



THALES

RAPPORT DE STAGE DE TROISIEME ANNEE

Forme d'onde Wi-Fi/OLSR pour réseaux ad-hoc tactiques

THALES COMMUNICATIONS

Responsable de stage :
Marc Bieth

Correspondante école :
Virginie Galtier

Supélec – Campus de Metz

Metz Technopole
2, rue Édouard Belin
57 070 Metz

Guillaume-Jean Herbiet

Élève en troisième année (option IIC)

LISTE DE DIFFUSION DU DOCUMENT

| Destinataire | Fonction |
|---------------|---|
| V. Galtier | Professeur à Supélec |
| M. Bieth | Ingénieur système, Thales Communications Responsable de stage |
| G. Michalon | Chef de laboratoire, Service Software Radio |
| R. Massin | Ingénieur système, Thales Communications |
| | |
| | |
| | |
| | |
| G.-J. Herbiet | Élève ingénieur, Supélec M. Sc. student in Computer Science, Georgia Institute of Technology |

RESUME

Le développement de conflits info-centrés et le développement des services accessibles par l'utilisation de protocoles IP conduisent à faire évoluer les communications tactiques vers des débits accrus et une flexibilité plus grande (auto configuration des réseaux, circulation de l'information affranchie d'une organisation hiérarchique).

Thales Communications répond à cette demande en développant des nœuds de communications tactiques basés sur des radios logicielles (postes programmables ou la forme d'onde utilisée est décorrélée du matériel) organisée en réseaux mobiles ad-hoc (dont les connections et le routage des informations s'établissent automatiquement en fonction de la topologie).

Diverses technologies civiles, reposant sur l'utilisation du standard 802.11 (Wi-Fi) en contexte multibond avec des protocoles de routage ad-hoc comme OLSR existent et la volonté de Thales est d'établir un tableau des performances ainsi qu'une cartographie des avantages et faiblesses de telles solutions.

Aussi, la présente étude, après une description de l'état de l'art du développement des technologies ad-hoc civiles, dépeint la mise en place d'un modèle d'une couche protocolaire basée sur CSMA/CA (couche d'accès au canal du standard 802.11) et du protocole de routage OLSR développé pour le simulateur de réseaux OMNeT++.

Ce modèle est par la suite utilisé pour simuler le comportement d'un réseau de nœuds utilisant cette architecture protocolaire et de mesurer les performances atteintes dans divers scénarios de topologie et de trafic et d'analyser le comportement de la forme d'onde dans ces conditions.

L'étude permet de montrer que, même si la forme d'onde basée sur Wi-Fi et OLSR se montre relativement performante (notamment à faible charge et lors de l'utilisation de flux TCP), elle ne semble pas assez robuste pour être envisagée dans des applications tactiques critiques.

Afin de corriger les faiblesses de la solution implémentée, diverses solutions, basées sur des standards récents (comme 802.11e) ou sur des recherches en cours (comme les améliorations possibles d'OLSR) sont proposées.

REMERCIEMENTS

Je tiens tout particulièrement à remercier :

- Marc Bieth, responsable de stage, pour sa compétence et son organisation ;
- Raphaël Massin, pour son aide et sa disponibilité ;
- Gilles Michalon, chef de Laboratoire ;
- Laurent Fachau, pour les informations sur les activités du service et la radio logicielle ;
- Sandrine Masson, pour l'aide apportée lors de l'intégration du routage OLSR à l'environnement de simulation ;
- L'ensemble du Service Software Radio, en particulier Sylvain Chabanne, Fabrice Desnoues et Bertrand Raulin, pour l'excellente ambiance au sein du service et les informations sur le mode de fonctionnement de l'entreprise.

SOMMAIRE

| | |
|---|-----------|
| 1. INTRODUCTION | 11 |
| 2. CONTEXTE DE L'ETUDE | 12 |
| 2.1 LE GROUPE THALES | 12 |
| 2.1.1 LES ORIGINES DE THALES | 12 |
| 2.1.2 LES DOMAINES DE COMPETENCES ET D'APPLICATIONS | 13 |
| 2.1.3 DONNEES CLES..... | 15 |
| 2.1.3.1 Une approche multidomestique | 15 |
| 2.1.3.2 Chiffres de 2005 | 16 |
| 2.1.4 DIVISION SYSTEMES TERRE ET INTERARMEES (DLJ) | 17 |
| 2.1.4.1 Généralités | 17 |
| 2.1.4.2 Activités | 17 |
| 2.1.4.3 L'unité de Communications tactiques | 17 |
| 2.1.4.4 Le service software radio | 18 |
| 2.2 L'EVOLUTION DES COMMUNICATIONS TACTIQUES | 19 |
| 2.2.1 RAPIDE ETAT DE L'EXISTANT | 19 |
| 2.2.2 GUERRE INFOCENTREE ET EVOLUTION DES BESOINS | 19 |
| 2.3 LE CONCEPT DE RESEAU AD-HOC | 20 |
| 2.3.1 PARADIGME DES RESEAUX AD-HOC MOBILES | 21 |
| 2.3.2 INTERET DES RESEAUX AD-HOC | 21 |
| 2.3.3 PRINCIPAUX ENJEUX..... | 21 |
| 2.3.3.1 Le partage du canal de communication | 21 |
| 2.3.3.2 Le routage des données | 22 |
| 3. PROBLEMATIQUES PARTICULIERES DES RESEAUX AD-HOC | 23 |
| 3.1 ROUTAGE EN RESEAU AD-HOC | 23 |
| 3.1.1 CLASSIFICATION DES ALGORITHMES..... | 23 |
| 3.1.1.1 Vecteur de distance et état du lien | 23 |
| 3.1.1.2 Protocoles proactifs et protocoles réactifs | 23 |
| 3.1.1.3 Autres éléments de taxonomie | 24 |
| 3.1.2 OLSR | 25 |
| 3.1.2.1 Principes d'OLSR | 25 |
| 3.1.2.2 Fonctionnement du protocole OLSR | 31 |
| 3.1.2.3 Simulations et expérimentations d'OLSR | 34 |
| 3.1.2.4 Optimisations d'OLSR | 35 |
| 3.1.2.5 Fast-OLSR | 36 |
| 3.2 IEEE 802.11 DANS LES RESEAUX MOBILES AD-HOC | 39 |
| 3.2.1 LE STANDARD 802.11 DE L'IEEE..... | 39 |
| 3.2.1.1 La couche physique | 40 |
| 3.2.1.2 La couche d'accès au média | 40 |
| 3.2.2 LES LIMITATIONS DE 802.11 DANS LES RESEAUX MOBILES AD-HOC..... | 47 |
| 3.2.2.1 L'importance du modèle de propagation | 48 |
| 3.2.2.2 Interactions avec la couche TCP | 52 |
| 3.3 AMELIORATIONS DES STANDARDS POUR LES RESEAUX MOBILES AD-HOC | 55 |
| 3.3.1 AMELIORATIONS DE TCP | 55 |
| 3.3.2 AMELIORATIONS DE LA COUCHE MAC | 56 |
| 3.3.2.1 Améliorations du protocole MACA..... | 57 |
| 3.3.2.2 Distributed Packet Reservation Multiple Access Protocol (D-PRMA)..... | 61 |
| 3.3.2.3 Distributed Priority Scheduling (DPS)..... | 61 |

| | | |
|------------|---|------------|
| 3.4 | QUALITE DE SERVICE DANS LES RESEAUX MOBILES AD-HOC | 62 |
| 3.4.1 | LA NORME 802.11E | 62 |
| 3.4.2 | PERFORMANCES DE 802.11E | 64 |
| 3.4.2.1 | Analyse des performances | 64 |
| 3.4.2.2 | Comportement avec des réseaux ad-hoc multibonds | 66 |
| 4. | ENVIRONNEMENT DE SIMULATION | 69 |
| 4.1 | OUTILS DE SIMULATION..... | 69 |
| 4.1.1 | OMNET++ | 70 |
| 4.1.2 | MOBILITY FRAMEWORK | 70 |
| 4.2 | ARCHITECTURE DE LA SIMULATION..... | 71 |
| 4.2.1 | GENERATEUR DE TRAFIC..... | 73 |
| 4.2.1.1 | Principe de fonctionnement | 73 |
| 4.2.1.2 | Fichier de description du trafic..... | 74 |
| 4.2.1.3 | Modifications apportées au générateur de trafic..... | 76 |
| 4.2.2 | GESTION DE LA MOBILITE | 77 |
| 4.2.2.1 | Principe de fonctionnement | 77 |
| 4.2.2.2 | Fichier de description de la mobilité | 78 |
| 4.2.2.3 | Modifications apportées au mobility framework..... | 79 |
| 4.2.3 | ROUTAGE DES DONNEES..... | 80 |
| 4.2.3.1 | Principe de fonctionnement | 80 |
| 4.2.3.2 | Adaptation à l'environnement de simulation | 81 |
| 4.2.4 | GESTION DE L'ACCES AU CANAL | 83 |
| 4.2.4.1 | Principe de fonctionnement | 83 |
| 4.2.4.2 | Adaptation aux contraintes des communications tactiques | 84 |
| 4.2.5 | MODELISATION DU CANAL RADIO | 86 |
| 4.2.5.1 | Description du modèle retenu..... | 86 |
| 4.2.5.2 | Implémentation du modèle | 87 |
| 5. | SIMULATIONS ET ANALYSE DES RESULTATS..... | 90 |
| 5.1 | REPONSE EN DEBIT SANS MOBILITE | 90 |
| 5.1.1 | DESCRIPTION DU SCENARIO | 90 |
| 5.1.2 | PRESENTATION ET ANALYSE DES RESULTATS..... | 91 |
| 5.1.2.1 | Résultats obtenus..... | 91 |
| 5.1.2.2 | Analyses et justifications | 92 |
| 5.1.3 | OPTIMISATIONS DE LA SIMULATION | 96 |
| 5.1.3.1 | Trames de routages prioritaires..... | 96 |
| 5.1.3.2 | Tolérance aux collisions | 98 |
| 5.1.3.3 | Zone de sensibilité intermédiaire..... | 99 |
| 5.1.4 | AMELIORATIONS DE LA FORME D'ONDE..... | 99 |
| 5.2 | MOBILITE ALEATOIRE | 100 |
| 5.2.1 | DESCRIPTION DU SCENARIO | 100 |
| 5.2.2 | PRESENTATION ET ANALYSE DES RESULTATS..... | 101 |
| 5.2.3 | INFLUENCE DU NOMBRE DE BONDS SUR LE DEBIT TCP..... | 102 |
| 5.2.4 | AMELIORATIONS DE LA FORME D'ONDE..... | 103 |
| 5.3 | 100 STATIONS DONT 4 MOBILES..... | 103 |
| 5.3.1 | DESCRIPTION DU SCENARIO | 103 |
| 5.3.2 | PRESENTATION ET ANALYSE DES RESULTATS..... | 104 |
| 5.4 | CLUSTER DE 7 STATIONS ET MOBILITE..... | 106 |
| 5.4.1 | DESCRIPTION DU SCENARIO | 106 |
| 5.4.2 | PRESENTATION ET ANALYSE DES RESULTATS..... | 106 |
| 5.4.3 | IMPACT DE LA TAILLE DE LA ZONE DE SENSIBILITE A LA PORTEUSE | 107 |

