de mettre en évidence les caractéristiques de ceci et les interactions entre eux, de décrire la circulation de différents objets traités par ces processus, et en fin d'observer le comportement du système dans son ensemble et dans son évolution dans le temps.

III.2.3 Modèle de simulation

On distingue plusieurs modèles de simulation selon s'ils sont statiques ou dynamiques, déterministes ou aléatoires, continus ou discrets.

Un modèle de simulation est dynamique ou statique selon si les valeurs de ses variables se modifient dans le temps ou non. Il est déterministe s'il ne contient aucune variable aléatoire, il est aléatoire dans le cas contraire. Dans le cas d'un

système aléatoire, les résultats de la simulation sont eux même aléatoires et ne donnent qu'une estimation du comportement du système simulé. Enfin, un modèle de simulation est continu si l'ensemble des instants considéré forme un intervalle compact sur l'axe des temps. Il est discret si celui-ci se compose d'un nombre fini ou dénombrable de valeurs isolées.

III.2.4 Gestion du temps et de l'Echéancier

La simulation consiste à gérer le temps ainsi que les actions qui sont liées aux différents instants du système réel et à faire fonctionner abstraitement le modèle qui représente ce système.

Un événement, dans la simulation, est un bloc de données : une date d'occurrence, une identité, etc. Ces données décrivent de façon exhaustive la modification du vecteur d'état. Dans la mesure où à un instant donné de nombreux événements peuvent être en attente d'émergence, on comprend la nécessité d'optimiser la gestion des données correspondantes. C'est le rôle de *l'échéancier* (event list), qui les stocke, les classes, pour les extraire à la date choisie. Un échéancier est donc une liste d'événements ordonnés chronologiquement selon l'heure à laquelle ils doivent être activés. Chaque événement inséré dans l'échéancier en est retiré lorsque son heure d'activation est égale à celle du système de simulation (donné par l'horloge centrale). Des actions associées à cet événement sont alors exécutées, ensuite l'horloge du système est avancée à l'heure d'activation de l'événement suivant. Le temps est donc géré par l'échéancier et par l'horloge centrale, ceci dans un modèle discret. Dans un modèle continu, le temps est discrétisé selon un pas donné et, à chaque avance du temps, les valeurs des variables du système sont mises à jour [ERA 96].

III.2.5 Simulation par événements discrets

La simulation par événements discrets désigne la modélisation d'un système réel tel qu'il évolue dans le temps, par une représentation dans laquelle les grandeurs caractérisant le système (variable) ne changent qu'en un nombre fini ou dénombrable de points isolés dans le temps. Ces points sont les instants où se passent les événements, c'est-à-dire le phénomène capable de modifier l'état de système et

nous appelle événement tout changement d'état du système réel se produisant à un instant donné, ainsi que les actions qui accompagnent ou caractérisent ce changement.

La simulation par événement discret consiste alors à prendre en compte dans la modélisation des taches actives, les seuls instants ou un événement se produit, et à concentrer l'activité des taches simulées sur ces instants là. Nous considérons comme système de simulation par événements discrets tous les systèmes basés sur cette abstraction [ERA96].

III.2.6 Simulateur

Un simulateur est une "maquette" logicielle, complétée par un modèle de l'environnement, que l'on fait évoluer pour y mesurer à loisir les grandeurs critiques.

La tache première d'un simulateur est d'assurer que la chronologie des événements soit respectée. A chaque occurrence d'un événement, les actions qui sont associées à celui ci sont exécutées

Il existe plusieurs simulateurs pour les réseaux mobiles, les plus connu sont NS2, OPNET et GloMoSim. Dans notre travail nous avons choisie d'utiliser GloMoSim. C'est un simulateur a événement discret conçu principalement pour simuler les réseaux ad hoc. Il est donc plus adapté aux réseaux ad hoc que les autres simulateurs.

III.3 PARSEC

PARSEC (PARallel Simulation Environnement for Complex system) est un langage de simulation à événement discret basé sur le langage C.

Développé à l'université d'UCLA (University California Los Angeles) le langage PARSEC fut dérivé du langage MAISIE, ayant subi d'importantes améliorations notamment dans sa syntaxe et son environnement d'exécution.

Le langage PARSEC adopte une approche d'interaction entre processus lors de la simulation d'événement discret grâce à la notion d'objet (l'objet fait référence à un processus physique) qui est modélisé dans le système physique par un processus logique dit « entity ». L'interaction entre processus physique (events) est modélisée par l'échange de messages entre les entités correspondantes. Un des importants

avantages du langage parsec est sa capacité d'exécuter un modèle de simulation à événement discret en utilisant différents protocoles de simulation parallèle asynchrone sur différentes architectures parallèles.

Le langage parsec fut conçu de manière à séparer le modèle de simulation et les algorithmes (séquentielle ou parallèle) utilisés lors de l'exécution du modèle.

III.3.1 Notations et définitions

III.3.1.1 Entité

Un programme parsec est une collection de fonctions écrites en langage C et de définitions d'entités. Une entité décrit une classe d'objets, dont chaque instance de type d'entité peut être crée pour modéliser un objet dans le système physique.

Chaque programme dans PARSEC doit avoir une entité principale « driver » qui initialise l'exécution de la simulation qui a le rôle de la fonction « main » du langage C. La déclaration d'une entité est similaire à la déclaration d'une fonction en C, mais l'entité ne retourne pas de valeur, cette déclaration définit un type d'entités « ename » sur lequel on peut créer plusieurs instances, en exécutant l'instruction 'new'.

III.3.1.2 Message

PARSEC utilise des objets de type « message » composés d'un nom et une liste de paramètres. La définition d'un message est similaire à la définition de structure en « C », les paramètres de message peuvent inclure des tableaux de taille fixe et qui passent par valeur.

Il est possible d'utiliser les pointeurs comme des paramètres de message, mais cela est dangereux dans une architecture de mémoire distribuée.

Les entités communiquent entre elles en utilisant les messages. Chaque entité a un buffer unique. Des primitives d'envoi et de réception asynchrones de messages sont fournies pour respectivement le dépôt et l'enlèvement des messages du buffer. Ces primitives sont :

- send ( ) : Une entité envoie un message par l'instruction send, chaque message sera estampillé et déposé dans le buffer d'entité destinataire.
- receive ( ) : Une entité accepte les messages de son buffer en exécutant l'instruction receive.

III.3.1.3 Evénement

Chaque événement dans le modèle de simulation à événement discret simule certaines activités dans le système physique qui entraîne d'autres objets, chaque événement est associé à une estampille qui indique le temps auquel l'événement correspondant apparaît dans le système.

Le temps de la simulation peut seulement avancer quand l'entité reçoit un message ou quand elle exécute l'instruction « hold ».

III.3.2 L'exécution parallèle

Le langage parsec utilise différents algorithmes de synchronisation séquentiels (global events list), et parallèles (conservatrice et optimiste) qu'on abordera par la suite.

III.3.2.1 Restriction :

Les programmes à exécution parallèle doivent obéir aux restrictions suivantes :

- Variables globales : la simulation parallèle en PARSEC ne doit pas utiliser les variables globales. Il est interdit pour deux entités d'avoir un pointeur sur la même donnée. L'état d'une entité doit être seulement déterminé par des variables déclarées dans cette entité, et ces variables ne doivent pas être partagées avec d'autres entités. Ces restrictions sont pour les raisons suivantes :
  - ✓ PARSEC assume que tout changement dans l'état de l'entité est le résultat de message, cette règle est violée avec un état d'entité globale.
  - ✓ La simulation optimiste à besoin que l'état d'entité sauvegardé ne fonctionne pas avec l'état partagé, les variables globales ne sont pas contrôlées par le système.
  - ✓ La mémoire partagée a une copie des variables globales quant à la mémoire distribuée elle a N copies non consistantes.
- Les pointeurs ne peuvent pas être passés dans des paramètres de message ou les paramètres d'entité.

III.3.2.2 Partitionnement

Lors d'une implémentation parallèle, le modèle de simulation doit être partitionné en assignant des entités parmi les processeurs. Le langage parsec

III.3.3 Compilation et exécution du PARSEC

a) Pour WINDOWS

Le langage parsec fonctionne sur une plate forme régie par un compilateur C/C++, d'où il est important d'installer Visual C++ v5 ou v6.

Il faut ensuite créer les variables d'environnement de VC++ si elles n'existe pas déjà, ou leurs ajuster ces valeurs :

INCLUDE

C:\Program Files\Microsoft Visual studio\VC98\Include;

C:\Program Files\Microsoft Visual; Studio\VC98\MFC\Include.

LIB

C:\Program Files\Microsoft Visual studio\VC98\Lib;

C:\Program Files\Microsoft Visual Studio\VC98\MFC\Lib.

PATH (système) :

C:\Program Files\Microsoft Visual Studio\VC98\Bin;

C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin.

Tel que *C:\Program Files\Microsoft Visual Studio* est le chemin de VC++.

Il faut aussi spécifier les chemins de parsec dans les variables d'environnements[26] :

INCLUDE : C:\Parsec\include

LIB : C:\Parsec\runtime

PATH : C:\Parsec\bin

b) Pour LINUX

La configuration de parsec est quelque peut différente. Nous devons exécuter les commandes suivantes :

```
# !/bin/csh
setenv PCC_DIRECTORY "le chemin du répertoire parsec"
${PCC_DIRECTORY}/bin/parsecc $*
```

Nous mettons le chemin de parsecdir/bin dans la variable PATH.

Voir ANNEXE A, pour plus de détaille sur le langage PARSEC

[26] PARSEC doit être implémenté dans la racine (partition C du disque dur), et il est recommandé d'éliminer les sous répertoires de manière à avoir C : Parsec : bin/lib/inculde...etc.

III.4 GloMoSim

GloMoSim [Xia98, JAY99] est un environnement de simulation à grande échelle pour les réseaux sans fil et filaires. Il a été conçu en utilisant la capacité de la simulation parallèle fournie par Parsec.

Comme la plupart des systèmes réseau, GloMoSim est conçu en se basant sur une approche en couches, similaire à l'architecture sept couches du modèle OSI. Comme le montre le tableau III-1 :

Couche	Protocoles
Mobilité	Random drunken et Random waypoint
Radio Propagation	Free space avec radio capture et sans capture
Liaison de donnée (MAC)	CSMA, IEEE 802.11 et MACA
Réseau (Routage)	IP avec AODV, Bellman-Ford, DSR, Fisheye, LAR scheme 1, NS DSDV, OSPF or WRP
Transport	TCP et UDP
Application	CBR, FTP, HTTP et Telnet

Tableau III-1 : Architecture en couches de GloMoSim

Des simples APIs sont définis entre les différentes couches du simulateur, le but étant de rendre plus rapide et plus facile aux développeurs d'intégrer de nouveaux modèles et protocoles au niveau des différentes couches.

Contrairement au simulateur réseau existant (OPNET et NS2 par exemple), GloMoSim a été conçu avec, comme principal but, de simuler des réseaux ad hoc très larges pouvant supporter plus d'un million de nœuds, en utilisant l'exécution parallèle pour réduire le temps d'exécution de la simulation. Pour atteindre cette scalabilité, GloMoSim utilise des techniques d'agrégation de nœuds et de couches, qui consiste à multiplexer plusieurs nœuds ou couches dans une seule entité PARSEC.

III.4.1 La technique d'agrégation des nœuds

Dans PARSEC, une approche simple pour la conception d'un modèle de simulation du réseau est de créer pour chaque nœud du réseau une entité. Même facile à comprendre, cette approche cause des problèmes :

- Si une entité doit être crée pour chaque nœud, les exigences de la mémoire augmentent pour un modèle avec un grand nombre de nœuds. Chaque entité est un processus indépendant qui exige une mémoire additionnelle de travail.

- La performance de la simulation se dégrade à cause du changement de contexte entre plusieurs entités.

Pour circonvenir ces problèmes, l'agrégation des nœuds a été introduite dans GloMoSim. Avec l'agrégation des nœuds, une seule entité peut simuler plusieurs nœuds du réseau dans le système. Quand le code de simulation pour un nœud particulier est exécuté, il n'a pas accès aux structures de données des autres nœuds.

La technique de l'agrégation des nœuds implique que le nombre des nœuds dans le système peut être augmenté en maintenant le même nombre d'entités dans la simulation. En fait, la seule exigence est que nous avons besoin seulement autant d'entités que de processeurs sur lesquels la simulation est exécutée. D'ici, une simulation séquentielle a besoin seulement d'une seule entité.

Dans GloMoSim, chaque entité représente une région géographique de la simulation. Les nœuds représentés par une entité particulière sont déterminés par leurs positions physiques. Chaque entité représente une région rectangulaire régulière (partition), une partition peut avoir huit partitions voisines au plus. Quand un nœud du réseau envoie un message, le message doit être envoyé au plus aux huit entités voisines.

Notons que si chaque entité représente un seul nœud, l'envoi d'un message devient très difficile. Deux options sont possibles pour l'envoi d'un message :

- La première option consiste à ce que chaque entité garde trace des autres entités dans son domaine de puissance. Cette option est difficile car la topologie du réseau change constamment lorsque la mobilité est introduite dans la simulation.
- La deuxième option est quand un nœud envoie un message. Il sera envoyé à toutes les autres entités dans la simulation. L'entité de la réception acceptera le message s'il est dans la portée de l'envoyeur.

Donc, une transmission du message devient très compliquée quand l'agrégation du nœud n'est pas utilisée.

III.4.2 La technique d'agrégation des couches

Une simple approche pour représenter les différentes couches dans la simulation consiste à attribuer une entité PARSEC pour chaque couche. Cependant, agréger les couches dans une seule entité est indispensable pour plusieurs raisons, à savoir :

- Il arrive souvent dans une simulation que des couches différentes aient besoin d'accéder à une variable globale (variable global indiquant l'état du CPU par exemple). Si nous considérons que les couches sont représentées par des entités différentes, il n'existe pas une méthode élégante qui permet d'accéder aux variables partagées. Les variables globales ne peuvent être utilisées dans de tel situation, parce qu'elles causent des problèmes avec les accès concurrents durant l'exécution parallèle.
- Si chaque couche est vue comme une entité différente dans la simulation, elle doit explicitement garder trace des identificateurs des couches supérieures et inférieure. Cela est nécessaire pour assurer l'échange des messages entre couches. En plus, pour l'exécution parallèle conservative, chaque entité doit spécifier la source et l'ensemble de destination des entités communicantes aussi bien que les valeurs de lookahead. Spécifier les valeurs lookahead pour une entité peut être une tache très difficile, ceci crée un travail additionnel pour le développeur de protocole qui est fondamentalement intéressé à modéliser un protocole de réseau particulier.

Pour ces raisons, les concepteurs de GloMoSim ont décidé d'intégrer les différentes couches dans une seule entité. Chaque couche est maintenant implémentée en trois fonctions appelées par le protocole associé. La première fonction est une fonction d'initialisation. Elle est appelée pour chaque couche de chaque nœud en début de la simulation. La deuxième fonction est automatiquement appelée quand une couche particulière d'un nœud particulier reçoit un événement ou paquet, en se basant sur le contenu du message, l'instruction appropriée doit être exécutée. La troisième fonction est appelée à la fin de la simulation pour collecter les statistiques finales.