| | | |
|------------|---|------------|
| 5.5 | FUSION DE RESEAUX | 109 |
| 5.5.1 | DESCRIPTION DU SCENARIO | 109 |
| 5.5.2 | PRESENTATION ET ANALYSE DES RESULTATS..... | 110 |
| 5.6 | SCENARIO OPERATIONNEL : SECURISER BRINON | 112 |
| 5.6.1 | DESCRIPTION DU SCENARIO | 112 |
| 5.6.2 | PRESENTATION ET ANALYSE DES RESULTATS..... | 113 |
| 5.7 | SYNTHESE DES RESULTATS | 117 |
| 6. | BILAN | 119 |

TABLE DES FIGURES

| | |
|---|----|
| Figure 2.1 : Le site de Colombes (Magellan) de Thales Communications..... | 12 |
| Figure 2.2 : Organigramme du groupe | 14 |
| Figure 2.3 : Effectifs du groupe Thales par zone géographique | 15 |
| Figure 2.4 : Destination géographique de l'activité du groupe Thales | 15 |
| Figure 2.5 : Activité et effectifs par division | 16 |
| Figure 2.6 : Structure de l'actionnariat du groupe Thales | 16 |
| Figure 2.7 : Analogie avec l'univers informatique..... | 18 |
| Figure 2.8 : Organisation traditionnelle des communications tactiques | 19 |
| Figure 2.9 : Évolution des besoins de communication | 20 |
| Figure 3.1 : Taxonomie des protocoles de routage | 25 |
| Figure 3.2 : Diffusion de messages avec les MPR..... | 26 |
| Figure 3.3 : Format du paquet OLSR | 27 |
| Figure 3.4 : Format du paquet MID | 28 |
| Figure 3.5 : Format du paquet <i>HELLO</i> | 29 |
| Figure 3.6 : Format du paquet TC | 30 |
| Figure 3.7 : Exemple de réseau ad-hoc | 31 |
| Figure 3.8 : Découverte de voisinage par échange de messages <i>HELLO</i> | 32 |
| Figure 3.9 : Réseau après choix des MPRs par N_0 | 33 |
| Figure 3.10 : Modèle de simulation de Fast-OLSR | 38 |
| Figure 3.11 : Architecture protocolaire de 802.11 | 39 |
| Figure 3.12 : Scénario de fonctionnement de CSMA/CA | 42 |
| Figure 3.13 : Transfert de données standard de 802.11 | 43 |
| Figure 3.14 : Configuration du nœud caché | 43 |
| Figure 3.15 : Transfert de données avec extension RTS/CTS | 44 |
| Figure 3.16 : Synchronisation en mode ad-hoc..... | 46 |
| Figure 3.17 : Gestion de l'énergie en mode ad-hoc | 47 |
| Figure 3.18 : Modèle radio raffiné..... | 49 |
| Figure 3.19: Le problème du nœud caché | 50 |
| Figure 3.20 : Le problème du nœud exposé | 51 |
| Figure 3.21 : Connexion en trois passes de TCP..... | 53 |
| Figure 3.22 : Dynamique de la fenêtre de congestion TCP | 54 |
| Figure 3.23 : Taxonomie des protocoles MAC ad-hoc | 56 |
| Figure 3.24 : Utilité du protocole MACAW (cas 1)..... | 58 |
| Figure 3.25 : Utilité du protocole MACAW (cas 2)..... | 59 |
| Figure 3.26 : Comparaison MACA/MARCH | 60 |
| Figure 3.27 : Fonctionnement de MACA/PR | 60 |
| Figure 3.28 : Structure de trame D-PRMA | 61 |
| Figure 3.29 : Fonctionnement de DPS | 62 |
| Figure 3.30 : Scénario statique de test d'efficacité de 802.11e | 66 |
| Figure 4.1 : L'environnement de simulation <i>tkenv</i> | 69 |
| Figure 4.2 : Architecture modulaire de OMNeT++ | 70 |
| Figure 4.3 : Architecture d'un nœud du mobility framework..... | 71 |
| Figure 4.4 : Architecture de la simulation | 72 |
| Figure 4.5 : Diagramme de collaboration du générateur de trafic..... | 73 |

| | |
|---|-----|
| Figure 4.6 : Exemple de fichier de trafic | 75 |
| Figure 4.7 : Exemple de fichier de mobilité | 78 |
| Figure 4.8 : Représentation des cordonnées | 79 |
| Figure 4.9 : Détails du calcul des durées inter-frames | 84 |
| Figure 4.10 : Modèle radio de la simulation | 87 |
| Figure 4.11 : Calcul du temps de collision d'un message | 89 |
| Figure 5.1 : Grille de 16 stations..... | 90 |
| Figure 5.2 : Réponse en débit du réseau pour deux valeurs de la zone de sensibilité à la porteuse..... | 92 |
| Figure 5.3 : Causes de perte de paquets (zone de sensibilité réduite)..... | 93 |
| Figure 5.4 : Causes de perte de paquets (zone de sensibilité étendue)..... | 93 |
| Figure 5.5 : Types collisions pour la zone de sensibilité réduite (à gauche) et étendue (à droite) pour un débit soumis de 800 kbits/s | 94 |
| Figure 5.6 : Temps passé dans les différents états radio durant la simulation pour la zone de sensibilité réduite (à gauche) et étendue (à droite) pour un débit soumis de 800 kbits/s..... | 95 |
| Figure 5.7 : Impact de la taille de la file MAC sur le fonctionnement d'OLSR..... | 95 |
| Figure 5.8 : Réponse en débit et temps de latence avec flux OLSR prioritaire | 96 |
| Figure 5.9 : Congestions du trafic de routage à 400 kbits/s | 97 |
| Figure 5.10 : Impact d'une file prioritaire OLSR sur le comportement du réseau (à g.) et du protocole de routage (à dr.)..... | 97 |
| Figure 5.11 : Influence du nombre de bonds et de la zone de sensibilité à la porteuse sur le débit TCP (pour 1 flux à gauche, pour deux flux tête-bêche à droite) | 102 |
| Figure 5.12 : Grille de 100 stations..... | 104 |
| Figure 5.13 : Nombre moyen de MPRs pendant la simulation..... | 105 |
| Figure 5.14 : Scénario de simulation avec cluster de 7 stations | 106 |
| Figure 5.15 : Scénario de fusion de réseau | 109 |
| Figure 5.16 : Évolution du voisinage (physique et vu d'OLSR) pour le nœud 10 durant la simulation | 111 |
| Figure 5.17 : Engorgement du nœud 1 | 111 |
| Figure 5.18 : Scénario opérationnel (sécuriser Brinon)..... | 113 |
| Figure 5.19 : Pourcentage de paquets reçus pour les différents flux..... | 114 |
| Figure 5.20 : Évolution de la taille des files MAC dans le scénario opérationnel..... | 115 |
| Figure 5.21 : Évolution du voisinage du nœud 2 pendant le scénario opérationnel | 116 |

TABLE DES TABLEAUX

| | |
|---|-----|
| Tableau 3.1: Classification simplifiée des protocoles de routage ad-hoc | 24 |
| Tableau 3.2 : Intervalles d'émission standard du protocole OLSR | 36 |
| Tableau 3.3 : Paramètres du standard 802.11 | 41 |
| Tableau 3.4 : Distance de transmissions des paquets de données et de contrôle | 49 |
| Tableau 3.5: Solutions d'adaptation de TCP aux réseaux ad-hoc | 57 |
| Tableau 3.6: Paramètres de priorité dans 802.11e (cf. [13]) | 63 |
| Tableau 4.1 : En-têtes ajoutés dans les différents modules d'un nœud de simulation | 74 |
| Tableau 4.2 : Interprétation du fichier de trafic | 75 |
| Tableau 4.3 : Interprétation du fichier de mobilité | 79 |
| Tableau 4.4 : Paramètres de calcul des durées inter-trames | 85 |
| Tableau 4.5 : Durées inter-trames pour les standards 802.11 et la forme d'onde tactique | 85 |
| Tableau 4.6 : Paramètres physiques de la simulation | 87 |
| Tableau 5.1 : Résultats de la simulation de réponse en débit sans mobilité | 91 |
| Tableau 5.2 : Impact de la résistance aux collisions | 98 |
| Tableau 5.3 : Impact de la zone de sensibilité intermédiaire | 99 |
| Tableau 5.4 : Résultats de la simulation à mobilité aléatoire (trafic UDP) | 101 |
| Tableau 5.5 : Résultats de simulation avec la grille de 100 stations | 104 |
| Tableau 5.6 : Résultats de la simulation avec cluster de stations (trafic UDP) | 107 |
| Tableau 5.7 : Résultats de la simulation avec cluster de stations et zone de sensibilité à la porteuse réduite (trafic UDP) | 107 |
| Tableau 5.8 : Résultats pour la simulation de fusion de réseaux (trafic UDP) | 110 |
| Tableau 5.9 : Résultats de simulation avec la grille de 100 stations | 113 |
| Tableau 5.10 : Conclusions sur les performances de la forme d'onde | 117 |

FORME D'ONDE WI-FI/OLSR POUR RESEAUX AD-HOC TACTIQUES

1. INTRODUCTION

Au terme de trois années de formation d'ingénieur à Supélec, le stage de fin d'étude est une étape de transition vers la vie active, permettant à la fois de mettre en application l'ensemble des connaissances acquises dans le cadre d'un projet industriel et d'intégrer, dans le même temps, le monde de l'entreprise.