III.4.3 Structure des répertoires de GloMoSim

GloMoSim est structuré en plusieurs sous-dossiers, à savoir:

application: contient les fichiers sources de la couche d'application

bin: contient l'exécutable et les fichiers d'entrée/sortie.

doc: contient une documentation pour le simulateur.

include: contient les fichiers d'inclusion (.h).

mac: contient les fichiers sources pour la couche mac.

main: contient les fichiers principaux tel que driver.pc (le point d'entrée)

network: contient les fichiers sources pour la couche réseau.

radio: contient les fichiers sources pour la couche radio.

transport: contient les fichiers sources pour la couche transport.

scenario: contient les fichiers d'entrées et d'initialisation pour les différents scénarios.

Cette organisation facilite la tache de modification et d'amélioration telle que l'ajout d'un nouveau protocole dans une couche particulière, ou l'ajout d'un nouveau calcul statistique.

III.4.4 Installation de GloMoSim

Avant d'exécuter GloMoSim, il faut s'assurer que PARSEC est correctement installé. Il faut ensuite compiler les fichiers sources de GloMoSim pour obtenir l'exécutable :

Les étapes de création de l'exécutable sont :

- Sous UNIX :
  - Exécuter "make depend" afin de créer la liste de dépendances dans le "Makefile".
  - S'assurer que le bon chemin pour le compilateur Parsec est spécifié dans le Makefile pour la variable "par".
  - Aller dans ./glomosim/main et Lancer "make" pour créer l'exécutable
- Sous Windows Nt:
  - Il suffit d'utiliser Makent.bat qui se trouve dans ./glomosim/main pour créer l'exécutable. Ce fichier contient toutes les commandes pour compiler et lier les fichiers. On n'a qu'à exécuter makent, mais il faut s'assurer que le chemin de

pcc.exe ainsi que le fichier cl.exe du compilateur visual c++ apparaît dans la variable système "path" car makent fait des appels directs à pcc.exe pour compiler et lier les fichiers « .pc » et cl.exe pour compiler les fichiers ".c" et ".cpp".

Si la compilation est effectuée avec succès, un fichier exécutable "glomosim.exe" est crée dans ./glomosim/bin. Pour vérifier que GloMoSim est crée correctement, il suffit d'exécuté "./glomosim config.in". On doit avoir en sortie un fichier de statistique "glomo.stat". On compare ce dernier avec "glomo.stat.sample" qui se trouve dans le même répertoire. On pourra s'assurer que l'exécutable génère les bons résultats.

Le fichier config.in est appelé le fichier d'entrée ou fichier de configuration. Il contient tous les paramètres nécessaires pour la simulation. Le contenu du fichier d'entré est expliqué en détail dans le paragraphe suivant.

III.4.5 Description du fichier d'entrée

Le fichier d'entrer représente la configuration matérielle et logiciel du réseau à simuler, il contient tous les paramètres nécessaires pour le bon déroulement de la simulation. Dans ce fichier, les lignes qui commencent par un "#" sont des commentaires. On peut classer les paramètres d'entrée comme suit :

- **Paramètres de simulation générale :**
  - Temps de simulation : SIMULATION-TIME 15 (seconde est l'unité par défaut). Ce paramètre désigne le temps maximum de la simulation (15 secondes dans cet exemple). Il peut être suivi par une lettre pour indiqué l'unité du temps, les unités possibles sont : NS : nano secondes, MS : milli secondes, S : seconde, M : minute, H : heure, D : jour.
  - SEED : c'est un nombre utilisé pour initialiser la génération des nombres aléatoirement produits dans la simulation, c'est-à-dire pour initialiser le moteur de génération des nombres aléatoires. Le changement du SEED permet de changer les nombres aléatoires de la simulation.

- Paramètres de topologie et mobilité

oTerrain: TERRAIN-DIMENSIONS (2000, 800) : Ces deux paramètres représentent la dimension du terrain dans lequel les nœuds vont être simulés. L'unité est le mètre.

Dans ce cas, nous avons un terrain d'une longueur de 2000 mètres, et d'une largeur de 800 mètres.

oNombre de nœuds : NUMBER-OF-NODES 50

Ce paramètre représente le nombre de nœuds à simuler.

oL'emplacement initial des nœuds NODE-PLACEMENT : ce paramètre représente la stratégie de placement des nœuds au début de la simulation. Les stratégies implémentées actuellement sont :

- RANDOM: Les nœuds sont placés aléatoirement dans le terrain physique.

- UNIFORM: Basé sur le nombre des nœuds dans la simulation, le terrain physique est divisé en plusieurs cellules, dans chaque cellule un nœud est placé aléatoirement.

- GRID: Si on choisit cette stratégie, il faut aussi préciser un autre paramètre (GRID-UNIT). Le placement des nœuds commence à (0, 0), et les noeuds sont placés dans un format de grille. La distance qui sépare chaque nœud de ses voisins est précisée par le paramètre GRID-UNIT. Pour cette stratégie, le nombre des nœuds doit être le carré d'un nombre entier.

- FILE: Les positions des nœuds sont lues dans un fichier, NOEUD-PLACEMENT-FILE est le paramètre qui prend le nom de ce fichier. Chaque ligne du fichier contient les cordonnées x et y d'un nœud séparé par un espace.

```
NODE-PLACEMENT     RANDOM
#NODE-PLACEMENT    UNIFORM
#NODE-PLACEMENT    GRID
#GRID-UNIT         30
#NODE-PLACEMENT    FILE
#NODE-PLACEMENT-FILE nodes.input
```

La stratégie RANDOM est choisie pour ce cas.

- Modèle de Mobilité

S'il n'y a aucun mouvement des nœuds dans le modèle, le paramètre MOBILITY est mis à NO, sinon il prend comme valeur le type de mobilité. Les modèles de mobilité implémentés actuellement sont :

- RANDOM-WAYPOINT: dans ce modèle, le nœud sélectionne aléatoirement une destination du terrain physique. Il se déplace vers cette destination avec une vitesse entre MOBILITY-WP-MIN-SPEED et MOBILITY-WP-MAX-SPEED. Après qu'il atteint sa destination, le nœud reste pour une période de temps égale à MOBILITY-PAUSE.
- RANDOM-DRUNKEN: dans ce modèle, si un nœud est placé dans la position (x, y), alors après chaque MOBILITY-INTERVAL, il peut se déplacer aléatoirement à l'une des ses positions avoisinantes, à savoir (x-1, y), (x+1, y), (x, y-1), et (x, y+1), ceci est possible si et seulement si la nouvelle position est dans le terrain physique.

  Les valeurs de MOBILITE-INTERVAL et MOBILITE-PAUSE sont représentées dans le même format vu pour le paramètre SIMULATION-TIME.
- TRACE: Dans ce cas le mouvement des nœuds est lu à partir du fichier présenté par MOBILITY-TRACE-FILE.

  Exemple :

  MOBILITY TRACE

  MOBILITY-TRACE-FILE mobility.in

  Dans cet exemple, le fichier qui contient l'historique du déplacement est nommé mobility.in. Le format de chaque ligne du fichier est:

  <temps de déplacement> <nœud> <position de la destination en x> <position de destination en y> <vitesse de déplacement>

  Pour un nœud donné, les temps de déplacement doivent être chronologiques.

- Modèle Radio

- Type : deux types de modèles sont implémentés au niveau de la couche radio :
  - Radio Accnoise : représente le modèle radio standard.
  - Radio Nonoise : représente un modèle radio abstrait.

RADIO-TYPE RADIO-ACCNOISE

#RADIO-TYPE RADIO-NONOISE

- La bande passante: Le paramètre suivant représente la bande passante utilisée par les nœuds pour transmettre des messages.

  BANDWIDTH 2000000

  Dans ce cas la bande est 2000000 bps
- Puissance de transmission radio (en dBm): RADIO-TX-POWER
- Puissance minimum pour recevoir un paquet (en dBm): RADIO-RX-THRESHOLD
- Domaine de puissance (POWER-RANGE) : appeler aussi "porté de communication", ce paramètre représente la distance maximal dans laquelle un signal émis peut être reçu correctement.

- Protocole MAC

Les paramètres suivants spécifient le choix du protocole utilisé au niveau de la couche MAC :

MAC-PROTOCOL 802.11

#MAC-PROTOCOL CSMA

#MAC-PROTOCOL MACA

#MAC-PROTOCOL MACA-BI

#MAC-PROTOCOL TSMA

#TSMA-MAX-NODE-DEGREE 8

Dans ce cas, le protocole utilisé au niveau MAC est 802.11.

- Protocole de Routage et protocole IP

Les paramètres suivants spécifient les protocoles de routage, ainsi que le protocole utilisé au niveau IP :

#ROUTING-PROTOCOL BELLMANFORD

ROUTING-PROTOCOL AODV

#ROUTING-PROTOCOL DSR

#ROUTING-PROTOCOL LAR1

#ROUTING-PROTOCOL WRP

#ROUTING-PROTOCOL FISHEYE

#ROUTING-PROTOCOL ZRP

#ZONE-RADIUS 2

#ROUTING-PROTOCOL STATIC

#STATIC-ROUTE-FILE ROUTES.IN

NETWORK-PROTOCOL IP

NETWORK-OUTPUT-QUEUE-SIZE-PER-PRIORITY 100

Dans ce cas, les protocoles AODV et IP sont sélectionnés. La taille de la Queue IP est de 100 paquets.

- Spécification de la couche application

La liste des générateurs d'application ou de trafic à exécuter est spécifiée par le fichier donné en argument au paramètre APP-CONFIG-FILE. Chaque ligne de ce fichier peut spécifier aussi bien une session d'application FTP, TELNET, ou CBR. Le format de cette ligne est le suivant :

- FTP :
  - FTP <source> <destination > <items à envoyer> <temps début>
- TELNET :
  - TELNET : <source> < destination > <durée> < temps début >
- CBR (Constant Bit Rate) :
  - CBR : <source> <destination> <items à envoyer> <taille de l'item> <intervalle>

    <temps début> <temps fin>

Tel que :

<source> : identificateur du nœud source

<destination> : identificateur du nœud destination

< items à envoyer> : le nombre de paquets à envoyer

<taille de l'item> : la taille du paquet

<intervalle> : durée entre 2 envois de paquets de la couche application.

<temps début> : temps de début de session

<temps fin> : temps de fin de session

<durée> : la durée de la session

- Type de statistique

Chaque type de statistique est représenté par un paramètre qui peut prendre la valeur YES si on veut faire apparaître les statistiques de ce type dans le fichier de sortie (glomo.stat), et NO dans le cas contraire.

APPLICATION- STATISTICS YES/NO
TCP-STATISTICS YES/NO
UDP-STATISTICS YES/NO
ROUTING-STATISTICS YES/NO
NETWORK-LAYER-STATISTICS YES/ NO
MAC-LAYER-STATISTICS YES/NO
RADIO-LAYER-STATISTICS YES/NO
CHANNEL-LAYER-STATISTICS YES/NO

Le fichier résultat de la simulation contient des statistiques sur les paramètres et les protocoles choisis. Chaque ligne dans ce fichier a le format suivant :

Node : < numéro de nœud>, Layer :<nom de la couche>, <statistique>

Les différentes statistiques obtenues pour chaque couche sont présentées en ANNEXE B.

III.5 Conclusion

Grâce à la simulation, on peut étudier le comportement du modèle sous des aspects qui ne sont pas directement accessibles dans les phénomènes réels :

- aucun phénomène réel ne permet de contracter ou de dilater l'échelle des temps pour procéder à une étude détaillée,
- aucun système réel ne répondra instantanément à la question "que se passe-t-il si...?" mais exigera des procédures longues et difficiles pour modifier les conditions d'une expérimentation, ce qui aura pour effet de décourager l'observateur.

La simulation est donc un outil puissant et universel. Les gains à tirer d'une expérience de simulation sont variés: description (validation d'une architecture), explication (observation, expérimentation sur une maquette), et prédiction (mesure de performances, ou prévision de comportement). Son champ d'application est très large, elle est utilisée dans les domaines scientifiques, économiques, pédagogiques, ...etc. Dans notre travail nous avons eu recours à la simulation dans le domaine des réseaux mobiles ad hoc.

Dans la première partie de ce chapitre nous avons présenté quelques concepts importants de la simulation. La deuxième partie a été consacrée au langage PARSEC. Dans la dernière partie nous avons décrit le simulateur GloMoSim, qui constitue la plate-forme que nous utiliserons pour simuler, et effectuer une étude comparative entre les protocoles MAC. Cela fera l'objet du chapitre suivant.

CHAPITRE IV : SIMULATION DES PROTOCOLES

IV.1 Introduction

L'expérimentation directe d'un réseau ad hoc tel que nous l'avons constaté dans le chapitre précèdent est assez coûteuse, voir impossible, de plus elle n'est pas flexible et ne permet pas de bonne étude comparative. Ceci étant, nous avons opté pour le passage par la simulation pour réaliser notre étude.

L'unique étude comparative des protocoles MAC a été réalisée par C.L. Barret et al. Ils ont étudié trois protocoles à savoir, 802.11, MACA et CSMA. Cependant plusieurs critères importants n'ont pas été pris en considération. Un résumé de cette étude est présenté dans le paragraphe suivant.

Dans notre étude de simulation, nous évaluons quatre protocoles MAC, à savoir 802.11, CSMA, MACA et MACA-BI. Les trois premiers sont déjà implémentés dans GloMoSim. Quant au quatrième nous l'avons ajouté.

IV.2 Travaux existants

Dans [Bar02] C.L. Barret et al. ont effectué une analyse comparative de trois protocoles de contrôle d'accès au canal, à savoir CSMA, 802.11 et MACA. Ils ont considéré dans leur étude un réseau ad hoc statique[27]. Le but principal de leur expérience étant d'étudier comment les critères de la topologie du réseau, du taux d'injection des paquets dans le réseau[28], et de la location spéciale des connections (l'emplacement des pairs source - destinataire) affectent les performances des protocoles.

Les performances des protocoles ont été mesurées selon quatre paramètres, à savoir le nombre de paquets reçus, la latence moyenne de chaque paquet, l'assignation équitable des ressources (l'équité), et le débit. Pour cela, les auteurs ont considéré trois scénarios, chacun étant composé de trois sous-scénarios. Chaque scénario est conçu pour tester ou vérifier certaines hypothèses.

Le premier scénario a été crée pour tester l'effet du problème des nœuds cachés sur les performances des protocoles. Les auteurs ont conclu que la présence des nœuds

[27] Dans un réseau statique les nœuds sont immobiles.
[28] Le taux d'injection est la charge du réseau. Plus l'intervalle d'injection des paquets est petit, plus la charge du réseau est élevée

cachés conduit CSMA à assigner les ressources (canal) de manière inéquitable aux différentes connections. 802.11 évite ce problème en utilisant le mécanisme RTS/CTS. CSMA fonctionne très bien en terme de latence (il a le plus faible temps de latence). MACA a la plus haute latence, ainsi qu'une assignation inéquitable des ressources. 802.11 réalise un temps de latence mauvais par rapport à CSMA, mais meilleur que MACA. L'avantage du 802.11 est qu'il permet un accès équitable au medium, et un meilleur débit.

Le deuxième scénario a été conçu dans le but de tester l'effet de la connectivité du réseau sur les protocoles. Les auteurs ont remarqué que le fonctionnement des protocoles échoue clairement dans les situations extrêmes (faible et forte connectivité).

Le troisième scénario est conçu pour étudier l'effet de la taille et la largeur du réseau sur les performances des protocoles. Pour cela, les auteurs ont utilisé trois topologies. La première est une grille carrée 7x7 où la largeur du réseau n'est pas très importante, ce qui implique de courts chemins entre les noeuds. La deuxième et la troisième topologie engendrent une largeur plus importante du réseau et de longs chemins (grille 3x15). Les auteurs ont remarqué de faibles performances pour les protocoles dans les deux dernières topologies comparées à la première, la raison étant la large connectivité du réseau (sparse connectivity) et les longs chemins entre les nœuds. Les auteurs ont conclu que CSMA est le meilleur protocole en terme de latence. 802.11 fonctionne mieux en terme de *paquets reçus* mais ses performances se dégradent quand le taux d'injection des paquets devient extrêmement élevé. CSMA par contre, a montré de bonnes performances avec l'augmentation du taux d'injection.

A travers leurs expériences, les auteurs ont abouti aux conclusions générales suivantes :

- Aucun protocole ne domine les autres à travers les divers paramètres de performance.
- MACA a été largement dominé que ce soit par CSMA/CA ou par 802.11. Les auteurs ont conclu que CSMA/CA est certainement le meilleur protocole pour un réseau légèrement chargé (en terme de nombre de connections).

Cependant, dans cette étude, les auteurs n'ont pas pris en considération l'effet de la mobilité, qui constitue une propriété principale d'un réseau ad hoc. Ils n'ont pas étudiés aussi l'effet de l'augmentation du nombre de noeuds sur les performances des protocoles (la scalabilité des protocoles). En plus, les auteurs n'ont pas mesuré les performances des protocoles selon plusieurs critères importants, notamment la consommation d'énergie, et le nombre de collisions.

IV.3 Environnement de simulation

Notre simulation a été réalisée en utilisant le simulateur GloMoSim que nous avons décrit dans le chapitre précèdent, et que nous avons étendu afin d'effectuer les mesures nécessaires.

Nous décrivons dans ce qui suit, les paramètres utilisés durant toute la simulation.

IV.3.1 Paramètres généraux

Nous avons simulé un réseau dont les noeuds mobiles sont placés aléatoirement dans un terrain d'une longueur de 2000 mètres et d'une largeur de 800 mètres. Le domaine de puissance de chaque nœud est 200 mètres et la bande passante est 2Mb/secondes. Le temps de simulation est fixé à 15 minutes, le nombre de nœuds a été fixé a 50 durant les deux premières étapes, et il a été fait varié entre 10 et 70 dans la troisième étape.

IV.3.2 Modèle de mobilité

Parmi les différents modèles de mobilité implémentés dans GloMoSim, et que nous avons présentés dans le chapitre précèdent, nous avons choisi le modèle WAYPOINT, nous considérons le déplacement des nœuds dans ce modèle plus proche au mouvement réel des nœuds, comparé aux autres modèles.

IV.3.3 Modèle de propagation

Nous avons choisi le modèle de propagation Free Space qui suppose que le signal émis se propage de l'émetteur vers le récepteur sans qu'il n'y ait des obstacles entre eux, la seule influence sur le signal transmis est donc la distance. Nous avons choisi ce modèle parce qu'il élimine l'effet des obstacles, ce qui exclut toute influence possible sur les résultats de la simulation.

IV.3.4 Protocole de routage

Au niveau routage, nous avons choisi le protocole AODV (*Ad hoc On-demand Distance Vector*) [Per99] [Per02]. C'est l'un des plus connu des algorithmes réactifs. Son principe de fonctionnement est le suivant :

Quand un nœud S a besoin d'une route à une certaine destination D, il diffuse un message de demande de route à ses voisins, où il mis le dernier numéro de séquence pour cette destination. La demande de route est inondée dans le réseau jusqu'à ce qu'elle atteigne un nœud qui a une route à la destination. Chaque nœud qui expédie la demande de route crée une route inversée au nœud S.

Quand la demande de route atteint un nœud qui a une route à D, ce nœud produit une réponse de route qui contient le nombre de sauts nécessaires pour atteindre D et le numéro de séquence pour D le plus récemment vu par le nœud produisant la réponse.

Chaque nœud qui participe à l'expédition de cette réponse vers le créateur de la demande de route (le nœud S), crée une route vers D. L'état créé dans chaque nœud se rappelle seulement du prochain saut et non pas de la route entière.

IV.3.5 le trafic du réseau

Nous avons utilisé l'application CBR (Constant Bit Rate). Nous avons sélectionné des sources et destinations éloignées pour assurer que la propriété essentielle d'un réseau ad hoc, à savoir "le multi-sauts", est présente dans le réseau. Les pairs source-destination sont spécifiés dans le fichier cbr.config.

La taille d'un paquet de données est de 1 Ko. Le taux de transfert de paquet de données est de 1 paquet/seconde.

IV.4 Démarche générale pour l'ajout d'un nouveau protocole

La structure de GloMoSim, vue précédemment, permet l'intégration des nouvelles couches, ainsi que des nouveaux modèles et protocoles.

Pour développer des protocoles dans GloMoSim, il suffit de modifier son code source. Afin de respecter la structure de GloMoSim, il est recommandé de mettre le code source du protocole à développer dans un nouveau fichier, et mettre ce dernier dans le sous répertoire adéquat (radio, mac, network,...). Ensuite, on définit des nouveaux paramètres associés à ce protocole dans le fichier d'entrée qui seront lu à

partir de GloMoSim en utilisant les fonctions API : Glomo_Read {Int|Double|String|Time} (), et on insère des nouvelles commandes de compilation dans le fichier Makent.bat .

Trois principales fonctions sont nécessaires lors de l'ajout d'un nouveau protocole :

- Une fonction d'initialisation: Alloue et initialise le modèle de donnée spécifique au protocole.
- Une fonction de terminaison: produit les statistiques pour le modèle.
- Une fonction de traitement des évènements simulés: Exécute les actions de simulations suivant les messages qui arrivent à cette fonction.

IV.5 Implémentation du protocole MACA-BI

Dans la version actuelle de GloMoSim (V2.03), MACA-BI n'est pas implémenté, nous l'avons ajouté afin de pouvoir le simuler. Nous nous sommes basés dans notre implémentation sur l'article original qui définit le protocole [Tal97]. Comme vu précédemment, MACA-BI est basé sur l'approche par invitation, c'est-à-dire qu'un nœud ne peut émettre ses données avant de recevoir une invitation (RTR). Pour améliorer les performances du protocole, les auteurs ont précisé qu'un nœud peut envoyer une demande d'émission RTS s'il ne reçoit pas d'invitations pendant un certain délai. Cependant, ils n'ont pas spécifié la valeur de ce dernier. Nous avons donc choisi le délai de manière à apporter les meilleures performances.

Le fonctionnement de MACA-BI tourne autour de quatre fonctions principales:

- Une fonction d'initialisation : cette fonction est appelée pour initialiser, pour chaque nœud, la structure de données du protocole. Cette dernière contient, des informations sur l'état de la couche MAC (Passive, Backoff, En émission d'un RTS, ...), le timer, la liste de prédiction, les statistiques et la priorité actuelle de la couche MAC[29].
- Une fonction de traitement des paquets reçus : cette fonction est appelée dès qu'un paquet est reçu de la couche Radio ou de la couche réseaux. Suivant le type de paquets et l'état de la couche MAC, les instructions adéquates sont exécutées.

[29] C'est la priorité du dernier paquet traité (Contrôle, Temps Réel ou non Temps réel).

- Une fonction de traitement des événements : cette fonction est chargée de gérer les événements déclenchés. Un événement est déclenché si son heure d'activation coïncide avec celle du système de simulation. Les instructions relatives à l'événement déclenché sont donc exécutées.
- Une fonction de finalisation : elle est appelée en fin de simulation pour collecter l'ensemble des résultats concernant le protocole.

IV.6 Les paramètres de comparaison

Nous avons choisi trois paramètres de comparaison, que nous avons jugés importants pour réaliser notre étude. Ces paramètres sont :

IV-6-1 La mobilité

La mobilité est un paramètre important pour l'évaluation des réseaux ad hoc, dans la mesure où elle cause le changement de la topologie. Il existe plusieurs définitions de ce paramètre, tel que la définition qui le représente par la vitesse des nœuds, ou le pause time dans le modèle waypoint. Ces définitions n'expriment pas vraiment le changement de la topologie, car il existe des cas où les nœuds se déplacent avec une grande vitesse ou avec un pause time très faible dans une même direction sans qu'il n'y ait des changements dans la topologie du réseau, et d'autres cas où les nœuds se déplacent avec une vitesse faible ou un grand pause time mais en s'éloignant les uns des autres causant des changements de liens importants.

Soit un exemple simple illustrant l'invalidité de ces définitions :

Soit un réseau contenant 2 nœuds liés qui se déplacent avec une grande vitesse et/ou un petit pause time mais dans la même direction, ces 2 nœuds restent liés. Mais si les nœuds se déplacent avec une vitesse faible et/ou un grand pause time dans des directions opposées l'un par rapport à l'autre, après un certain temps le lien sera défaillant. Pour cela, on a opté pour une autre définition de la mobilité présentée dans [LAR 98].

Cette définition est basée sur le mouvement relatif entre les nœuds, ce qui exprime bien le changement de la topologie du réseau.

La mobilité est une fonction de la vitesse et du modèle de mouvement. Elle est représentée par un nouveau paramètre (facteur de mobilité), qui est calculé durant la simulation avec un certain taux Δt. La formule du facteur de mobilité Mob est :

$$Mob = \sum_{i=0}^{n-1} Mi / n$$

$$M_x = \sum_{t=0}^{T} | A_x(t) - A_x(t + \Delta t) | / T$$

$$A_x(t) = \sum_{i=0}^{n-1} dist(n_x, n_i) / n-1$$

ou :

$dist(n_x,n_y)$: Distance entre le nœud x et le nœud y.

n : le nombre de nœuds.

$A_x(t)$: la distance moyenne entre le nœud x et tous les autres nœuds à l'instant t.

M_x : la mobilité relative moyenne du nœud x par rapport à tous les autres nœuds durant le temps de simulation.

T : le temps de simulation.

Δt : l'intervalle de temps utilisé dans le calcul.

On calcule $A_x(t)$ après chaque Δt c.-à-d. pour t=0, t= Δt, t=2 Δt,……,t=T.

Pour notre simulation nous avons utilisé l'implémentation [Dja03] qui ajoute le calcul du facteur de mobilité dans GloMoSim et un autre paramètre dans le fichier d'entrée qui est INTERVAL TIME afin de donner une valeur à Δt. Si ce paramètre n'est pas fixé, INTERVAL TIME prend la valeur par défaut qui est égale a T /1000.

Durant toute la simulation, on a choisi INTERVAL TIME = 0.1 seconde.

IV.6.2 La charge du réseau

C'est le nombre total des paquets de données envoyés par la couche application. Nous avons choisi l'application CBR implémentée dans GloMoSim. La syntaxe pour définir une application CBR est de la forme suivante :

<nœud source, nœud destination, taille du paquet, taux d'envoi de paquet, le temps de début de session, le temps de la fin de session>

Ou : nœud source, nœud destination sont respectivement l'identité des nœuds source et destination de la session CBR, et le taux d'envoi de paquet est le temps séparant les envois de deux paquets successifs.

Ce paramètre (la charge) est caractérisé par trois autres paramètres : La taille du paquet, le taux d'envoi de paquet et le nombre de source CBR. Pour le faire varier, nous avons préféré la variation du nombre de source et nous avons fixé la taille à 1KO, et le taux d'envoi de paquet à 1 seconde.

Plus la charge du réseau augmente plus la bande passante est occupée, le risque de collision augmente et donc d'une manière ou d'une autre les performances se dégradent. Il est important d'observer la manière dont s'effectue la dégradation dans chaque protocole.

IV.6.3 La scalabilité

Ce paramètre consiste à étudier les effets de la croissance du nombre de nœud sur les performances du réseau. Pour ceci, nous avons fait varier le nombre de nœud n sur l'intervalle [10, 70] où $n = n_0 + 10$ et $n_0 = 10$, avec dans un premier lieu une mobilité nulle (Vmax = 0) et en second une mobilité moyenne (Vmax = 2.5), tout en préservant une même densité de dispersion de ces nœuds.

Dans la densité choisie chaque nœud à quatre voisins, dont la distance le séparant des autres est 200 m, réalisant une surface de 400 m * 400 m pour 5 nœuds (le nœud en question et ses voisins).

Ainsi il est possible de déterminer la surface pour un nombre de nœud donné.

VI.7 Les métriques de performances mesurées

Pour comparer et évaluer les performances des protocoles MAC, nous avons choisi quatre métriques de performance que nous avons considéré pertinents, et qui vont nous aider à étudier le comportement des protocoles sous différentes conditions, ainsi que leur influence sur le réseau.

Pour un scénario, nous avons utilisé trois exécutions avec trois Seed différents, les données finales représentent la moyenne des trois exécutions. Ce qui nous permettra d'avoir des résultats plus exacts.

Les métriques mesurées sont :

IV.7.1 Energie consommée

Les nœuds mobiles sans alimentés par des batteries de capacités limitées. Nous avons donc jugé intéressant d'évaluer les performances des protocoles en terme de consommation d'énergie. Un protocole MAC est autant meilleur s'il cause moins de consommation d'énergie par rapport aux autres dans les mêmes conditions.

Le calcul de l'énergie consommée est déjà implémenté dans GloMoSim en utilisant le modèle de radio NCR wavelan, la formule de calcul de l'énergie consommé dans la batterie du nœud i (Power_consumed $_i$) est :

$$\text{Power\_consumed}_i = \sum_{\text{reception paquet}} \text{trans\_delay} * (\text{radio\_receive\_rate} - \text{radio\_sleep\_rate}) + \sum_{\text{transmission paquet}} \text{trans\_delay} * (\text{radio\_transmit\_rate} - \text{radio\_sleep\_rate}) + \text{radio\_sleep\_rate} * (\text{radio\_turnOffTime} - \text{radio\_turnOnTime})$$

Tel que :

trans_delay = PacketSize / bandwidth + synchronizationTime

PacketSize : c'est la taille d'un paquet

bandwidth : la taille de la bande passante

synchronizationTime = 192 micro seconde

radio_transmit_rate = 3 / second

radio_receive_rate = 1.48 / second

radio_sleep_rate = 0.18 / second

radio_turnOnTime : le temps de la mise en marche de l'interface de communication (début de la simulation)

radio_turnOffTime : le temps d'arrêt de l'interface de communication (fin de simulation).

Dans notre étude, nous ne nous intéressons pas à l'énergie consommée pour chaque nœud mais à l'énergie moyenne consommée dans le réseau, nous utilisant donc la métrique suivante :

$$\text{average\_power} = \sum_{i=0}^{n-1} \text{power\_consumed}_i / n$$

Où power_consumed$_i$ est l'énergie consommé par le noeud i .

Il est a noter qu'une station consomme plus d'énergie en mode d'émission qu'en mode réception, et consomme moins d'énergie en mode repos (idle), et encore moins en mode sleeping. Le tableau IV-1 présente un exemple de la consommation d'énergie d'une carte réseau Cabletron 802.11 dans les différents modes :

Transmission	Réception	Idle	Sleeping
1400 mW	1000 mW	830 mW	130 mW

Tableau IV-1 [Pet02]: consommation d'énergie dans les différents modes

IV.7.2 Les collisions

Une collision se produit lorsque un nœud reçoit un paquet alors qu'il est en mode réception (il est entrain de recevoir un autre paquet). Le calcul du nombre de collisions produites au niveau de chaque nœud est implémenté dans GloMoSim. Pour notre étude, nous nous intéressons aux nombres de collisions moyennes dans le réseau pour pouvoir établir la comparaison. Nous avons donc ajouté cette métrique.