Durant ma scolarité, à la fois à Supélec et au Georgia Institute of Technology, j'ai pu développer diverses connaissances dans le domaine des réseaux informatiques, en particulier concernant réseaux sans-fil (type Wi-Fi ou WiMAX) et cellulaires.

Mon cursus de double-diplôme avec ces deux établissements m'autorisant à effectuer une mission longue (plus de sept mois), j'ai donc cherché à effectuer mon stage dans ce domaine, au sein d'un grand groupe à même de me proposer un sujet intéressant et innovant. De plus, souhaitant continuer ma formation avec un *philosopher degree* (équivalent américain du doctorat), mon intérêt s'est surtout porté sur des missions de recherche et développement.

Aussi, c'est avec grand enthousiasme que j'ai répondu à l'offre faite par le Service Software Radio (SSR) de Thales Communications. Cette équipe travaille sur le développement de réseaux sans-fil « ad-hoc », domaine que j'avais découvert dans le cadre d'un cours électif à Supélec et qui m'avait fortement intéressé.

Le sujet proposait d'établir une architecture réseau basée sur des technologies civiles adaptées aux contraintes militaires et d'en mesurer les performances afin d'obtenir un « étalon » des résultats atteignables à partir d'une solution civile en y effectuant peu de modifications. Ce qui allait me permettre d'avoir une plus grande connaissance et une plus grande maîtrise, théorique et expérimentale, des différentes couches système (accès au canal, routage, ...) mise en jeux dans de tels réseaux.

Les travaux de Thales, non présentés dans ce document pour des raisons de confidentialité, ont ensuite été comparés à la solution ainsi obtenue.

Mon stage s'est donc déroulé du 2 mai au 15 décembre 2006, au sein du Service Software Radio, dont les locaux sont situés au sein du bâtiment de Thales Communications à Colombes (dans les Hauts-de-Seine).

2. CONTEXTE DE L'ETUDE

Cette partie présente le contexte global dans lequel s'est déroulé le stage : profil et activités de la société hôte (Thales Communications), importance stratégique des communications mobiles dans le domaine tactique et introduction aux techniques et enjeux des réseaux mobiles ad-hoc.

2.1 LE GROUPE THALES

Thales est un groupe d'envergure internationale, fournisseur et maître d'œuvre de systèmes électroniques et de communication pour les marchés de la Défense, de l'Aéronautique et de la Sécurité.

Présent sur les 5 continents, Thales emploie plus de **60 000 collaborateurs** dans plus de **50 pays** et a atteint en 2005, un **chiffre d'affaires de 10,3 Milliards d'Euros**.



Figure 2.1 : Le site de Colombes (Magellan) de Thales Communications

2.1.1 LES ORIGINES DE THALES

Thomson-Brandt a été créé en 1893 et était chargée d'exploiter, en France, les brevets de la Thomson Houston Electric Corporation dans le domaine de la production et du transport de l'électricité.

CSF quant à elle, fut créée en 1918 et était l'un des pionniers dans le domaine des transmissions hertziennes. Avant la seconde guerre mondiale, elle joue un rôle important dans le développement de la radiodiffusion, des radiocommunications sur ondes courtes, de l'électroacoustique, du radar et de la télévision. Elle développe également l'industrie des semi-conducteurs au silicium.

En 1968, Thomson-Brandt fusionne avec la CSF (Compagnie générale de télégraphie Sans Fil) créant ainsi Thomson-CSF.

Après la chute du mur de Berlin, l'environnement des industries de défense est profondément et durablement altéré. Thomson-CSF restructure ses activités et engage une politique active de croissance externe. Elle fait l'acquisition de sociétés européennes.

Thomson-CSF réussit alors à étoffer son large spectre d'activités d'électronique de défense et aussi ses activités d'électronique professionnelle civile. Elle est désormais mondialement présente.

En 1996 débute un processus de privatisation donnant lieu à un regroupement, autour de Thomson-CSF et dans le cadre d'un partenariat stratégique avec Alcatel, des activités électroniques spatiales et de défense et des activités de communication militaires d'Alcatel, des activités électroniques professionnelles et de défense de Dassault électronique, ainsi que des activités satellites d'Aérospatiale. Le 14 avril 1998, ces 4 sociétés signent un accord de coopération.

En 2000, la dynamique d'implantation multi-domestique des activités de défense dépasse les frontières européennes. Le groupe se développe encore par acquisition et croissance interne et en décembre 2000, Thomson-CSF devient Thales.

Depuis 2001, dans un contexte géopolitique et économique profondément bouleversé à la suite des attentats du 11 septembre, Thales renforce ses compétences et son engagement dans les domaines les plus technologiques de l'industrie de la défense, notamment les communications et la transmission de données des forces armées.

Thales développe également son savoir-faire de maître d'œuvre et ses activités de services, pour mieux répondre aux attentes des États prescripteurs, confrontés à la complexité croissante des programmes et à la haute technicité des systèmes et des équipements de défense.

2.1.2 LES DOMAINES DE COMPETENCES ET D'APPLICATIONS

Leader mondial dans l'aéronautique, la défense et la sécurité, présent sur tous les continents, Thales s'appuie sur un réseau de plus de **60 000 collaborateurs**, pour proposer à ses clients, une **innovation constante** (la R&D, Recherche et Développement représente 1,9 milliards d'euros, soit 18 % du chiffre d'affaires) et une **excellence technologique (20 000 chercheurs** dans les secteurs de pointe, réalisent **250 inventions par an** en moyenne, validées par 12 000 brevets).

Parmi ses **principaux clients**, on trouve, le Ministère de la défense, la DGA (Direction Générale de l'Armement), les trois coprs d'armée, l'OTAN ou des organismes publics ou civils comme les Ministère des Finances, le musée du Louvre, le musée des Arts Premiers du Quai Branly, ou le Stade de France.

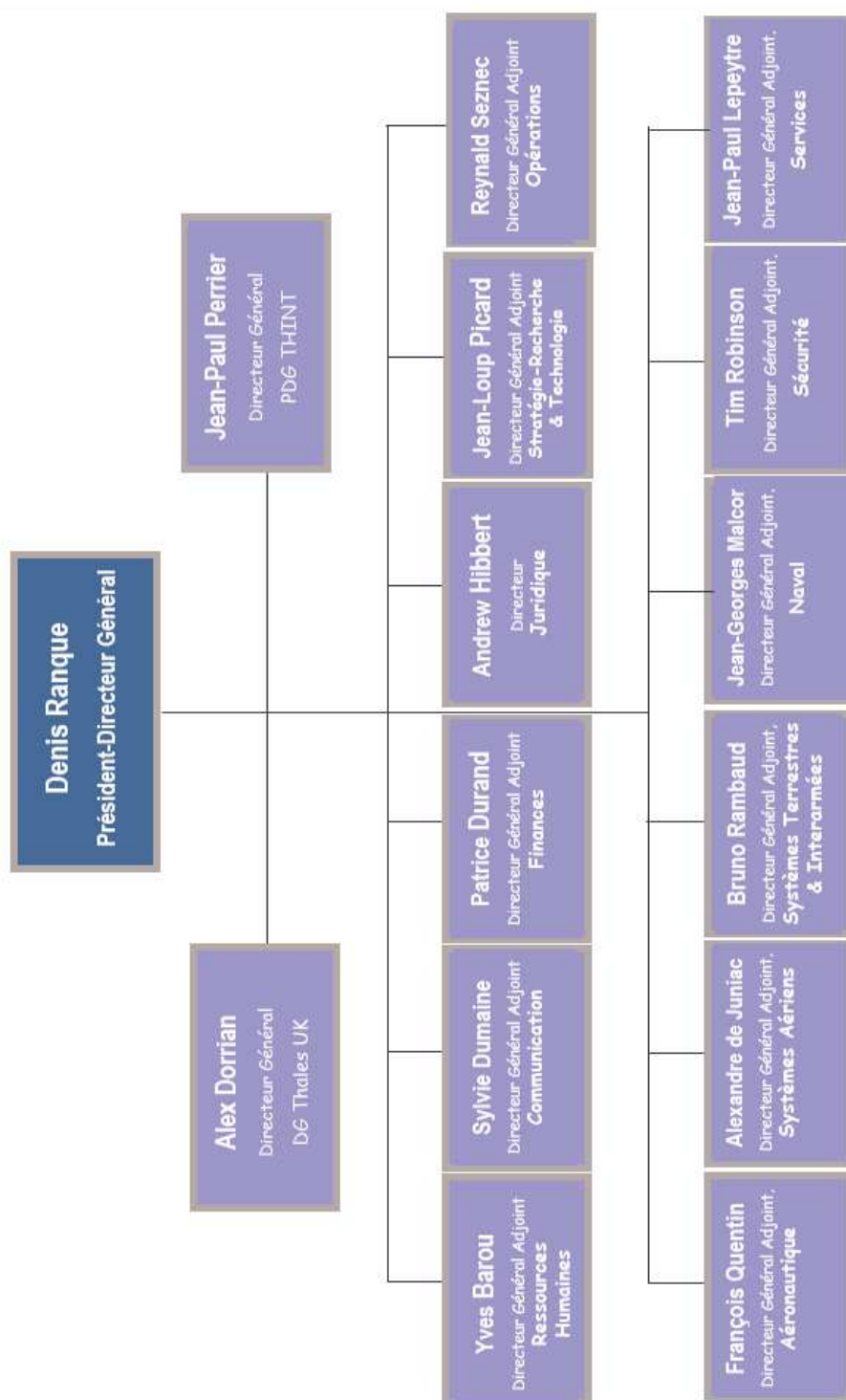


Figure 2.2 : Organigramme du groupe

Thales se compose de **6 divisions** fonctionnant en synergie, afin d'assurer aux clients leur présence internationale, une distribution de l'information en temps réel, créant ainsi une **plate-forme commune** de technologies et de compétitivités de proximités.