IV.7.3 La fraction de réception de données

C'est la fraction entre le nombre de paquets envoyés par la couche application et le nombre de paquets reçu par leurs destinations finales. C'est un paramètre très important qui permet d'évaluer la fiabilité du protocole. Plus la valeur de ce paramètre est proche du 1 plus le nombre de paquets perdus est réduit, et plus le protocole est fiable. Par contre si la valeur s'éloigne de 1 le nombre de paquets perdu est important, ce qui n'est pas acceptable.

IV.7.4 Le délai d'attente au niveau MAC

C'est le délai d'attente moyen d'un paquet de donnée au niveau de la couche MAC, c'est-à-dire depuis l'arrivée du paquet de la couche réseaux jusqu'à ce qu'il soit envoyé à la couche Radio. Si on note cette métrique par Average_delai, on aura:

$$Average\_delai = \sum_{i=0}^{n-1} delai_i / n$$

n : nombre de nœuds.

$delai_i$: C'est le délai moyen des paquets de donnée dans le buffer du nœud i.

$$delai_i = \sum_j delai(P_j) / Nbr_i$$

Nbr_i : Nombre de paquets de données envoyés a la couche Radio dans le nœud i.

$delai(P_j)$: Délai d'attente d'un paquet P_j. $delai(P_j) = Te - Ta$

Te : Temps d'envoi du paquet à la couche radio.

Ta : Temps d'arrivée du paquet à la couche MAC.

Nous avons ajouté cette métrique dans GloMoSim, elle est très importante pour étudier la qualité de service du protocole. Un protocole est autant meilleur s'il engendre un délai d'attente minimum.

IV.8 La démarche de simulation

Notre simulation est divisée en trois étapes :

1- La mobilité : dans cette étapes nous faisons varier le facteur de mobilité, et nous fixons tous les autre paramètres. Cela nous permet d'observer l'effet de la mobilité sur les métriques mesurées.

2- La charge : dans cette étape nous faisons varié la charge, commençant par un réseau légèrement chargé pour arriver à un réseau extrêmement chargé. Ce qui nous permettra d'observer le comportement des protocoles dans les différents niveaux de la charge. Nous avons utilisés deux types de mobilité: nulle et moyenne.

3- La scalabilité : dans cette étape, nous étudions l'effet de l'augmentation du nombre de nœuds sur les performance des protocoles. Pour cela nous avons fait varier le nombre de nœuds de 10 à 70 nœuds, sous deux type de mobilité : nulle est moyenne, tout en ayant fixé les autres paramètres.

IV.9 Les résultats de simulation

IV-9-1 La mobilité

Dans cette expérience, nous avons fixé la charge du réseau ainsi que la connectivité à des valeurs moyennes, telles que:

La charge : 12 sources CBR

Le domaine de puissance : 200m

Nous avons fait varier la vitesse relative des nœuds de la valeur zéro où les nœuds sont immobiles, à la valeur 5 m/s qui constitue une vitesse importante. Les résultats obtenus sont :

IV.9.1.1 L'énergie consommée

On remarque sur la figureIV.1 que les consommations d'énergie des protocoles 802.11 et CSMA sont stables, et ne sont pas influencée par l'augmentation de la mobilité. La consommation de CSMA est légèrement meilleure par rapport à 802.11, la raison étant que CSMA, contrairement aux autres protocoles, n'utilise ni l'échange de paquets de contrôle avant l'émission d'une donnée (RTS/CTS), ni des acquittements (ACK). Donc, en terme d'émission de paquets, 802.11 dépasse CSMA et consomme donc plus d'énergie.

MACA et MACA-BI sont clairement influencées par la mobilité, leurs consommation sont très élevés comparée à CSMA et 802.11, surtout quand la mobilité augmente. La cause est que MACA et MACA-BI utilisent l'échange de paquets de contrôle avant chaque transmission de donnée sans écouter le canal, c'est-à-dire que chaque station, dès qu'elle a un paquet de donnée à émettre, elle envoie un RTS sans tenir compte de l'état du canal. Cela engendre des collisions fréquentes surtout avec l'augmentation de la mobilité[30], ce résultat va se confirmer dans la partie d'analyse des collisions (voir IV.9.1.2). Des retransmissions sont donc nécessaires, ce qui augmente la consommation d'énergie.

On remarque aussi que MACA-BI consomme plus d'énergie comparée à MACA. La cause est la suivante : dans MACA-BI chaque nœud possède une liste de ses voisins (les nœuds qu'il peut inviter). Quand la mobilité augmente, les mouvements des noeuds augmentent, et donc la probabilité que les voisins de chaque nœud se déplacent augmente aussi. Le nœud enverra des RTR à des nœuds qui se trouvent hors de sa portée. Comme il ne reçoit pas de réponse, des retransmissions successives vont avoir lieu, ce qui augmente la consommation d'énergie.

[30] Le déplacement des noeuds conduit à des défaillances fréquentes des liens.

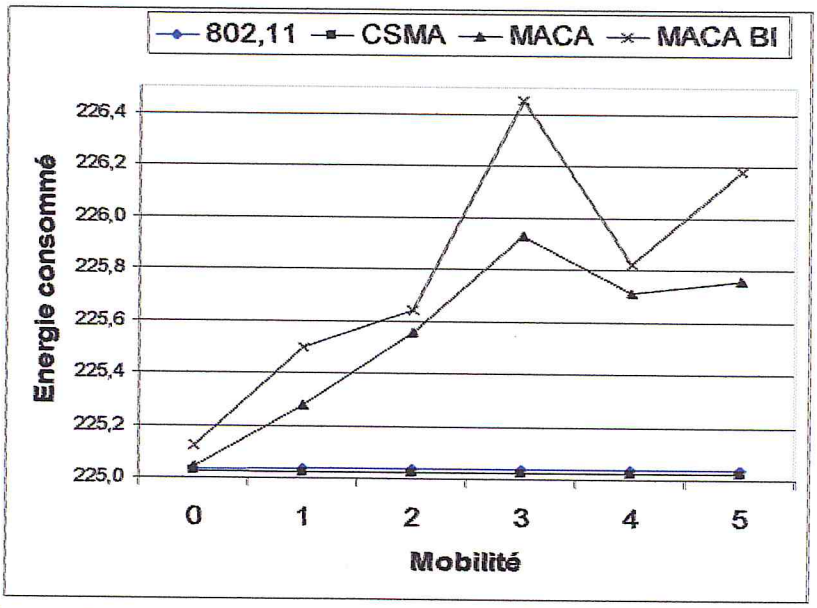


Figure IV.1: L'énergie consommé en fonction de la mobilité

IV.9.1.2 Les Collisions

En terme de collision, on remarque que 802.11 et CSMA causent moins de collisions et ne sont pas influencés par la mobilité (figureIV.2). 802.11 est meilleur que CSMA, la cause étant que 802.11 utilise, en plus de l'écoute du canal, un mécanisme de Backoff avant chaque transmission d'un paquet de donnée, ce qui minimise les risques de collisions. Cette dernière fonctionnalité de 802.11 lui rapporte un gain en terme de collisions, mais une perte en terme de délai d'attente. La partie suivante (IV.9.1.4) va le confirmer. CSMA fait preuve de meilleures performances en terme de collisions par rapport à MACA et MACA-BI, parce que CSMA n'introduit aucun paquet de contrôle dans le réseau. L'overhead du réseau est par conséquent faible, ce qui décroît les possibilités de collisions. La non-utilisation du mécanisme de réservation du canal (RTS/CTS) et d'acquittement (ACK) permettent à CSMA de gagner en terme de collisions, mais pas en terme de fraction de réception (voir IV.9.1.3).

On constate aussi que MACA et MACA-BI sont influencées par la mobilité. En plus, le nombre de collisions de ces deux protocoles est assez élevé par rapport aux autres protocoles. Cela est dû à l'utilisation d'échange de paquets de contrôle sans

écouter le canal, ce qui augmente considérablement les possibilités de collisions directes[31] et indirectes[32], surtout dans les fortes mobilités.

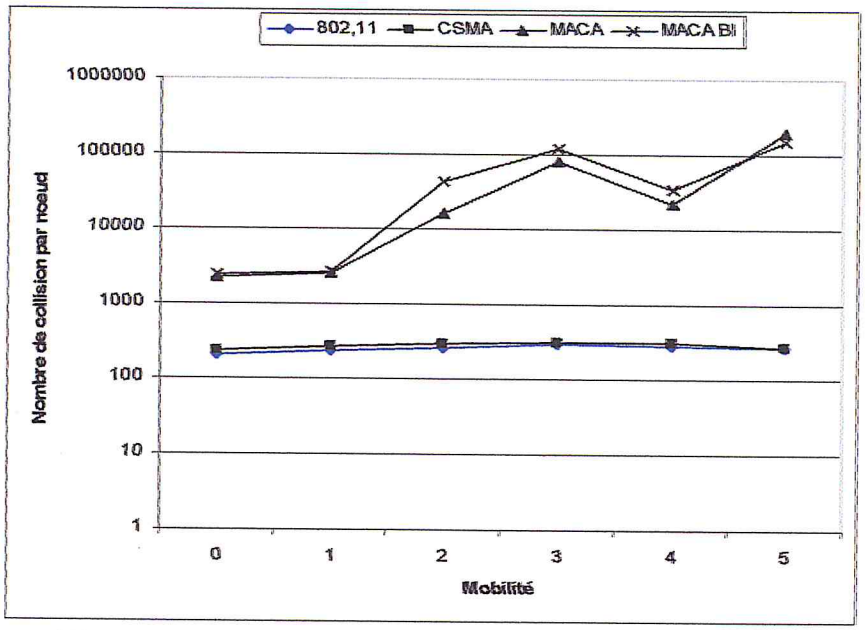


Figure IV.2 : Les collisions en fonction de la mobilité

IV.9.1.3 La fraction de réception des données

Nous remarquons sur la figureIV.3 que 802.11 n'est pas influencé par la mobilité, et qu'il fait preuve de meilleures performances comparé aux autres protocoles. Il réalise une fraction de réception proche du 1, quelle que soit la valeur de la mobilité. Les meilleures performances de 802.11 s'expliquent par le fait que ce dernier utilise une stratégie de Backoff plus un mécanisme de réservation du canal RTS/CTS avant chaque transmission d'un paquet de donnée, ajoutant à cela l'écoute du support et surtout l'utilisation des ACK. Tout cela permet de minimiser les conflits entre paquets et d'assurer la fiabilité.

Les autres protocoles sont largement influencés par la mobilité. Quand la mobilité est nulle, on remarque que MACA et MACA-BI réalisent une faible perte, environ 17% de l'ensemble des paquets envoyés, alors que les pertes dans CSMA sont élevées (40%). Ce résultat s'explique par le fait que CSMA procède directement à l'envoi des paquets de données dés que le canal devient libre. Cela engendre des collisions des paquets de données à cause du problème du terminal

[31] À cause de la non-utilisation de l'écoute du canal.
[32] A cause du problème des nœuds cachés qui devient plus fréquent quand la mobilité augmente.

caché[33] et donc des pertes de données. Avec l'augmentation de la mobilité dans le réseau, nous constatons une dégradation importante de CSMA, MACA et MACA-BI (une perte d'environs 75% pour une forte mobilité). Ceci est dû principalement à l'absence d'acquittement.

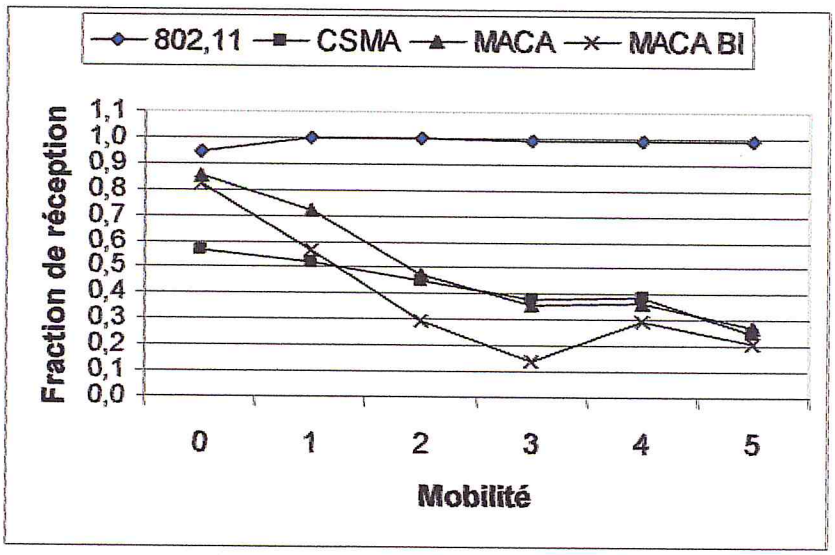


Figure IV.3: La fraction de réception en fonction de la mobilité

IV.9.1.4 Le délai d'attente au niveau MAC

802.11 et CSMA ne sont pas influencés par l'augmentation de la mobilité (figure IV.4). En plus, nous constatons que CSMA réalise un délai légèrement inférieur, comparé à 802.11. La cause est que CSMA n'utilise ni mécanisme RTS/CTS ni mécanisme de Backoff avant chaque transmission, ce qui minimise le délai d'attente des paquets de données. Par contre, MACA et MACA-BI sont influencés par la mobilité. Quand la mobilité des nœuds est nulle, MACA réalise un meilleur délai (égal a celui de CSMA). Avec l'augmentation de la mobilité, le délai de MACA et MACA-BI augmente considérablement. Cela est dû à l'échec du mécanisme RTS/CTS de MACA a cause du mouvement des nœuds, et aux collisions entre RTS (ou RTR pour MACA-BI) à cause de la non-utilisation de l'écoute du support par ces deux derniers protocoles. Tout cela engendre aux paquets plus d'attente avant l'envoi.

[33] CSMA n'utilise aucun mécanisme pour pallier au problème du nœud caché.

Une autre cause peut expliquer les faibles performances de MACA-BI consiste en l'échec du mécanisme d'invitation à cause du mouvement aléatoire des nœuds[34].