2.1.3 DONNEES CLES

2.1.3.1 UNE APPROCHE MULTIDOMESTIQUE

Présent à travers le monde, Thales est un acteur mondial de la défense et de la sécurité. Pour offrir à ses nombreux clients une infrastructure industrielle facilitant la participation aux différents programmes nationaux et répondre à leurs exigences de sécurité nationale, le groupe adopté une approche multidomestique.

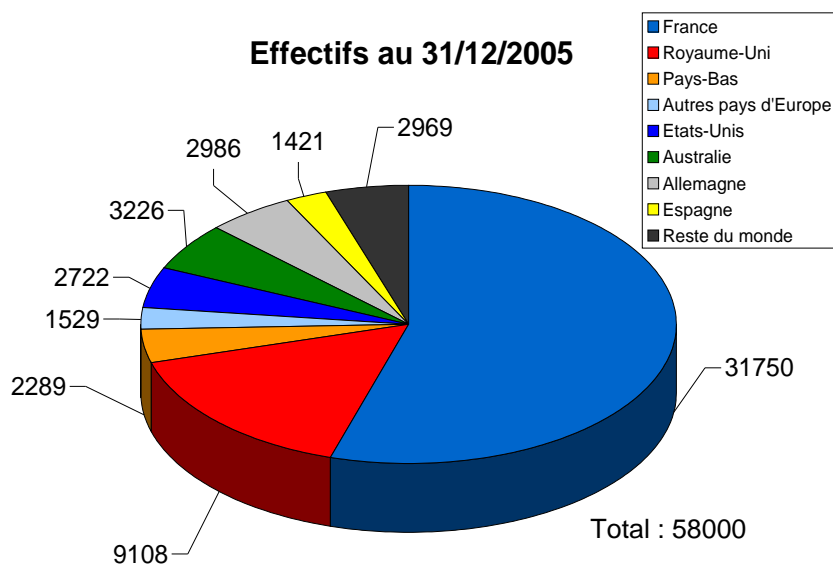


Figure 2.3 : Effectifs du groupe Thales par zone géographique

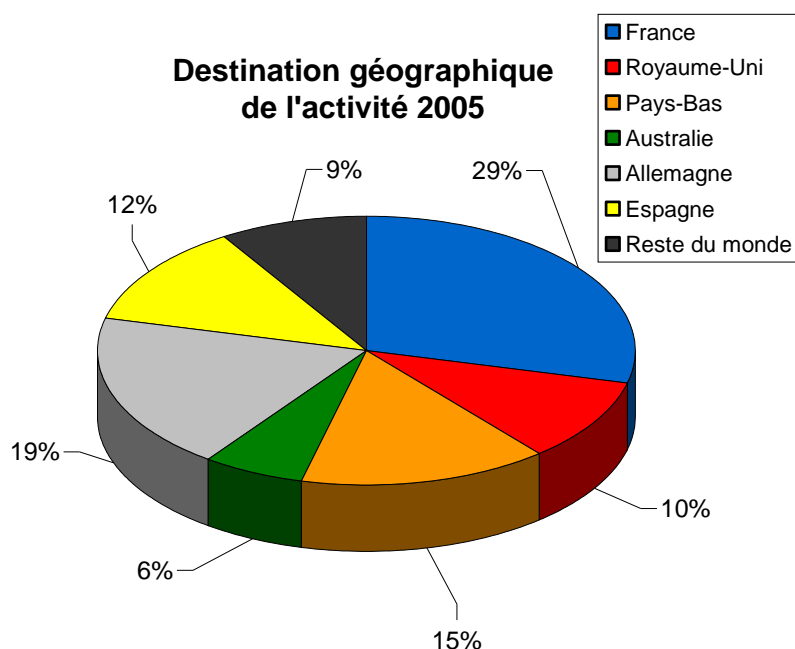


Figure 2.4 : Destination géographique de l'activité du groupe Thales

Le découpage du groupe en entités nationales, tout en assurant davantage de moyens financiers et une plus grande synergie entre les programmes de R&D, permet un ajustement plus étroit aux demandes du client et une meilleure pénétration dans les marchés sensibles de la sécurité intérieure et la défense nationale.

2.1.3.2 CHIFFRES DE 2005

- Carnet de commandes : 18,7 milliards d'Euros ;
- Chiffre d'affaires : plus de 10 milliards d'Euros ;
- Recherche et développement : 1,9 milliards d'Euros (18% du CA) dont 400 millions d'Euros autofinancés ;
- 60 000 experts dont 27 000 hors France ;
- 20 000 chercheurs dans les technologies de pointe ;
- 30 accords de coopérations avec des universités et laboratoires de recherches publics d'Europe, d'Asie et des États-Unis ;
- 171,9 millions d'actions (dont seulement 51,7% flottant en bourse).

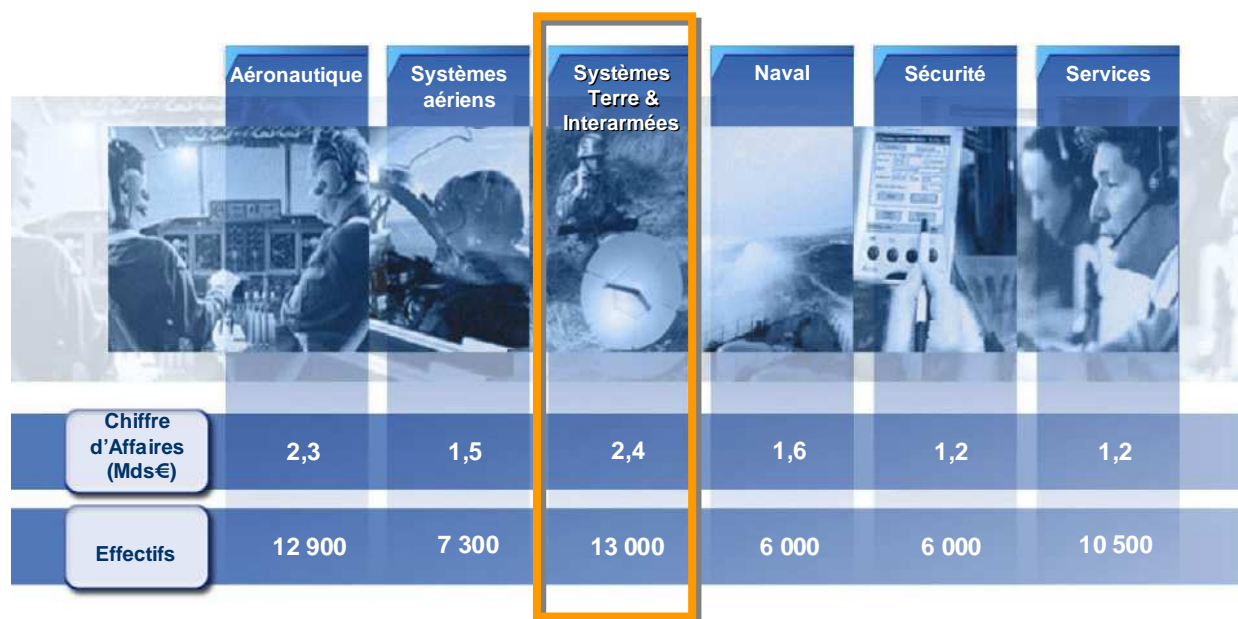


Figure 2.5 : Activité et effectifs par division

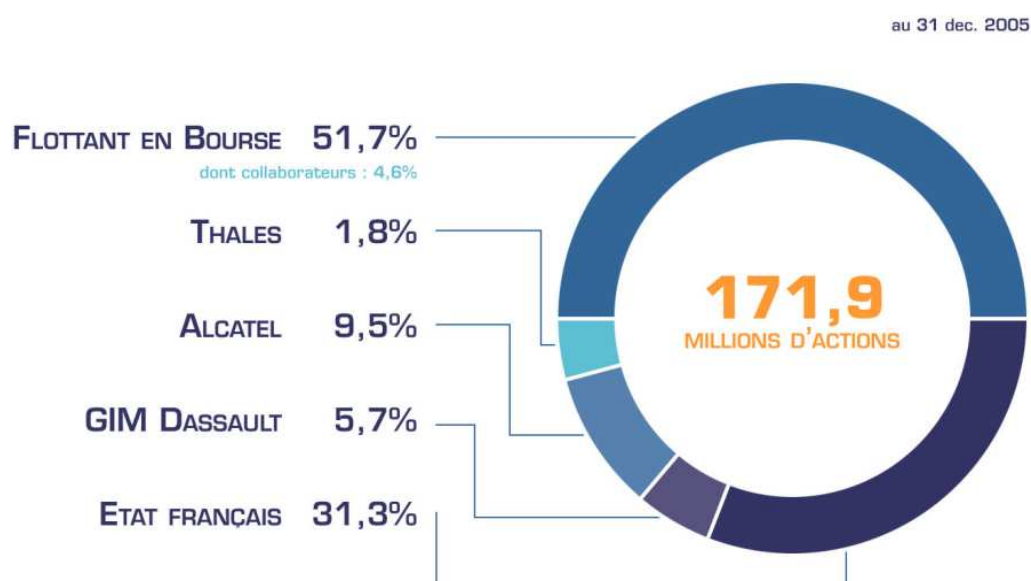


Figure 2.6 : Structure de l'actionnariat du groupe Thales

2.1.4 DIVISION SYSTEMES TERRE ET INTERARMEES (DLJ)

2.1.4.1 GENERALITES

Créée en juin 2004, cette division devient le leader mondial dans les équipements de communication et d'optronique pour les forces terrestres, aériennes et navales et est également un fournisseur de services (associés aux systèmes vendus).

Ces systèmes trouvent leur application sur différents lieux de déploiement et pour différents types d'opérations.

Avec ses 13 000 employés répartis sur une vingtaine de pays, la division réalise en 2005, un bénéfice de 2,4 milliards d'Euros dont 30% a été attribué à la Recherche et au Développement.