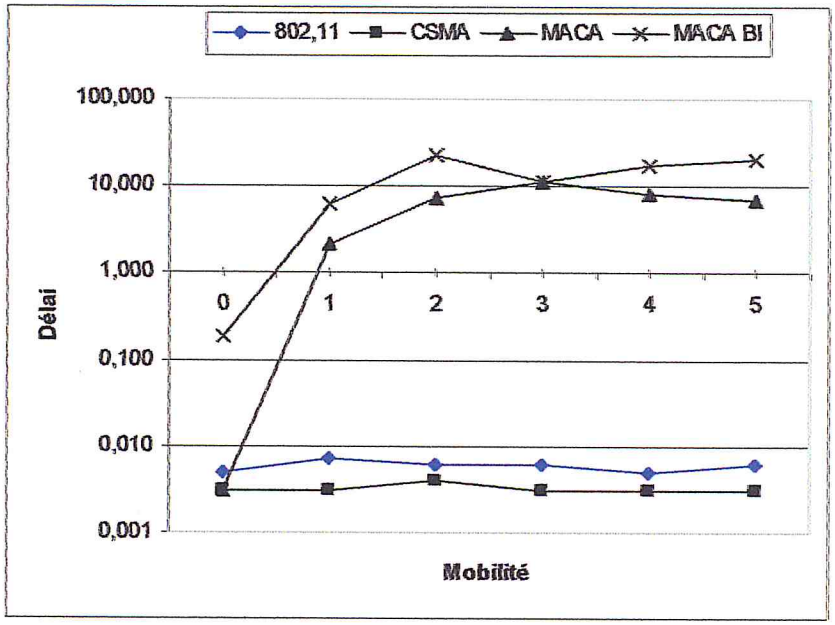


Figure IV.4 : Le délai d'attente en fonction de la mobilité

IV.9.2 La charge

Dans cette phase, nous avons fixé tous les paramètres et nous avons fait varier la charge du réseau. Après les simulations nous avons obtenu les résultats suivants :

IV.9.2.1 Energie consommé

Suivant les figureIV.5a et IV.5b, nous remarquons que CSMA et 802.11 sont légèrement influencés par la charge du réseau, quel que soit le type de mobilité.

Quand la mobilité est nulle (Figure IV.5a), MACA éprouve une consommation d'énergie proche de celle de CSMA et 802.11. Le protocole MACA-BI est tributaire de l'augmentation de la charge dans le réseau. Lorsque le trafic du réseau devient important (au delà de 1212Ko), la consommation d'énergie de MACA-BI augmente considérablement et d'une manière exponentiel. La raison est la suivante: quand le trafic du réseau augmentent, les collisions des paquets de données et des paquets de contrôle augmente, ce qui fait que les tables de prédictions au niveau de chaque nœuds ne seront pas mise a jour, des retransmission gratuits vont donc avoir lieu ce qui cause une consommation d'énergie importante.

[34] Quand la mobilité augmente, les voisins à inviter d'un nœud se déplace et donc sortent fréquemment de la portée du nœud inviteur.

Quand la mobilité est introduite dans le réseau (Figure IV.5b), la consommation d'énergie de MACA devient importante. Pour les mêmes raisons vues précédemment (voir IV.9.1.1), mais aucune influence par l'augmentation de la charge n'est observée.

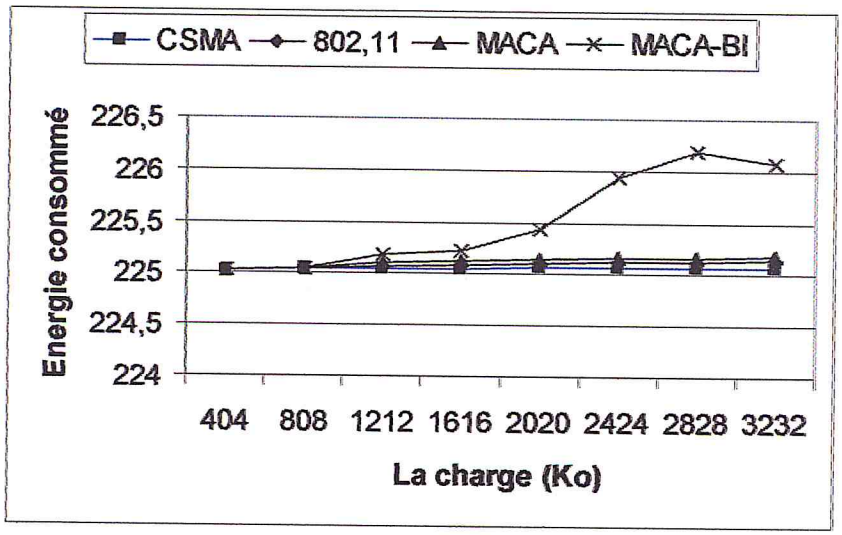


Figure IV.5a : L'énergie consommé en fonction de la charge – mobilité nulle

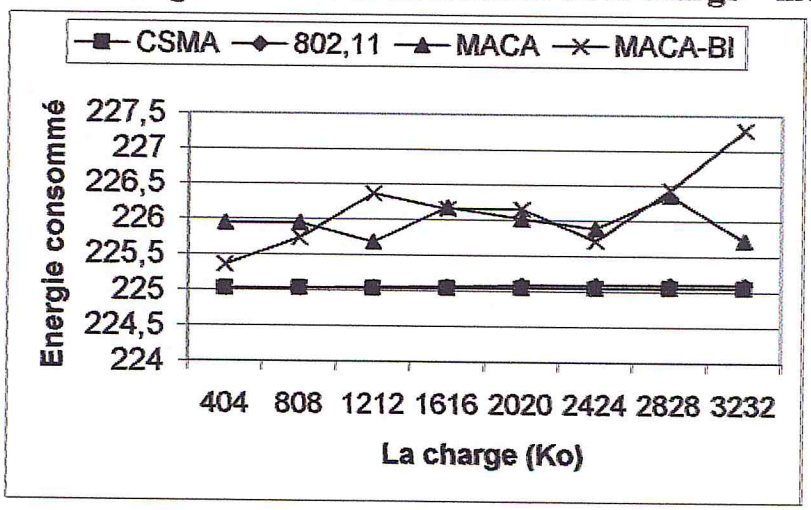


Figure IV.5b :L'énergie consommé en fonction de la charge – mobilité moyenne

IV.9.2.2 Les collisions

Tous les protocoles sont influencés par l'augmentation de la charge du réseau. Quel que soit le type de mobilité, nous remarquons que 802.11 et CSMA engendrent un nombre réduit de collisions. 802.11 produit moins de collisions par rapport à CSMA, pour les mêmes raisons qu'on a vues précédemment.

Quand la mobilité est nulle (figure IV.6a) et le réseau n'est pas chargé, 802.11, MACA et MACA-BI réalisent moins de collisions comparées à CSMA. La cause est que CSMA n'incorpore aucun mécanisme d'évitement de collisions (collisions-avoidance). Quand la charge augmente, le nombre de collisions de MACA et MACA-BI devient plus importantes comparé à CSMA et 802.11, la cause étant que MACA et MACA-BI utilisent un échange de paquets de contrôle sans écoute du canal. Ajoutons a cela le fait qu'ils n'utilisent pas de Backoff après chaque transmission, ce qui augmente la possibilité de collisions.

En plus, quand le trafic du réseau est moyen MACA-BI réalise moins de collisions par rapport à MACA, la raison étant que MACA-BI introduit moins de paquets de contrôle dans le réseau par rapport à MACA, ce qui minimise les chances de collisions, mais quand le trafic devient élevé, le mécanisme d'invitation de MACA-BI échoue, par conséquent le nombre de collisions augmente.

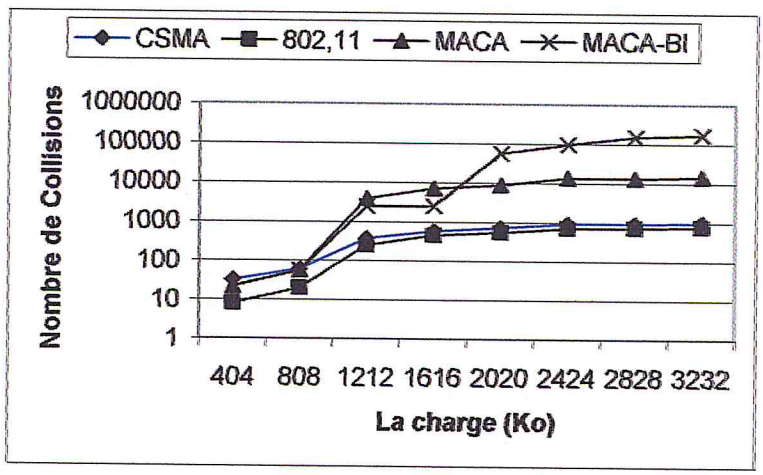


Figure IV.6a : Les collisions en fonction de la charge– Mobilité nulle

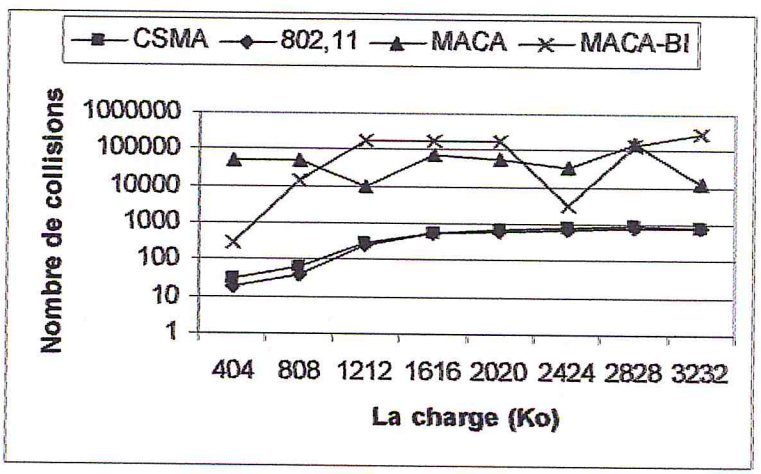


Figure IV.6b : Les collisions en fonction de la charge – Mobilité moyenne

IV.9.2.3 Fraction de réception des données

Dans les figures ci-dessous (IV.7a et IV.7b), nous remarquons que tous les protocoles sont tributaires de la charge du réseau. Quand la charge est faible, la fraction de réception des protocoles est parfaite (égal à 1). Quand elle augmente, les performances de MACA et MACA-BI se dégradent doucement jusqu'à atteindre environs 0,3 pour MACA et 0,2 pour MACA-BI dans un réseau extrêmement chargé, cela est dû aux mêmes raison vue précédemment (voir IV.9.1.3). On remarque aussi une dégradation importante pour CSMA, la cause étant la non-utilisation d'un mécanisme de réservation du canal, comme nous avons vue précédemment. Quand la mobilité est nulle, la fraction de réception de 802.11 reste la meilleure. Elle ne va pas en deçà de 0,8, quelle que soit la charge. Ceci s'explique par les mêmes causes données dans IV.9.2.2.

Quand la mobilité est introduite dans le réseau (figureIV.7b), nous remarquons une dégradation pour tous les protocoles, à cause de l'ajout du facteur de défaillances des liens causés par le déplacement fréquent des nœuds.

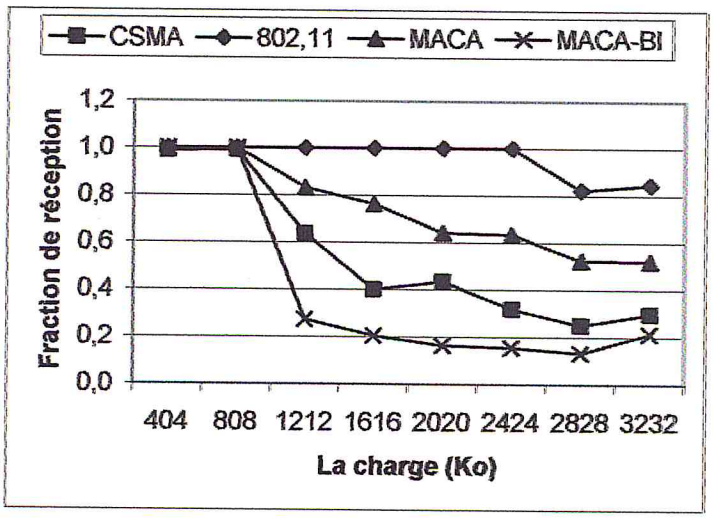


Figure IV.7a : La fraction de réception en fonction de la charge – Mobilité nulle

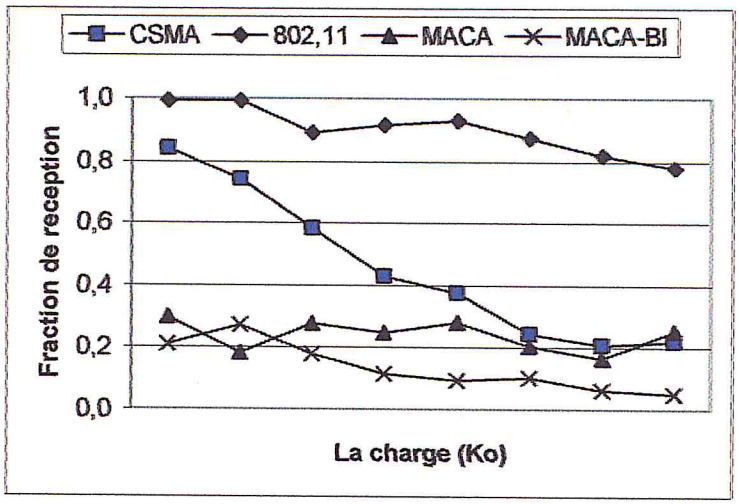


Figure IV.7b : La fraction de réception en fonction de la charge – Mobilité moyenne

IV.9.2.4 Le délai d'attente au niveau MAC

On remarque sur la figure IV.8a que le délai d'attente de MACA est meilleur surtout quand le réseau n'est pas chargé (un délai moins de 0.001 Seconde). Cela s'explique par le succès du mécanisme RTS/CTS quand la charge est faible. Avec l'augmentation de la charge, les performances de MACA en terme de délai se dégradent. Cela est dû à la non-utilisation de l'écoute du canal ce qui conduit à des collisions fréquentes des RTS. Un temps Backoff est donc nécessaire avant la retransmission des RTS, ce qui augmente le délai d'attente des paquets. CSMA est indépendant de la charge du réseau, parce qu'il n'utilise pas d'échange RTS/CTS avant la transmission d'un paquet de donnée. Le délai d'attente est donc faible. Le Backoff utilisé avant chaque transmission, en plus des temporisateurs IFS, conduit 802.11 à réaliser un délai d'attente plus important que CSMA, MACA et MACABI (pour une faible charge). Dans MACA-BI, quand le nœud a un paquets de donnée à émettre, il n'essaie pas d'envoyer une demande de transmission RTS, mais il attend une invitation du destinataire prévu, ce qui explique le délai important de MACA-BI, surtout quand le réseau est chargé, ce qui cause des collisions directes et indirectes entre les RTRs.

Quand les nœuds sont mobiles, nous remarquons une augmentation importante des délais de MACA et MACA-BI, la cause étant d'une part, l'échec du mécanisme RTS/CTS et du mécanisme d'invitation et d'autre part, les collisions fréquentes, ce qui nécessite des retransmissions des RTSs (ou RTRs), après un

temps Backoff, ce qui accroît le délai d'attente des paquets de données dans le buffer.