2.1.4.2 ACTIVITES

- Maîtrise d'œuvre et intégration de systèmes terrestres (plates-formes, radios, caméras, Internet tactique,...) ;
- Maîtrise d'œuvre et intégration de systèmes Interarmées (radios, équipements de transmission, équipements de navigation, équipements de mise en réseau,...) ;
- Équipements de Communications et Optroniques (laser, infrarouge, traitement du signal, logiciels embarqués, gestion de réseaux, mâts optroniques,...) ;
- Technologies optiques (vision de nuit, objectifs spéciaux et zoom, machines à froid pour caméras thermiques, laser méga joule, ...) ;
- Services.

2.1.4.3 L'UNITE DE COMMUNICATIONS TACTIQUES

L'unité UCT est en charge de la conduite des projets, programmes, produits et systèmes répondant aux besoins de Systèmes de Communications Tactiques pour les Forces Armées Terrestres, de façon directe ou par l'intermédiaire de systémiers.

En complément, UCT fournit des prestations et équipements dans les domaines Durcissement, Instrumentation, Sûreté des Systèmes, au profit de grands clients civils et étatiques, mais aussi des entités de Thales.

Les activités d'UCT couvrent 4 domaines :

- **Radios Tactiques**, avec comme principales lignes de produits : PR4G F@stnet (radio VHF IP, GPS, 64 kb/s), TRC 3700 (Radio HF, Stanag's, 9600 b/s), HCDR (Radio Haut Débit) et T-SRN (Software Radio Node), futures générations de Radio Logicielle Multiservices ;
- **Réseaux Tactiques**, avec comme principales lignes de produits : TRC 4000 (Faisceaux Hertiens à 8 et 34 Mb/s en bande IV et V), familles de *switch*, routeur, IP service, gestion pour les applications réseaux ;

- **Internet Tactique Mobile**, avec comme principales lignes de produits : SLC (Serveur Logiciel de Communication offrant des services de communication multimédia sur réseaux contraints aux applications C4ISR), WLAN (Wireless Lan courte portée/haut débit) ;
- **Durcissement, Instrumentation et Sûreté Des Systèmes** avec une expertise en durcissement électromagnétique (CEM, Tempest, IEM, foudre, ...) et une gamme de systèmes et produits à haute sûreté de fonctionnement.

La quasi totalité des lignes de produits est développée en coopération avec les différentes sociétés Thales en Europe (Royaume-Uni, Norvège, Allemagne, Suisse, Italie, Espagne, Pays-Bas,...).

2.1.4.4 LE SERVICE SOFTWARE RADIO

Dans le cadre des activités de communications tactiques de Thales, le service software radio est chargé du développement de nœuds de communication basés sur le concept de « radio logicielle » (ou software radio) qui permet de rendre indépendant le matériel (modems, cartes, bus) et le logiciel (formes d'onde réalisant des services différents).

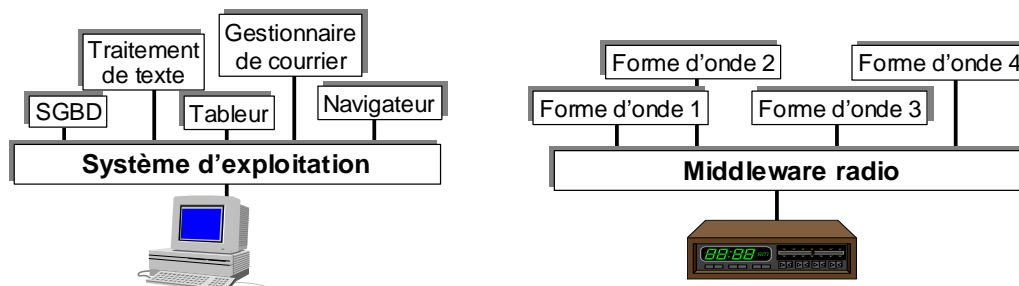


Figure 2.7 : Analogie avec l'univers informatique

Les radios logicielles de nouvelle génération (à canaux multiples, multi-bande et programmables) utilisent une architecture standard et supportent de nouvelles formes d'onde (ensemble des couches protocolaires) qui permettent le déploiement de radiodiffusions mobiles à haut débit. Ces "radios" incorporent aussi des services d'intercommunication entre des réseaux divers, facilitant la mise en oeuvre d'un « réseau de réseaux » et l'utilisation de réseaux radio ad-hoc comme réseau relais entre deux réseaux fixe (par exemple, deux réseaux d'infrastructure de deux nations alliées).

Le développement de ces radios logiciels exige des nombreux développements et avancées techniques dans les domaines suivants : logiciel et réseaux (protocoles de radio, techniques IP, *middleware*, etc), matériel (radios large bande, modems à grande vitesse, intégration plus grande c'est à dire poids, volume et consommation réduits par suite de la miniaturisation et de la performance accrue de composants électroniques) dans une approche de plus en plus standardisée (pour l'architecture, la sécurité, le partage de formes d'onde pour réseau de coalition, etc).

Spécifiquement, les percées techniques de radio logicielle faciliteront le développement de produits plus sécurisés offrant une flexibilité et une évolutivité accrues (puisqu'elles peuvent être reprogrammées).

2.2 L'EVOLUTION DES COMMUNICATIONS TACTIQUES

La partie suivante dresse l'état actuel des communications et fait le points sur les nouveaux besoins en terme de services sur le champ de bataille, en mettant ces demandes en perspective avec les nouvelles capacités apportées par le développement des radios logicielles.

2.2.1 RAPIDE ETAT DE L'EXISTANT

Traditionnellement, les systèmes de communication tactiques sont organisés de manière hiérarchique, et reposent sur trois composants principaux :

- Un cœur de réseau fixe, qui interconnecte le quartier général et les groupes d'un niveau supérieur ou égal à la brigade ;
- Un sous-système CNR (*Combat Net Radio*) pour les communications avec les troupes et les groupes d'un niveau inférieur à la brigade
- Un sous-système local LAS (*Local Area Subsystem*) permettant des communications (données, VoIP) à l'intérieur du quartier général ou entre « abonnés » dans des véhicules lourds à l'arrêt.

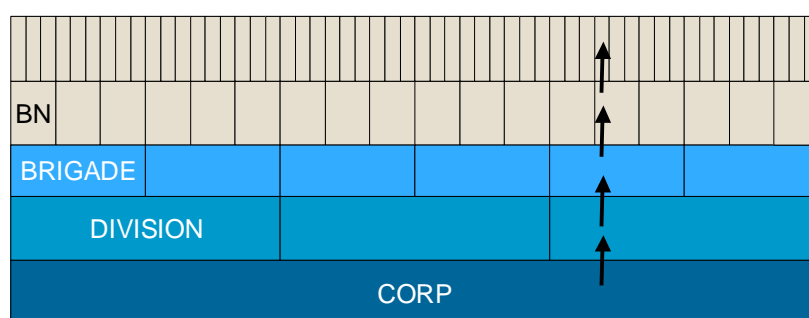


Figure 2.8 : Organisation traditionnelle des communications tactiques

Cette structure de communications par des échelons organisationnels hiérarchiques ne permet que difficilement une circulation rapide des données dans un contexte de conflit info-centré. Dans ce cadre, une structure de communications qui est indépendante de frontières organisationnelles spécifiques est nécessaire car elle permet d'organiser les communications de façon transverse entre éléments de manœuvre qui coopèrent, selon le besoin et la localisation géographique plutôt qu'en fonction de la hiérarchie.

2.2.2 GUERRE INFOCENTREE ET EVOLUTION DES BESOINS

Les techniques de guerre actuelles reposent sur des systèmes d'armement évolués, chacun d'entre eux utilisant ses capteurs propres pour prendre des décisions et déterminer des cibles. Avec le concept de guerre info-centrée, les capteurs et des tireurs sont connectés dans un réseau où les décisions utilisent des informations de positionnement (*situation awareness*) recueillies par plusieurs capteurs proches ou éloignés.

Ce concept suppose un système de communication de grande échelle capable de connectivité homogène à travers un système d'armement et garantissant la transmission de données avec des retards déterminés.

Les communications sur le champ de bataille incluent au moins la voix, des données de messagerie, et des données temps réel. De plus, des transferts de base de données et de vidéos peuvent être réalisés. De fait, la demande en débit et en bande passante augmente de manière permanente.

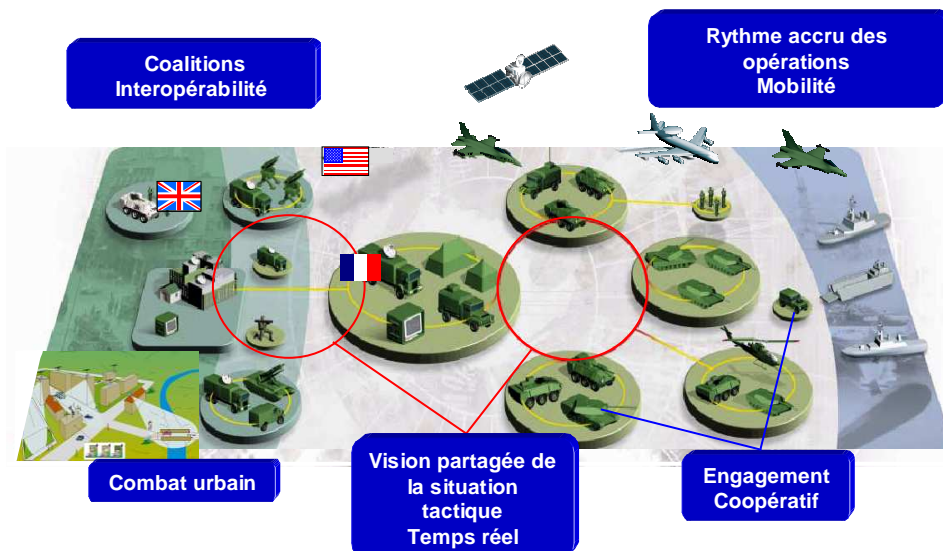


Figure 2.9 : Évolution des besoins de communication

En raison de la nature différente de ces types de transmissions (conversationnel, massivement à sens unique, transactionnel, en temps réel) il est nécessaire :

- d'appliquer des techniques QoS pour optimiser les communications et satisfaire les exigences des différents utilisateurs ;
- d'offrir la gestion de préséance pour des utilisateurs militaires.