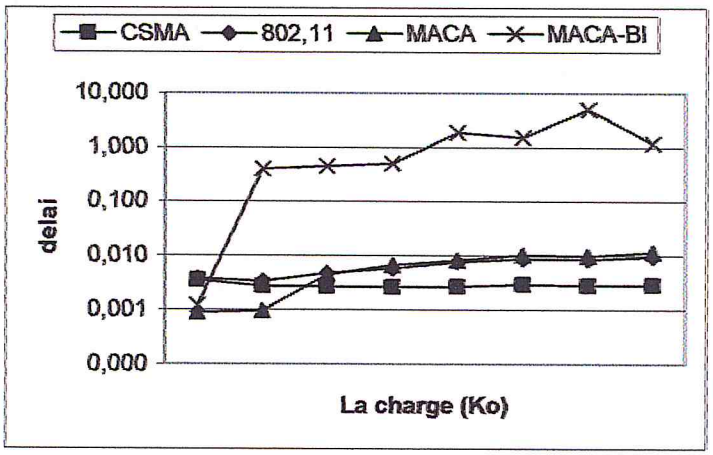


Figure IV.8a : Le délai d'attente en fonction de la charge – Mobilité nulle

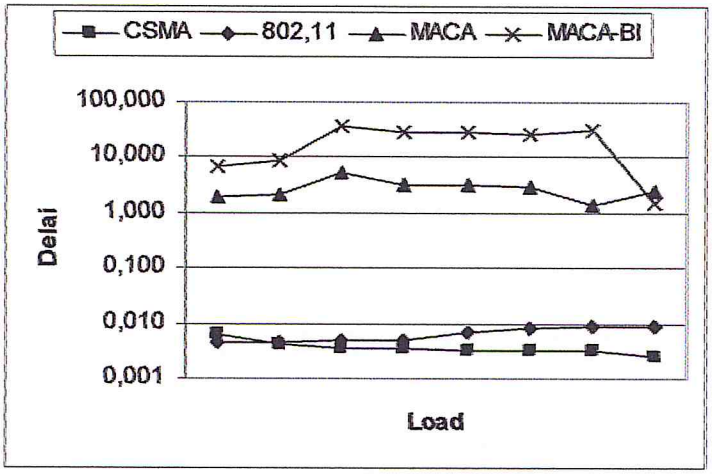


Figure IV.8b : Le délai d'attente en fonction de la charge – Mobilité moyenne

IV.9.3 La scalabilité

Dans cette étape nous avons fixé la charge et la connectivité du réseau à des valeurs moyennes, et nous avons fait varier le nombre de noeuds. Pour chaque valeur du nombre de noeuds (de 10 à 70), nous avons calculé la taille du terrain adéquat (voir IV.6.3), pour garder la même densité du réseau. Et ainsi nous éliminons l'effet du changement de la connectivité et nous étudierons uniquement l'effet de la scalabilité. Les résultats obtenus sont :

IV.9.3.1 Energie consommée

Quel que soit le type de mobilité, nous remarquons que tous les protocoles sont influencés par la scalabilité. La consommation d'énergie de 802.11 est meilleure que celle de CSMA, ce dernier est meilleur comparé à MACA et MACA-BI. Les meilleures performances de 802.11 s'expliquent par le fait qu'il utilise un mécanisme de Backoff avant chaque transmission, ce qui permet de réduire les collisions (la partie IV.9.3.2 le confirme). Les retransmissions sont donc minimes, d'où une faible consommation d'énergie. CSMA est meilleur que MACA et MACA-BI, la cause étant que ces derniers passent par un échange de paquets de contrôle avant chaque transmission d'un paquet de données sans écoute du canal, ce qui génère plus de collisions et donc plus de retransmissions.

Comme nous constatons dans les figures ci-dessous, aucun des protocoles n'est scalable. Plus le nombre de nœuds augmente, plus la longueur des chemins augmente. Etant donnée qu'un échange de paquets de contrôle et/ou une transmission de paquet de données sont nécessaires pour chaque saut, la consommation d'énergie augmente automatiquement avec l'augmentation du nombre de nœuds.

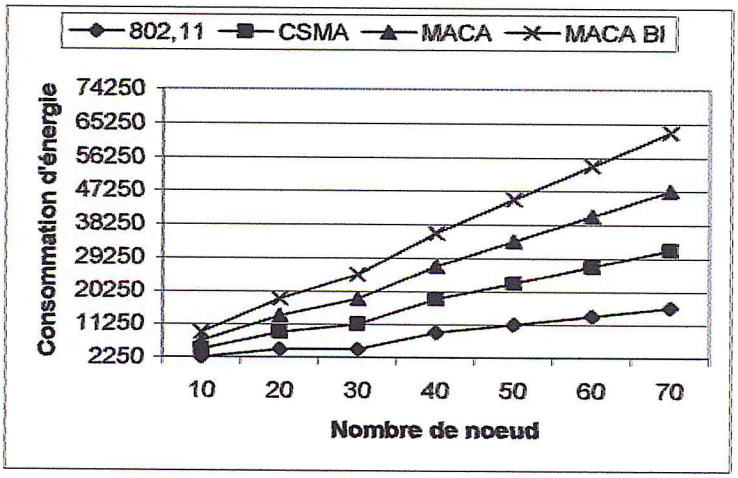


Figure IV.9a : L'énergie consommé en fonction de la scalabilité - Mobilité nulle

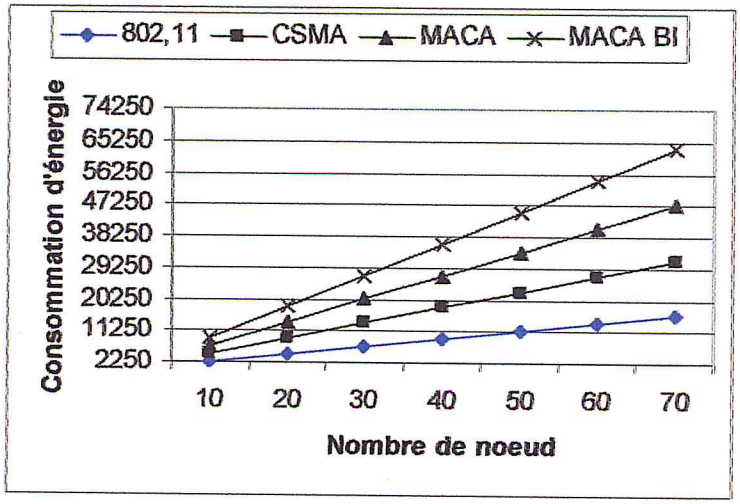


Figure IV.9b : L'énergie consommé en fonction de la scalabilité - Mobilité moyenne

IV.9.3.2 Les collisions

Quel que soit le type de mobilité, nous remarquons que 802.11 et CSMA sont légèrement influencés par l'augmentation du nombre de nœuds. 802.11 engendre moins de collisions par rapport a CSMA (la raison est la même que dans IV.9.2.2). Par contre, MACA et MACA-BI sont largement influencés par l'augmentation du nombre de nœuds. La raison étant l'augmentation de la longueur des routes chaque fois que le nombre de nœuds augmente, ce qui augmente les possibilités de collisions à cause de la non-utilisation de l'écoute du support d'une part, et de la non-utilisation du mécanisme de Backoff avant chaque nouvelle transmission d'autre part. Quand la mobilité est introduite dans le réseau (Figure IV.10b), nous remarquons une augmentation considérable dans le nombre de collisions, la raison étant la défaillance des liens qui devient de plus en plus fréquente à cause des mouvements des nœuds. Des retransmissions gratuites[35] d'RTS (ou RTR) auront donc lieu, ce qui augmente l'overhead du réseau et donc la possibilité de collisions. Le nombre de ces retransmissions augmente avec l'augmentation du nombre de nœuds.

[35] On parle de retransmissions gratuites, quand un nœud envoie des RTS ou RTR à un nœud hors de sa portée.

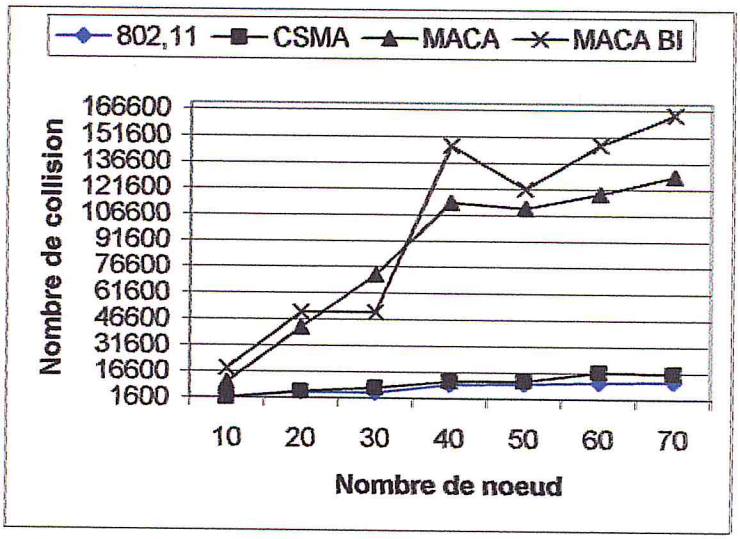


Figure IV.10a : Les collisions en fonction de la scalabilité - Mobilité nulle

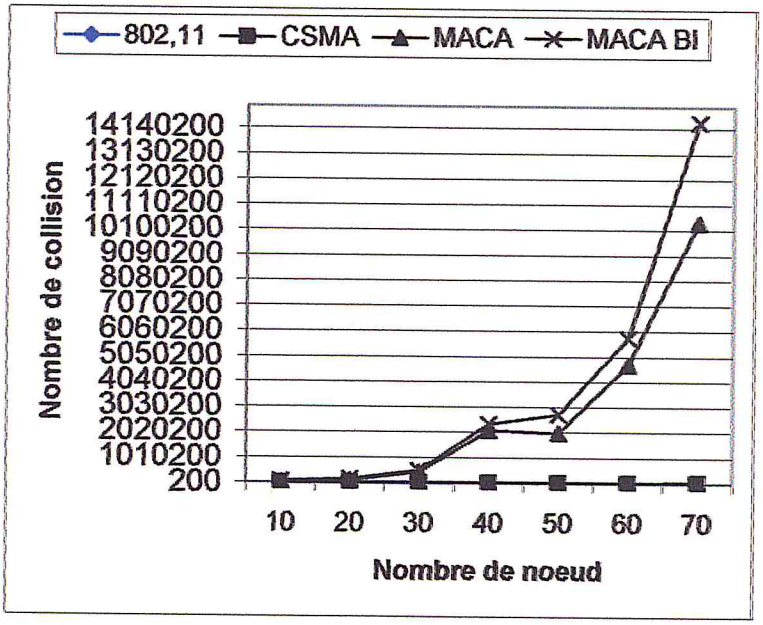


Figure IV.10b : Les collisions en fonction de la scalabilité - Mobilité moyenne

IV.9.3.3 La fraction de réception des données

Quand la mobilité est nulle (Figure IV.11a), on remarque que tous les protocoles sont légèrement influencés par l'augmentation du nombre de nœuds. La fraction de réception de 802.11 est la meilleure pour la même raison que dans IV.9.2.3. MACA et MACA-BI réalisent une fraction de réception acceptable (> 80%), meilleure à celle de CSMA, aussi pour les mêmes raisons citées dans IV.9.1.3.

Quand la mobilité est introduite (Figure IV.11b), la fraction de 802.11 reste la meilleure et n'est pas influencée par l'augmentation du nombre de nœuds. Cela

implique que l'augmentation du nombre de nœuds et donc des longueurs des chemins n'a pas d'influence sur la fraction de réception des données. Quelle que soit la longueur des chemins, les paquets de données ont plus de chance d'arriver a leurs destination (sans collisions) à cause du mécanisme Backoff qu'introduit 802.11 avant chaque nouvelle transmission. Ajoutons à cela le NAV qui oblige les stations voisines de l'émetteur et du récepteur de resté inactif jusqu'à la fin de la transmission, ce qui minimise les collisions. Par contre, nous remarquons une dégradation importante des performances de MACA et MACA-BI, cela est due au changement de topologie (causé par la mobilité) qui s'ajoute au longueur des chemins causé par l'augmentation du nombre de nœuds, ces deux facteurs ensemble montrent une influence sur la fraction de réception de ces protocoles, qui n'utilisent pas l'écoute du support et qui engendrent beaucoup de collisions.

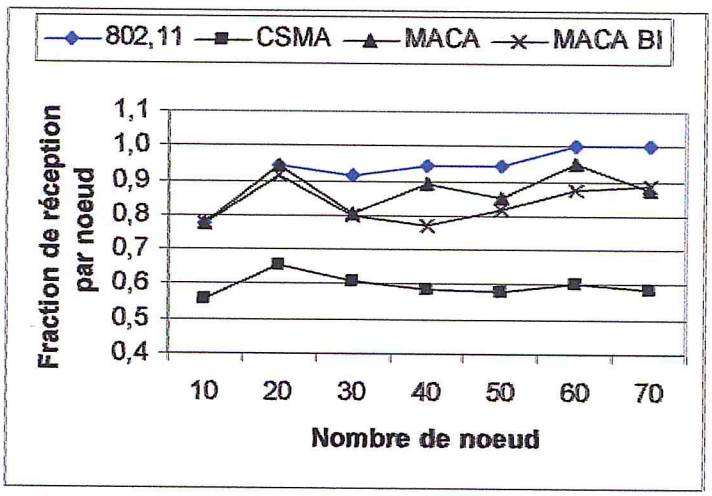


Figure IV.11a : La fraction de réception en fonction de la scalabilité – Mobilité nulle

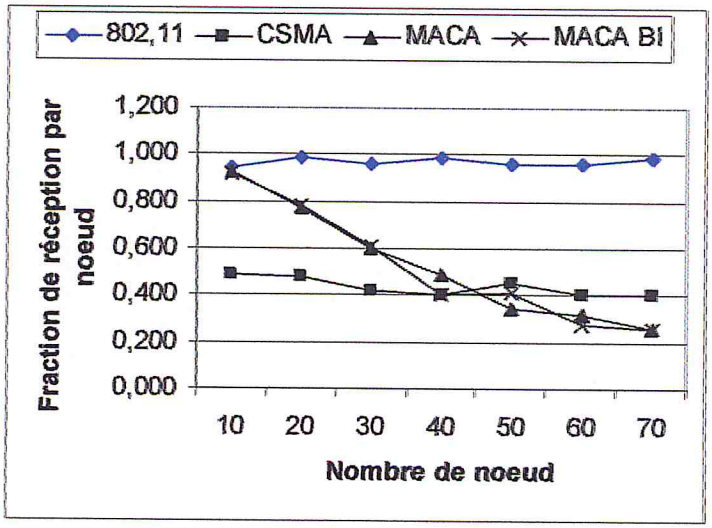


Figure IV.11b: La fraction de réception en fonction de la scalabilité – mobilité moyenne

IV.9.3.4. Le délai d'attente au niveau MAC

Quand le nombre de nœuds est faible, on remarque que le délai de MACA et MACA-BI est élevé, comparé a 802.11 et CSMA (la cause est la même que dans IV.9.2.4). Avec l'augmentation du nombre de nœuds, le délai de 802.11 est CSMA augmente légèrement. Le délai de CSMA se rapproche des délais de MACA et MACA-BI, mais celui de 802.11 reste meilleure. Ceci quand la mobilité est nulle. Quand on introduit la mobilité dans le réseau (Figure IV.12b), le délai de MACA et MACA-BI augmente considérablement (pour la même cause que dans IV.9.2.4), surtout quand le nombre de nœuds est faible ou très élevé.

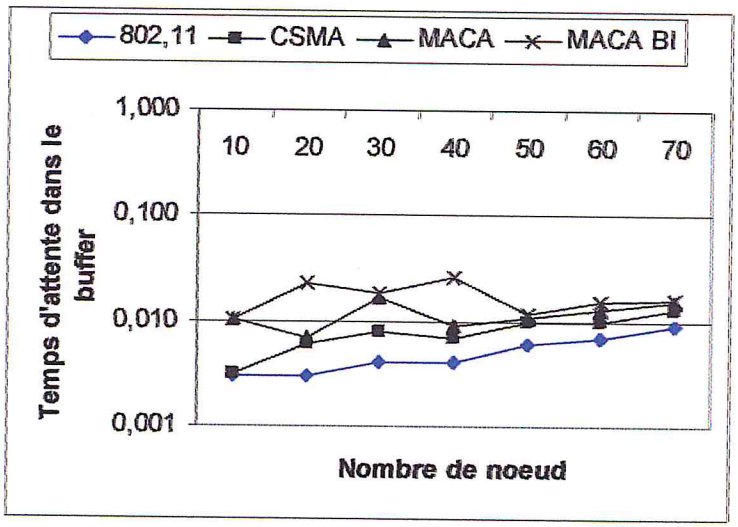


Figure IV.12a : Le délai en fonction de la scalabilité – mobilité nulle

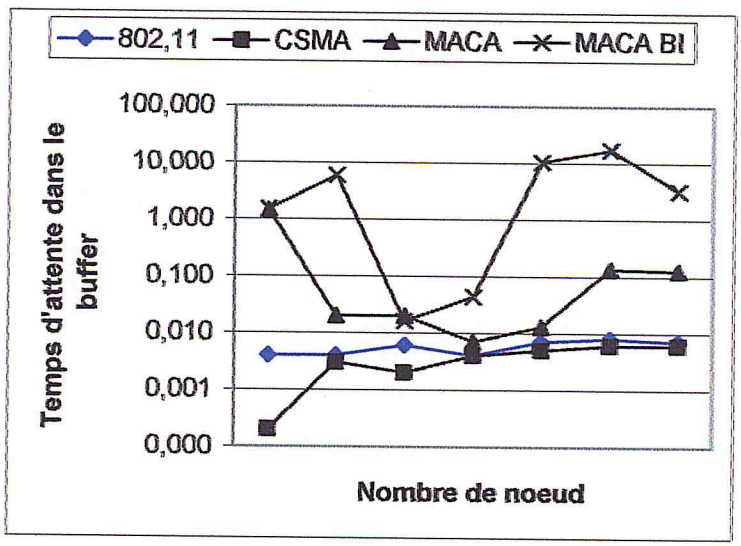


Figure IV.12b : Le délai en fonction de la scalabilité – mobilité moyenne

IV.10 Conclusion

Nous avons présenté dans ce chapitre une analyse comparative détaillée des protocoles MAC les plus connus, à savoir 802.11, CSMA, MACA et MACA-BI. Nous avons étudié l'effet de la mobilité et de la charge sur les performances des protocoles, ainsi que leurs scalabilité. Les performances des protocoles ont été mesurées en terme d'Energie consommée, de collisions, de fraction de réception et de délai d'attente au niveau de la couche MAC. Nous avons abouti aux résultats globaux suivants :

CSMA et 802.11 réalisent une consommation d'énergie meilleure dans toutes les situations. Par contre, la consommation de MACA et MACA-BI est élevé surtout quand la mobilité et/ou la charge augmentent. En plus, quand le réseau n'est pas chargé, MACA-BI consomme moins d'énergie par rapport à MACA.

802.11 assure une fraction de réception importante dans tous les scénarios. CSMA fait preuve de bonne performance en terme de collisions et de délai, par contre il échoue clairement en terme de fraction de réception, et cause une perte de données importante surtout quand la mobilité et/ou la charge du réseau deviennent importantes[36].

Quelle que soit la mobilité, le nombre de collisions de 802.11 et CSMA reste stable et meilleur, comparé à MACA et MACA-BI. Cependant, quand la charge du réseau augmente, les collisions augmentent aussi. En terme de collisions, MACA et MACA-BI sont influencés par l'augmentation de la mobilité et de la charge.

Dans un réseau stable et pas chargé, le délai d'attente des paquets de données de MACA et MACA-BI est meilleur, comparé à CSMA et 802.11, mais dés qu'une faible mobilité et/ou une faible charge apparaissent, les délais de MACA et MACA-BI augmentent considérablement par rapport à ceux de CSMA et 802.11.

En terme de scalabilité, nous avons constaté que le protocole 802.11 est le plus scalable comparé au autres protocoles, il s'adapte mieux à l'augmentation du nombre de nœuds et réalise une meilleur scalabilité vis-à-vis de toutes les métriques de performance mesurées.

[36] La fraction de CSMA est parfaite quand la mobilité est nulle et le réseau n'est pas chargé.

Conclusion

L'absence d'infrastructures centrales et le changement fréquent de la topologie dans les réseaux ad hoc rendent la fonction du contrôle d'accès au canal assez compliquée et difficile à mettre en place, comparée aux réseaux cellulaires. Au cours de ces dernières années, beaucoup de travaux de recherche on été consacré à la résolution de ce problème. Plusieurs protocoles MAC ont été proposés. Cependant, et comparé au problème de routage qui a été bien traité par les chercheurs, rares sont les analyses comparatives des protocoles MAC. D'ailleurs, aucune n'a pris en considération un réseau ad hoc avec des nœuds mobiles.

Ceci étant, nous avons pris en considération dans notre étude la mobilité. Nous avons comparé et évalué les performances des protocoles MAC les plus connus, à savoir 802.11, CSMA, MACA et MACA-BI. Pour cela, nous avons procédé en quatre chapitres. Dans le premier, nous avons donné une idée détaillée sur les réseaux mobiles ad hoc. Le deuxième chapitre, nous l'avons consacré à l'étude des protocoles MAC. Pour chaque protocole, nous avons spécifié les principales caractéristiques et fonctionnalités. Dans le troisième chapitre, nous avons introduit la notion de simulation, outil que nous avons utilisé pour réaliser notre étude. Nous y avons présenté également le langage de programmation PARSEC que nous avons utilisé pour implémenter le protocole MACA-BI, ainsi que les métriques utilisées dans la simulation. La présentation du simulateur GloMoSim, employé dans notre étude, a fait également partie de ce chapitre. Enfin, le quatrième chapitre répond à notre problématique de départ, à savoir la comparaison et l'évaluation des protocoles MAC. Notre études nous a mené aux résultats suivants :selon les quatre métriques de performance, 802.11 a largement dépassé les autres protocoles. Cependant, nous avons remarqué que sa fraction de réception de données se dégrade dans un réseau chargé et avec mobilité. En plus, ses performances en terme de collisions se dégradent avec l'augmentation de la charge du réseau. Le protocole CSMA, quant à lui, fonctionne très bien dans un réseau non-chargé, avec l'augmentation de la charge et/ou la mobilité, les performances de ce dernier se dégradent. Il cause plus de collisions, et réalise une fraction de réception assez faible. Les performances de MACA et MACA-BI se dégradent clairement quand la mobilité ou la charge augmente. Cependant, nous

avons remarqué que, dans un réseau statique et non chargé, les performances de MACA et MACA-BI sont meilleures en terme d'énergie consommée, de collisions, de fraction de réception et de délai d'attente. Nous avons remarqué aussi dans ce cas que l'approche orientée récepteur (Receiver-Initiated) utilisée par MACA-BI est égale ou meilleure comparée a l'approche orientée émetteur qu'utilise MACA (Sender-Initiated). Néanmoins, quand la charge du réseau augmente, l'approche par invitation échoue clairement, et cause une consommation d'énergie importante, un nombre de collisions élevé, une faible fraction de réception et un délai d'attente important.

Nous constatons donc, qu'aucun des protocoles MAC n'est parfaitement meilleur et adéquat pour les réseaux mobiles ad hoc.

Perspectives

- Aucun des protocoles MAC n'assure la fiabilité des paquets Broadcast (le mécanisme RTS/CTS n'est pas utilisé quand il s'agit d'un paquets broadcast). Ceci étant, il serait intéressent de développer une technique qui résout ce problème. Cela peut améliorer nettement les performances du protocole.
- Il serait intéressent de concevoir un nouveau protocole de contrôle d'accès au canal qui s'adapte mieux aux caractéristiques des réseaux ad hoc, au lieu d'adapter un protocole des réseaux sans fil.
- Notons qu'actuellement un axe de recherche très intéressent consiste en la conception d'une nouvelle classe de protocoles MAC dite les protocoles paramétrés, qui s'adaptent au changement de la topologie et de la charge du réseau. Cette classe est nommée PARADYCE (parameterized adaptive efficient protocols).

Références Bibliographiques

[Abr70] : N. Abramson. "The ALOHA system - Another alternative for computer communication". In AFIP Conf. Proc. Fall Joint Comput. Conf., 1970.

[Bar02] : C.L. Barret, M.Drozda , Marathe, M.V. "Comparative Experimental Study of Media Access Protocols for Wireless Radio Networks". In Proc. IEEE Wireless Communications and Networking Conference (WCNC'02), Orlando, Florida, March 2002, to appear

[Bha94] : V. Bharghavan et al. "MACAW: A Media Access Protocol for Wireless LANs". ACM SIGCOMM '94, ACM, pp. 212–25, 1994.

[Dja03] : Djamel Djenouri, Derhab Abdelouahib and Nadjib Badache. "Mobility impact on ad hoc routing protocols". ACS/IEEE international conference on computer systems & applications. Tunis, Tunisia 14-18 July 2003.

[Duc92] : D.Duchamp and N.F. Reynolds. "Measured performance of Wireless LAN". In 17th Conference on Local Computer Networks, pages 494--499, Minneapolis, Minnesota, September 1992. IEEE.

[For94] : G .H Forman and J .Zahrojan. "The challenge of mobile computing". IEEE Computer, 27 (4), pp 38-47, April 1994.

[Fro00] : Magnus Frodigh, Per Johansson and Peter Larsson. "Wireless ad hoc networking, The art of networking without a network". Ericsson Review No. 4, pp 248-263, 2000.

[Ful95] : C. L. Fullmer and J. J. Garcia-Luna-Aceves. "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks". Conf. Applications, Tech., Architectures and Protocols for Comp. Commun. (SIGCOMM), pp. 262–73, 1995.

[Gar96] : J. Garcia-Luna-Aceves and C. L. Fullmer. "Floor acquisition multiple access (FAMA) in single-channel packet-radio networks". Technical report, UCSC, Sept. 1996.

[JAY99]: Jay Martin. "GloMoSim Tutorial", 11/18/99 ,
url : http://pcl.cs.ucla.edu/slides/workshop99 /Jaytut-pw99/index.htm

[Kar90] : P. Karn. "MACA: A New Channel Access Protocol for Packet Radio". ARRL/CRRL Amateur Radio 9th Comp. Net. Conf., pp. 134–40, 1990.

[Kaw01] : V. Kawadia, S. Narayanaswamy, R. Rozovsky, R. S. Sreenivas, and P. R. Kumar. "Protocols for Media Access Control and Power Control in Wireless Networks". Proceedings of the 40th IEEE Conference on Decision and Control, pp. 1935-1940, Orlando, FL, Dec. 4-7, 2001.

[Kle75] : L. Kleinrock and F. A. Tobagi. "Packet Switching in Radio Channels: Part I: Carrier Sense Multiple-Access Models and their Throughput-Delay Characteristics". IEEE Trans. Commun., vol. 23, no. 12, pp. 1400–16, 1975.

[Lin97] : Chunhung Richard Lin and Mario Gerla. "MACA/PR: An Asynchronous Multimedia Multihop Wireless Network". In Proceedings of IEEE INFOCOM '97.

[Nas00] : A. Nasipuri, S. Ye, J. You and R.E. Hiromoto. "A MAC Protocol for Mobile Ad Hoc Networks Using Directional Antennas". Proc of the IEEE WCNC 2000.

[Pau03] : John Paul O Grady and Aidan McDonald. "State of the Art: Ad Hoc Networking", State of art surveys, Page 16 -136, May 2003 .
url : www.m-zones.org/deliverables/ d1_1/papers/1-02-adhoc_nw.pdf

[Per02] C.E. Perkins, E.M. Royer, and S.R. Das. "Ad hoc on-demand distance vector (AODV) routing". IETF Internet Draft, http ://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-12.txt, November 2002.

[Per99] C.E. Perkins and E.M. Royer. "Ad-hoc on-demand distance vector routing". In Proc. Second Annual IEEE Workshop on Mobile Computing Systems and Apllications, pages 90–100, February 1999.

[Pet02] : Peter. "Fundamental Knowledge Introduction and Some Novel Issues Survey in ad hoc wireless network". National Taiwan University,2002.
url : http://inrg.csie.ntu.edu.tw/2002/Survey%20of%20Ad%20Hoc%20Network.ppt

[Poz02] : Tim Pozar. "Les réseaux ad hoc : le futur des transmissions sans fil?". Paru le 09/04/2002. url: http://www.fing.org/index.php?num=2818,2.

[Puj01] : AL AGHA, Khaldoun, Guy PUJOLLE, et Guillaume VIVIER. "Réseaux de mobiles & réseaux sans fil". Paris, Eyrolles, 2001.

[Ris03] : Djamel Djenouri, Derhab Abdelouahib and Nadjib Badache. "Les protocoles de routage dans les réseaux ad hoc," RIST review (Revue sur l'information scientifique et technique), Vol 12 N 2, pp 77-112, Septembre 2003.

[Ros00] : Gail Rosen and Amin Atrash. "Ad Hoc Networking Extended Research Project". 7 june 2000

[Roy99] : Elizebeth M. Royer , C-K Toh. "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks". IEEE Personal Communications, pages 46-55, Apr 1999.

[Sin98] :Suresh Singh and Mike Woo. "Power Aware Routing in Mobile Ad Hoc Networks".ACM MOBICOM, Dallas, Texass,USA, pp. 181-190, 1998.

[Sob96] : J. L. Sobrinho and A. S. Krishnakumar. "Distributed multiple access procedures to provide voice communications over IEEE 802.11 wireless networks". In GLOBECOM '96, pages 1689-1694. IEEE, 1996.

[Sun01]: Jun-Zhao Sun. "Mobile ad hoc networking: an essential technology for pervasive computing". In Proc. International Conferences on Info-tech & Info-net, Beijing, China, pp. 316-321, 2001.

[Tal97] F. Talucci, M. Gerla and L. Fratta. "MACA-BI (MACA By Invitation):A Receiver-Oriented Access Protocol for Wireless Multihop Networks". Waves of the Year 2000, PIMRC '97, the 8th IEEE Int'l. Symp. Pers., Indoor and Mobile Radio Commun., vol. 2, pp. 435–39, 1997.

[Toh02]: C.-K. Toh. "Ad hoc Mobile Wireless Networks : Protocols and Systems". Prentice Hall PRT 2002.