De plus, un réseau tactique moderne doit être capable de grandir un déploiement rapide, qu'il soit petit et dispersé ou bien qu'il corresponde à un engagement massif et dense sans longue phase de préparation de mission, c'est à dire simplement en utilisant des données d'initialisation actuelles.

Le réseau de communications doit donc s'auto configurer puisqu'il n'est pas imaginable de compter sur un spécialiste de communications pour exécuter la lourde tâche de configuration du réseau dans le cadre d'un déploiement rapide de ses nœuds.

Ces besoins et exigence nécessitent donc le recours, lors de la conception de la partie logicielle d'une *software radio*, à des formes d'ondes de type ad-hoc (c'est à dire capable de se constituer de manière spontanée et automatique en réseau) avec des capacités haut-débit, et gérant la priorité des flux.

2.3 LE CONCEPT DE RESEAU AD-HOC

Le développement des technologies de communications sans-fil dans le domaine militaire ou civil a conduit à une explosion du nombre de terminaux équipés et au développement d'applications qui ne peuvent plus se satisfaire du simple modèle de communication reposant sur une infrastructure architecturée le plus souvent autour d'un cœur de réseau reposant sur une technologie filaire avec lequel les terminaux se contentent de communiquer par voie radio.

2.3.1 PARADIGME DES RESEAUX AD-HOC MOBILES

Aussi, l'intérêt est d'organiser ces terminaux mobiles en MANET (*mobile ad hoc network*) qui sont des réseaux auto-organisés et auto-configurés, sans-fil et permettant des communications multibonds entre les terminaux et où la topologie du réseau change dynamiquement, au gré de la mobilité de ses membres.

Un MANET permet donc aux terminaux portables d'établir des communications n'importe où et n'importe quand sans recours à une quelconque infrastructure centralisée.

2.3.2 INTERET DES RESEAUX AD-HOC

Les réseaux ad-hoc sont donc très intéressants dans le cadre de communications tactiques du domaine militaire ou de la sécurité. Ils peuvent également jouer un rôle important dans de nombreuses applications civiles comme l'organisation de conférences, de conventions ou de classes électroniques.

Enfin, ils peuvent être la solution pour l'organisation des communications dans un contexte de crise où aucune infrastructure pérenne ne subsiste, comme après une catastrophe naturelle.

2.3.3 PRINCIPAUX ENJEUX

Dans un réseau mobile ad-hoc, les nœuds présents partagent un même canal radio et tentent coopérer pour relayer les données de l'émetteur à la destination. La mise en place d'un tel réseau soulève donc le problème du partage de l'accès au canal et d'un routage efficace des messages au sein d'un réseau dont la topologie évolue dynamiquement. De plus, un tel réseau doit être capable d'assurer la synchronisation et l'auto configuration des éléments qui le constituent.

2.3.3.1 LE PARTAGE DU CANAL DE COMMUNICATION

Pour des raisons tant économiques que de fiabilité, les postes radio utilisés dans le domaine des communications tactiques sont le plus souvent des appareils mono voie, travaillant en *half-duplex* (chaque poste possède une unique chaîne radio qui peut *alternativement* émettre ou recevoir). Cette limitation du matériel impose une gestion correcte de l'accès au canal afin d'utiliser au maximum les capacités de chacun des postes, en fonction du trafic qu'il doit émettre, relayer ou recevoir.

Le fait que plusieurs nœuds doivent communiquer en utilisant le même support de communication (même canal radio) impose de partager, le plus judicieusement possible, son accès entre les membres du réseau. La principale difficulté de cette tâche dans les réseaux MANET vient du fait qu'aucune infrastructure centralisée ou aucune station particulière ne possède une vue globale du réseau permettant d'effectuer facilement cette allocation.

Deux principales méthodes sont utilisées pour partager l'accès au canal :

- Un **accès sans contention** : où soit le temps d'accès est partagé en unités temporelles qui sont réparties entre les stations (TDMA : *time division multiple acces*) et chaque station a la garantie d'être la seule à émettre pendant l'unité de temps qui lui a été allouée, soit le spectre d'émission est partagé permettant une séparation fréquentielle des émissions (FDMA), soit les communications sont séparées par un codage orthogonal (CDMA) ;

- Un **accès avec contention** : où les stations essaient toutes d'émettre dès qu'elles en ont le besoin. Pour éviter les émissions simultanées (qui conduisent à un brouillage des données envoyées sur le canal), des mécanismes d'écoute du canal, de signalisation et d'attente sont mis en place. Parmi ces derniers figurent ALOHA, CSMA/CD et CSMA/CA, qui proposent des mécanismes plus ou moins évolués pour gérer l'accès au canal et détecter ou empêcher les éventuelles collisions.

Le détail de ces différents mécanismes est donné dans les parties 3.2.1.2 et 3.3.2 de ce document.

2.3.3.2 LE ROUTAGE DES DONNEES

Du fait de leur rôle de relais, les nœuds d'un MANET ne fonctionnent plus uniquement comme un terminal mais aussi comme un routeur qui doit donc tenir à jour des routes (les plus optimales possible) vers les autres membres du réseau et relayer les données à destination de nœuds qui peuvent ne pas être dans sa zone de transmission directe.

Le routage de donnée joue donc un rôle central dans le fonctionnement d'un tel réseau. De plus, cette opération doit faire face à la forte mobilité des nœuds, à leur probable grand nombre et aux ressources de communication limitées (bande passante, énergie pour l'émission, etc...).

Les protocoles de routage ad-hoc doivent donc pouvoir s'adapter à des changements de topologie fréquents et imprévisibles tout en limitant le nombre de communications et de calculs nécessaires à leur fonctionnement. Un plus grand aperçu des technologies de routage ad hoc est donné dans la partie 3.1.

3. PROBLEMATIQUES PARTICULIERES DES RESEAUX AD-HOC

3.1 ROUTAGE EN RESEAU AD-HOC

Pour circonvenir aux problèmes levés par le routage de paquets dans les réseaux mobiles ad-hoc et qui ont été décrits dans la partie précédente, de nombreuses solutions ont été élaborées, offrant un panel étendu de techniques avec chacune leurs propriétés différentes.

3.1.1 CLASSIFICATION DES ALGORITHMES

Devant un tel essor d'imagination, diverses techniques de classifications ont été mises en œuvre afin de mieux cerner les fondements et les propriétés de chaque algorithme.

3.1.1.1 VECTEUR DE DISTANCE ET ETAT DU LIEN

Dans le monde filaire, il est d'usage de différencier les protocoles selon qu'ils sont basés sur la technique du vecteur de distance (*distance vector* en anglais) ou bien sur l'état du lien (*link state*).

Dans le premier cas, chaque nœud du réseau envoie à ses voisins une liste des nœuds qu'il est possible d'atteindre par lui (directement ou indirectement) avec la distance (en nombre de sauts ou d'intermédiaires) jusqu'à destination.

Chaque nœud n'a donc en sa possession qu'une topologie partielle du réseau (la distance et la direction vers laquelle se trouvent les autres nœuds). Cet échange de vecteurs de distance converge au bout de quelques cycles (plus aucune nouvelle information n'est échangée) et l'algorithme permet d'obtenir à coup sûr le plus court chemin depuis n'importe quel nœud vers n'importe quelle destination.

Dans le second cas, chaque nœud diffuse, par *broadcast*, à l'ensemble des nœuds du réseau la liste de ses voisins (et donc l'état de ses liens, d'où le nom de ce type de protocoles). A partir des informations collectées par les différents envois, chaque nœud calcule et maintient une topologie complète du réseau (avec les coûts en calcul et en mémoire que cela nécessite). Il peut ainsi à tout instant déduire toutes les routes possibles vers une destination donnée et choisir la meilleure.

3.1.1.2 PROTOCOLES PROACTIFS ET PROTOCOLES REACTIFS

Du fait de la capacité limitée du canal physique et à la mobilité des nœuds qui engendre de nombreux changements de topologie, l'étude des protocoles de routage insiste sur la distinction en protocoles proactifs et réactifs. En effet, ces propriétés conditionnent la manière dont sont envoyés les messages de contrôle destinés à maintenir les informations de routage, ce qui influe directement sur l'*overhead*¹ engendré par le protocole et sur la correction de l'acheminement des paquets.

Le premier type d'algorithme de routage repose sur un échange périodique de messages de contrôle (ces échanges peuvent être locaux, c'est à dire limités au voisinage d'un nœud ou

¹ L'*overhead* est défini comme le rapport entre la quantité d'information échangée pour le contrôle du réseau (c'est à dire tout ce qui n'est pas des données utiles) et le trafic total du réseau.

globaux). Comme cet échange de message a lieu même si aucun changement dans la topologie du réseau n'est intervenu, il génère un trafic de contrôle assez important qui est autant de bande passante perdue pour les données. En contrepartie, comme tous les nœuds ont une image à jour du réseau, le routage est immédiat : lorsqu'un paquet doit être envoyé, ce qui permet d'avoir un temps de latence réduit au minimum.

Les algorithmes réactifs font eux le pari inverse en estimant qu'il est suffisant d'envoyer des messages de contrôle « à la demande » et de n'établir le chemin de routage uniquement lorsqu'il y a des paquets à envoyer ou que la topologie du réseau a changé. Cette technique permet en théorie de limiter l'*overhead* (sauf si les changements sont trop fréquents) au prix d'une plus grande latence dans l'acheminement des paquets puisqu'il faut déterminer la route à leur faire suivre avant chaque envoi.

Tableau 3.1: Classification simplifiée des protocoles de routage ad-hoc

| | Vecteur de distance | État du lien |
|-----------------|---------------------|--------------|
| Proactif | DSDV | OLSR |
| Réactif | AODV | DSR |

3.1.1.3 AUTRES ELEMENTS DE TAXONOMIE

P. Kuosmanen a établi une taxonomie plus détaillée des différents protocoles de routage pour réseaux ad-hoc[5]. Parmi les différents éléments de classification retenus figurent notamment :

Le *modèle de communication* : prenant en compte des communications **mono-canal** (basés pour la plupart sur CSMA/CA) ou **multi-canaux** (qui s'appuient sur des réseaux utilisant TDMA ou CDMA) ;

La *structure* : selon que les protocoles sont **uniformes** (sans hiérarchie entre les nœuds) ou **non-uniformes** (où une hiérarchisation des nœuds permet de réduire la quantité d'informations de contrôle à envoyer).

Pour ces premiers, on distingue les protocoles reposant sur la **topologie** du réseau (où chaque nœud maintient une carte assez précise du réseau) et les protocoles basés sur la **destination** (qui n'ont qu'une connaissance locale du réseau, comme les protocoles à vecteur de distance).

Pour les protocoles non-uniformes, la hiérarchisation s'effectue soit sur la **proximité** (distinction voisins/nœuds éloignés) ou sur un **partitionnement** (avec des nœuds de haut ou bas niveau qui jouent alors un rôle différent).

Le *type de diffusion* : qui définit si les protocoles s'appuient sur la transmission des données vers une destination unique (**unicast**), vers plusieurs nœuds de destination (**multicast**) ou bien vers un groupe de nœuds situés dans une zone géographique précise (**geocast**). Le premier type de protocole est de loin le plus répandu ;

La *fonction de coût* : qui distingue les protocoles basant le calcul du coût d'un lien sur le **nombre sauts**, la **bande-passante**, l'**énergie** nécessaire ou bien encore le **degré de stabilité** des liens (ce qui tend à privilégier les liaisons stables).

A partir de ces critères, l'auteur dresse la taxonomie suivante, le détail n'étant donné que pour les protocoles *unicast* qui sont les plus courant :

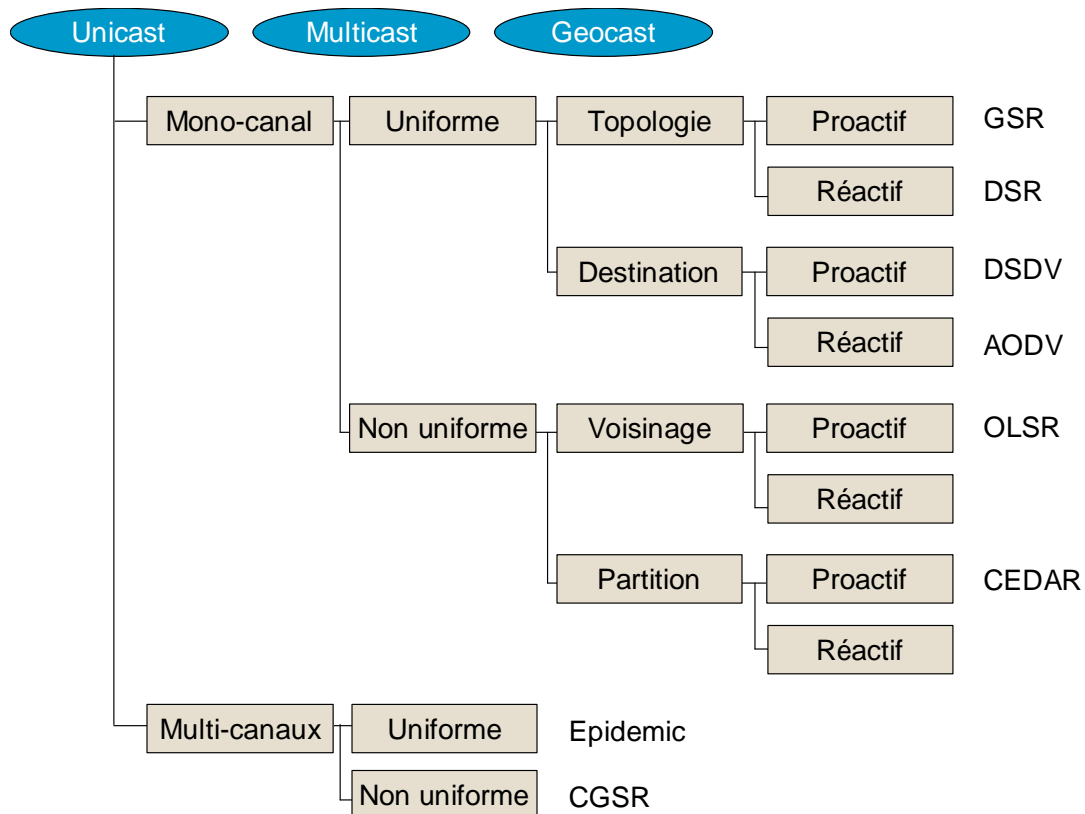


Figure 3.1 : Taxonomie des protocoles de routage

3.1.2 OLSR

OLSR (*Optimized Link State Routing*) est un protocole de routage pour les réseaux ad-hoc développé par l'équipe du projet Hipercom de l'INRIA, dirigée par Philippe Jacquet. OLSR est basé sur l'algorithme de l'état du lien et est proactif puisqu'il emploie des échanges de messages périodiques pour maintenir des informations sur la topologie du réseau en chaque nœud.

OLSR permet d'effectuer les décisions de routage d'une manière distribuée et permet au réseau de s'organiser spontanément. Il prend aussi en compte l'existence de liens uni-directionnels, qui existent souvent du fait de la dissymétrie entre la taille et la puissance des équipements composant un réseau mobile ad-hoc.

3.1.2.1 PRINCIPES D'OLSR

Comme décrit dans la taxonomie présentée en Figure 3.1, OLSR est un protocole à état du lien proactif, et non uniforme puisqu'il s'appuie sur une distinction basée sur le voisinage pour établir une hiérarchisation entre les nœuds.

En fait, OLSR essaie de limiter la quantité de messages de contrôle en optimisant leur nombre et leur diffusion au sein de l'ensemble du réseau. Pour cela, il s'appuie sur l'élection de relais multipoints (ou *MultiPoint Relays, MPRs*) parmi les voisins de chacun des nœuds du réseau qui seront les seuls à relayer les messages de contrôle de leurs électeurs (appelés *multipoint relay selectors*).

La manière dont sont choisis les *MPRs* permet ainsi d'avoir une diffusion optimisée des informations de contrôle sur l'ensemble du réseau, en minimisant l'*overhead* (notamment en évitant, lors des *broadcasts* qu'un nœud ne reçoive le même message de différentes sources).

De plus, l'élection de ce sous-graphe du réseau permet d'assurer une limitation du nombre de messages et de liens annoncés. Dans la configuration standard d'OLSR, seuls les *MPRs* annoncent leurs liens avec leurs électeurs. Cette information étant suffisante pour déterminer une route d'un point à un autre du réseau.

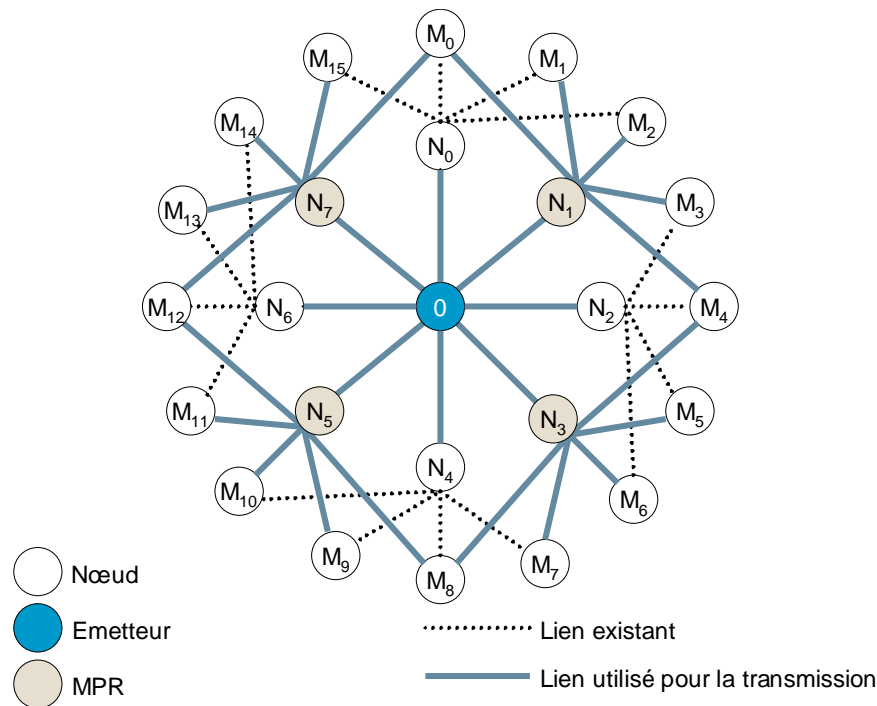


Figure 3.2 : Diffusion de messages avec les MPR

OLSR fonctionne de manière complètement décentralisée et l'échange de message de contrôle supporte un certain nombre de pertes de paquets et ceux-ci peuvent-être reçus dans le désordre (l'ordre des messages étant spécifiée dans un numéro de séquence contenu dans le paquet).

OLSR échange ses messages de contrôle sur le port UDP 698 en utilisant un format de paquet unifié (qui permet l'agrégation de messages ou *piggybacking*) et qui ne requièrent pas de changement au format des paquets IP. Il faut également noter que OLSR, bien que développé pour IPv4 (avec des tailles d'adresses de 32 bits) peut être facilement adapté à IPv6 (simplement en augmentant la taille des champs d'adressage dans les paquets à 128 bits et en augmentant la taille minimale des paquets en conséquence).