[Tor02]: Patrick Tortelier (page consulté le 15/01/2004). " Les réseaux Ad Hoc".
url : http://cnfrs.gettelecom.fr/actualites/ag03_tortelier.pdf

[Xia98] : Xiang Zeng and Rajive Bagrodia and Mario Gerla. "{GloMoSim}: A Library for Parallel Simulation of Large-Scale Wireless Networks". Workshop on Parallel and Distributed Simulation, pages = "154-161", 1998.
url = "citeseer.nj.nec.com/zeng98glomosim.html"

[site1] http://www.fing.org/index.php?num=2818,2

ANNEXE :

ANNEXE A : PARSEC

Les mots clés de PARSEC :

After	**finalize**	**self**
at	**in**	**send**
clocktype	**message**	**stacksize**
driver	**new**	**timeout**
ename	**or**	**to**
entity	**receive**	**when**

La définition d'une entité :

entity-def : := **entity** ident **(** [*parameters*] [**stacksize** (*size_expr*)] *body*
parameters : := une liste de paramètres en langage C
size_expr : := une expression entière différente d'un tableau
body : := **{**une série d'instruction en PARSEC et en C [*finalize_st*]**}**
finalize_st : := **finalize {** une série d'instruction en C **}**

La création d'une entité :

new-st : := [*ename-expr* =] **new** ident**(** [*arg*] . . .**)** [**at** *node-no*]
ename-expr : := une expression de type ename
arg : := expression d'argument en C
node-no : := une expression entière de valeur positive

La déclaration d'un message :

Message-def : := **message** ident **{** *declarations***}** [ident] . . .;
Declarations : := [*type* ident[,ident] . . .;] . . .
Type : := **ename** | **clocktype** | **message** ident | une déclaration en C

Envoi de message :

send-st : := **send** *msg-expr* **to** *ename-expr* [**after** *time-expr*];
ename-expr : := une expression de type ename
msg-expr : := *msg-type* **{** [*arg*] . . . **}** | *msg-ident*
time-expr : := une expression entière positive
arg : := *array-param* | expression en C
array-param : := *un pointeur* [::une expression entière positive]
msg-type : := un type de message définie pour l'entité *ename-expr*
msg-ident : := une variable de type *msg-type*

Réception de message :

receive-st : := *resume-clause* [**or** *timeout-resume*]
resume-clause : := **receive (***msg-list***)** [**when** (*guard*)] *statement*
timeout-resume : := *receive-st* | *timeout-st*
timeout-st : := **timeout after** | **in (***delay-time***)** *statement*
msg-list : := *msg-type mvar* [, *msg-list*]
delay-time : := une expression en C de type **clocktype**
guard : := expression *sans side-effects*
mvar : := une variable de type *msg-type*
statement : := une instruction en C ou en PARSEC

Les fonctions de bibliothèque du langage Parsec :

Les opérations sur l'horloge de simulation :

- Simclock (void) : retourne la valeur de courante de l'horloge de simulation

- Setmaxclock (clocktype) : cette fonction précise le temps de simulation maximale au valeur spécifié dans le paramètre de clocktype.
- atoa(char*, clocktype*) : place la valeur de clocktype représenté par une chaîne de caractère dans le paramètre clocktype.
- ctoa(clocktype, char*) : elle mis la valeur de clocktype dans un format de chaîne de caractère

Les fonctions de simulation conservative :

- ***add_dest(ename)**: ajoute l'entité spécifiée à l'ensemble de destination de l'entité courante*
- ***add_source(ename)**: ajoute l'entité spécifiée à l'ensemble de source de l'entité courante.*
- ***del_dest(ename)**: supprime l'entité spécifiée de l'ensemble de destination de l'entité courante .*
- ***del_source(ename)**: supprime l'entité spécifiée de l'ensemble de source de l'entité courante.*
- ***setlookahead(clocktype, clocktype)**: précise la valeur de lookahead pour l'entité courante.*
- ***setdestlookahead(clocktype, clocktype, ename) :précise la valeur de lookahead pour une destination spécifié.***

Les générateurs de nombre aléatoire :

- ***double pc_erand(unsigned short[3])**: retourne une valeur entre [0.0, 1.0]*
- ***long pc_jrand(unsigned short[3])**: retourne une valeur entre [-2^{31}, 2^{31}]*
- ***long pc_nrand(unsigned short[3])**: retourne une valeur entre [0, 2^{31}]*

Compilation du programme Parsec :

Syntaxe : **pcc** [options] noms de fichiers

Le compilateur du langage Parsec (pcc) accepte toutes les options de compilation du langage C,

Ainsi les options de compilations suivantes :

```
-protocol Spécifie un des algorithmes de synchronisation:
          mpc Message-passing C: ignore les messages timestamps
          cons Conservative
          opt Optimistic
-c Genére des fichiers ".o" et ".pi.
-E Genére des fichiers ".c" et".pi" files.
-P Inhibe les numéros de ligne de translation. (Normalement, le
compilateur PARSEC insère les numéros de ligne dans le fichier C
intermédiaire, de sorte que le compilateur affiche les numéros
de ligne des erreurs correctement).

-env affiche les variables d'environnement de parsec.
      PCC_DIRECTORY : répertoire d'installation de PARSEC.
      PCC_CC  compilateur C utilisé par PARSEC.
      PCC_LINKER : linker utilisé par PARSEC.
      PCC_CC_OPTIONS : options pour passer au compilateur C.
      PCC_PP_OPTIONS : options par défaut du compilateur PARSEC.
      PCC_LINKER_OPTIONS : options par défaut du linker C.
-pcc_directory : utilise PCC_DIRECTORY spécifié.
-pcc_cc : utilise un compilateur C spécifie.
```

```
-pcc_linker : utilise un linker spécifie.
-pcc_pp_options : utilise ces options pour la précompilation.
-pcc_cc_options : utilise ces options pour la compilation.
-pcc_linker_options : utilise ces options pour l'édition de
lien.
-ini Sauvegarde les fichiers d'initialisation auto-generés.
-ff active la  compilation des fonctions amies.
stack Change la taille par défaut de la pile pour les entités.
-help affiche les options de compilation.
-V affiche le numéro de version du compilateur.
-user_main renomme la fonction main par parsec_main.
-shared_lib renomme la fonction main par parsec_main et crée une
bibliothèque partagée
-clock précise la représentation par défaut pour clocktype. Les
options valides sont:
unsigned unsigned long, un type entier de 32 bit (par défaut)
longlong long long, un type entier de 64 bit
```

Les exemples suivants montrent comment PARSEC compile des programmes dans une architecture séquentielle.

% pcc -o example example.pc

Cela génère un fichier exécutable *example* dans le répertoire de travail courant.

% pcc -o example example.pc xxx.c yyy.o

Cela génère un fichier exécutable nommé *example* dans le répertoire de travail courant. Le fichier *example* est compilé et lié avec xxx.c et yyy.o.

ANNEXE B : Glomosim

Les statistique Obtenue dans glomosim

Statistiques de la Couche radio (physique) :

Nombre total de paquets provenant de la couche mac
Nombre total de paquets de provenant du canal
Nombre total de collisions
L'énergie consommée.

Statistique de la couche MAC :

CSMA

Nombre de paquets provenant de la couche réseau
Nombre de paquets perdus dus au débordement du buffer
Nombre de paquets UNICAST envoyés a la couche canal
Nombre de paquets BROADCAST envoyés a la couche canal
Nombre de paquets UNICAST reçu clairement
Nombre de paquets BROADCAST reçu clairement

MACA

Nombre de paquets provenant de la couche réseau
Nombre de paquets perdus dus au débordement du buffer
Nombre de paquets UNICAST envoyés au canal
Nombre de paquets BROADCAST envoyés au canal
Nombre de paquets UNICAST reçu clairement
Nombre de paquets BROADCAST reçu clairement
Nombre de paquets RTS (Request To Send) envoyés
Nombre de paquets CTS (Clear To Send) envoyés
Nombre de paquets RTS obtenus
Nombre de paquets CTS obtenus
Nombre de paquets erronée «noisy packets» obtenus

802.11

Nombre de paquets provenant de la couche réseau
Nombre de paquets perdus dus au débordement du buffer
Nombre de paquets UNICAST (non-fragmenté) envoyés au canal
Nombre de paquets BROADCAS envoyés au canal
Nombre de paquets UNICAST reçus clairement
Nombre de paquets BROADCAST reçus clairement
Nombre de paquets retransmis dus à une expiration de délai des paquets CTS
Nombre de paquets retransmis dus à une expiration du délai des paquets ACK
(acquittement)
Nombre de paquets retransmis dus à une expiration du délai de paquet FRAG
ACK
Nombre de paquets perdus dus au dépassement du nombre maximum de retransmission

Statistiques de la Couche du réseau

Nombre de paquets provenant de TCP
Nombre de paquets envoyé à TCP
Nombre de paquets provenant d'UDP
Nombre de paquets envoyé à UDP
Nombre de paquets provenant d'OSPF
Nombre de paquets envoyé à OSPF
Nombre de paquets TCP perdus dus au dépassement ttl (time to live)
Nombre de paquets UDP perdus dus au dépassement ttl
Nombre de paquets OSPF perdus dus au dépassement ttl
Nombre moyen de sauts que les paquets TCP ont traversé
Nombre moyen de sauts que les paquets UDP ont traversé
Nombre moyen de sauts que les paquets OSPF ont traversé

Statistiques de la Couche de routage

Bellmanford

Nombre total de paquets qui ont bouclé
Nombre total de table de routage transmis
Nombre total de mises à jour des tables envoyées par événement
Nombre total de mises à jour des tables de routage
Nombre total de paquets reçus de la couche MAC
Nombre total de paquets reçus de la couche transport
Nombre total de paquets envoyés
Nombre total de paquets qui appartiennent à ce nœud
Nombre total de sauts traversés par les paquets du nœud
Nombre total de paquets perdus
Nombre total de paquets perdus dus à l'absence d'information de routage

OSPF

Le nombre de Paquet HELLO envoyé
Le nombre de Paquet HELLO Reçu
Nombre de Paquet LSA retransmis
Nombre total de Paquet LSA envoyé
Nombre de Paquet LSA Reçu
Nombre de paquet ACK d'état de lien envoyé
Nombre de paquet ACK d'état de lien reçu
Nombre de mises à jour de la Table de routage

DSR

Nombre de paquets de requêtes transmis
Nombre de paquets de réponses transmis
Nombre de paquets d'erreur transmis
Nombre total de paquets de contrôle transmis
Nombre de paquets de données transmis
Nombre de paquets de données générés

Nombre de paquets de données reçus

AODV

Nombre de paquets de requêtes transmis
Nombre de paquets de réponses transmis
Nombre total de paquets de contrôle transmis
Nombre de paquets de données transmis
Nombre de paquets de données générés
Nombre de paquets de données reçus

LAR

Nombre de paquets de requêtes générées
Nombre de paquets de requêtes transmis par les nœuds intermédiaires
Nombre de paquets de réponse générées
Nombre de paquets de réponse transmis par les nœuds intermédiaires
Nombre de paquets d'erreurs générés
Nombre de paquets d'erreurs transmis par les nœuds intermédiaires
Nombre de paquets de données générées
Nombre de paquets de données transmis par les nœuds intermédiaires

WRP

Nombre de paquets de routage envoyés
Nombre de paquets de routage reçue

FSR

Nombre de paquets de mises à jours de 1^{ier} niveau (intra scope)
Nombre de paquets de mises à jours de 2^{ime} neniveau (inter scope)
Nombre de paquets reçus du protocole UDP
Messages de contrôle en octets

Statistiques de la Couche du transport

TCP

Nombre de paquets envoyés à la couche réseau
Nombre de paquets de données envoyés
Nombre de Paquets de données en séquence
Nombre de paquets de données retransmis
Nombre de paquets ACK-ONLY envoyés
Nombre de paquets de contrôle (SYN|FIN|RST) envoyés
Nombre de paquets de Fenêtre mises à jour envoyés
Nombre de paquets de données reçus
Nombre de paquets ACK reçus en séquence
Nombre de paquets ACK dupliqués reçus
Nombre de paquets de contrôle (SYN|FIN|RST) reçus
Nombre totale de paquets erronés
Nombre de paquets reçus dont le champ CCKSUM indique une erreur
Nombre de paquets reçus avec une mauvaise taille

UDP

Nombre de paquets provenant de la couche application
Nombre de paquets envoyés à la couche application

Générateurs de trafic

FTP client / serveur

Le temps de début de session
Le temps de fin de session
Nombre d'octets envoyés
Nombre d'octets reçus
Le débit

Telnet client / serveur

Le temps de début de session
Le temps de fin de session
Nombre d'octets envoyés
Nombre d'octets reçus
Le débit

CBR client / serveur

Le temps de début de session
Le temps de fin de session
Nombre de paquets/octets envoyés
Nombre de paquets/octets reçus
Le débit
Le délai moyen de transfert de données

ANNEXE C : les Spécifications

1) MACA :

```
Variable Definitions
        TPROP = Propagation Delay across the channel
        TRTS = Time to transmit an RTS packet
        TCTS = Time to transmit a CTS packet
        TDATA = Time to transmit a DATA packet
Procedure START ( )
Begin
        call PASSIVE ( )
End

Procedure PASSIVE ( )
Begin
        While (No Packet Received & No Local Packet) wait;
        If (Packet Received) Then call REMOTE (received packet)
        Else call RTS ( );
End
Procedure RTS ( )
Begin
        Transmit RTS;
        Timer ← TCTS + 2TPROP;
        While (Timer not expired & No Packet Received) wait;
        If (Timer expired) Then call BACKOFF ( );
        Else DO CASE of (received packet type)
        Begin
                Local CTS_ call XMIT ( );
                Default: call REMOTE (received packet);
        End;
End
Procedure BACKOFF ( )
Begin
        Retransmit Timer ← 2* Retransmit Timer;
        Timer ← RANDOM (0, Retransmit Timer);
        While (Timer not expired & No Packet Received) wait;
        If (Timer expired) Then call PASSIVE ( );
        Else call REMOTE (received packet);
End
Procedure XMIT ( )
Begin
        Transmit Data Packet;
        Retransmit Timer ←1;
        call PASSIVE ( );
End
```

```
Procedure REMOTE (packet)
Begin
        DO CASE of (packet type)
        Begin
                Local RTS;
                        Transmit CTS;
                        timer ← TDATA;
                Other RTS: timer ← TCTS;
                CTS:    timer ← TDATA;
                DATA:
                        If (Local DATA) Then pass packet to upper layer;
                        call PASSIVE ( );
        End;
        While (Timer not expired & No Packet Received) wait;
        If (Timer expired) Then call PASSIVE ( );
        Else call REMOTE (received packet);
End
```

2) MACA-BI :

```
Procédure Passive :
        status = PASSIVE
        call Wait (random(Ts)) // Ts : average floor generation intertime

Procédure Send RTR :
        status = SEND RTR
        ghest neighbour's/buffer occupancy
        select neignbour
        adjust Ts
        transmit (RTR)
        call Wait(2*Tp) // Tp : temps de propagation maximum.

Procédure Receive :
        Receive (PCK)
        if status is REMOTE
        then switch (PCK)
                case : RTR to me
                        transmit (DATA)
                        call passive
                case : RTR to others
                        status = REMOTE
                        call Wait (Td+2*Tp) // Td : temps de transmission d'un
                                                   paquet de donnée.
                case : DATA
                        PCK aux couches supérieures
                        call Passive
                case : ERROR
                        call Passive
        else  switch (PCK)
                case : RTR to others
                        call Wait (MAX(Timer,Td+2*Tp))
                case : DATA
                        PCK aux couches supérieures
                        call  Wait (Timer)
                case : DEFAULT
                        call Wait (Timer)

Procédure Wait (T):
        Timer = T
        if (Timer non expirer et aucun paquet n'est détecter) wait
        if (un paquet est détecter) call Receive
        switch (status)
         case : PASSIVE
                call Send RTR
         case : SEND RTR
                call Passive
         case : REMOTE
                call Passive
```