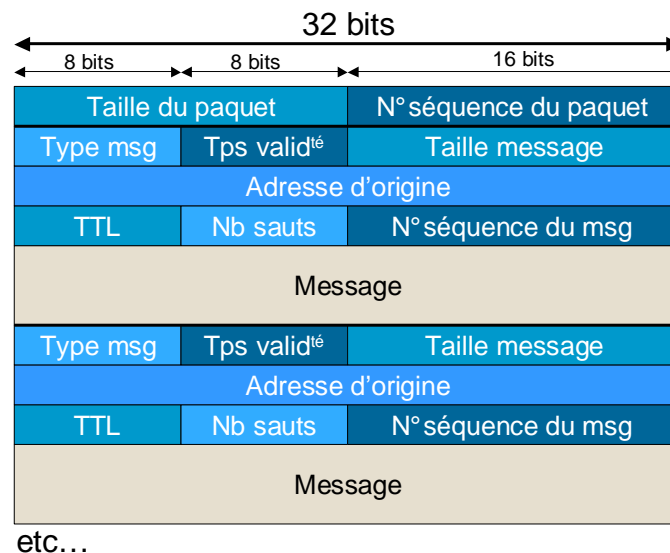


Figure 3.3 : Format du paquet OLSR

Les paragraphes suivants décrivent plus en détail le fonctionnement du protocole. Plus d'informations peuvent être trouvées dans l'article introductif du protocole[1] ou dans la RFC² dont il a fait l'objet[2].

3.1.2.1.1 Bases d'informations

Comme décrit précédemment, OLSR est un protocole proactif, il maintient donc constamment des informations sur la topologie du réseau, et puisque qu'il s'appuie aussi sur l'algorithme d'état du lien, chaque nœud possède une image assez complète de la carte du réseau.

Ceci explique en partie le fait qu'OLSR soit un protocole demandant une assez grande quantité de mémoire puisqu'il nécessite le stockage et la mise à jour de sept bases d'information différentes :

- *Multiple Interface Association Information Base* : chaque nœud du réseau est identifié par une seule et unique adresse par OLSR (appelée **adresse principale**) même s'il possède plusieurs interfaces avec des adresses différentes. Cette table fait le lien entre les interfaces et l'adresse principale à laquelle elles sont rattachées ;
- *Link Set* : maintient la liste et l'état (symétrique, asymétrique) des liens entre les interfaces du nœud et les interfaces voisines ;
- *Neighbor Set* : maintient une liste des voisins (identifiés par leur adresse principale) avec l'état du lien (symétrique, asymétrique) et leur capacité de routage (mesurée par un entier de 0 –aucune volonté– à 7 –plus haute volonté–) ;
- *Two Hop Neighbor Set* : décrit les liens symétriques entre les voisins symétriques et les voisins de distance 2 (tous identifiés par leur adresse principale) ;

² Une *RFC* ou *Request For Comments* est un document standardisé dans lequel le concepteur (ou le groupe de concepteurs) d'un protocole définit de manière précise le fonctionnement de celui-ci. Cette définition est soumise aux commentaires d'autres lecteurs éclairés afin de l'améliorer. Les RFC sont identifiées par un numéro unique (3626 pour OLSR) et gérées par l'*IETF* (*Internet Engineering Task Force*).

- *MPR Set* : Liste des adresses principales des voisins qui ont été choisis comme *multipoint relay selector* ;
- *MPR Selector Set* : maintient la liste des voisins qui ont élu ce nœud comme *MPR* avec un temps d'expiration associé ;
- *Topology Information Base* : stocke des informations sur la topologie du réseau en listant, pour chaque nœud, un l'adresse principale d'un voisin permettant d'y accéder (souvent un MPR du nœud) ainsi qu'un numéro de séquence associé et un instant limite de validité.

3.1.2.1.2 Messages échangés

Les bases d'informations décrites dans la partie précédente sont remplies par l'envoi périodique de messages de contrôle par les nœuds du réseau. Ces messages sont au nombre de trois, et sont détaillés dans ce paragraphe.

3.1.2.1.2.1 Multiple Interface Declaration (MID)

Tous les nœuds possédant des interfaces multiples doivent périodiquement envoyer des messages d'information décrivant leur configuration. Ceci est fait en diffusant dans l'ensemble du réseau les messages MID.

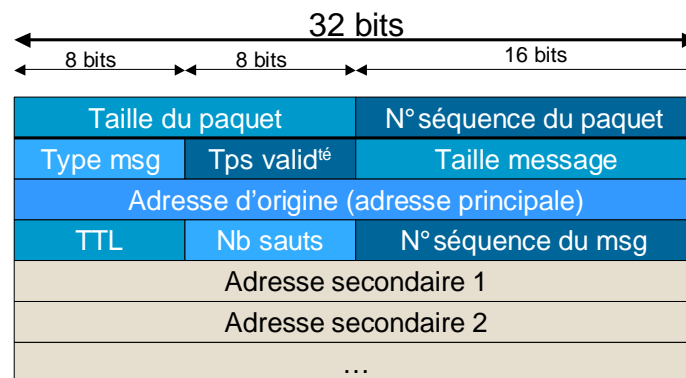


Figure 3.4 : Format du paquet MID

Ces messages contiennent (en plus des en-têtes définis dans la Figure 3.3) la liste des adresses secondaires du nœud. L'adresse principale est celle utilisée dans le champ d'adresse d'origine de l'en-tête. Les messages MID servent à peupler la *Multiple Interface Association Information Base*.

3.1.2.1.2.2 Messages HELLO

L'échange périodique **uniquement entre voisins** des messages HELLO a pour vocation de peupler le Link Set (détection des liens), les Neighbor Set et Two Hop Neighbor Set (détection du voisinage) et le MPR Selector Set (signaler quels voisins ont été élus comme MPR par le nœud courant).

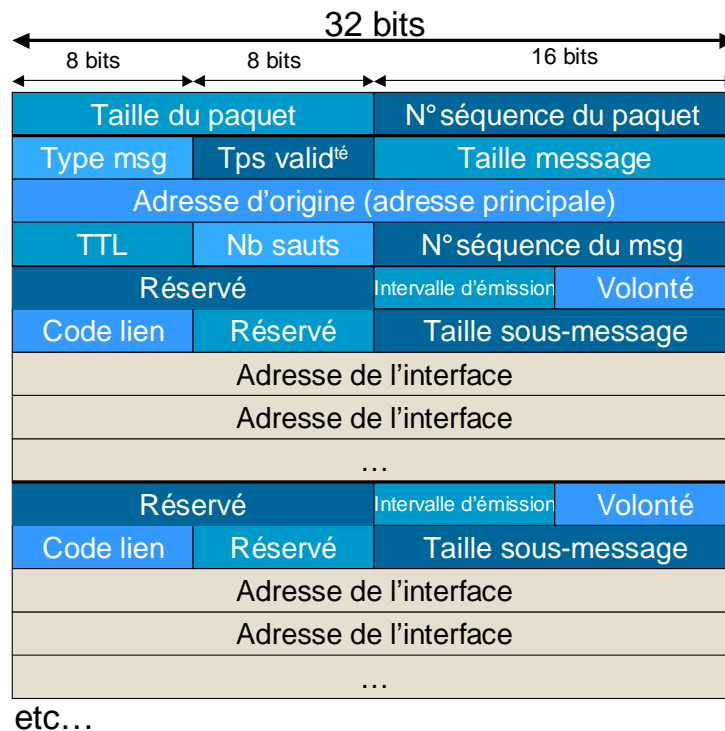


Figure 3.5 : Format du paquet *HELLO*

Les messages *HELLO* transportent en particulier l'intervalle d'émission des messages *HELLO* par le nœud d'origine (qui peut être utilisé pour tester l'état du lien –en comparant la date du dernier message reçu avec cet intervalle), la volonté du nœud d'origine à effectuer les opérations de routage pour la liste d'adresses qui fait suite à ce champ et le code du lien qui permet de spécifier la nature des liens entre le nœud d'origine et la liste d'adresses qui fait suite à ce champ.

Le code lien encode à la fois le type du lien (non-spécifié, asymétrique, symétrique, perdu) et le type du voisin à l'autre bout du lien (symétrique, MPR, non-voisin).

3.1.2.1.2.3 Message TC (Topology Control)

Les messages de contrôle de topologie sont diffusés dans tout le réseau (en utilisant le relais optimisé par les MPR décrit en Figure 3.2). Ils servent à diffuser l'information de topologie le réseau et sont utilisés par les nœuds pour construire les routes.

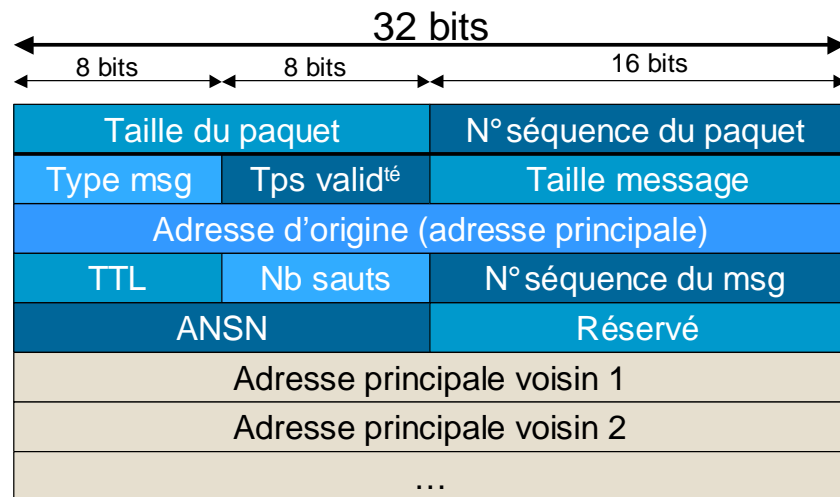


Figure 3.6 : Format du paquet TC

Les messages TC contiennent un numéro de séquence (ANSN) incrémenté par le nœud à chaque changement de topologie constaté. Il permet de vérifier la fraîcheur des informations reçues par un nœud.

Dans ce message figure aussi une liste d'adresse de voisins qui permettent d'atteindre le nœud d'origine et que ce dernier souhaite publier. Selon les recommandations de la RFC3626[2], cet ensemble de nœuds doit au moins être égal à l'ensemble des *multipoint relay selectors* du nœud d'origine du message.

Remarquons qu'il existe un type de message supplémentaire (appelé *HNA* pour *Host and Network Association*) qui permet d'effectuer les opérations de routage entre un réseau ad-hoc et un autre réseau n'implémentant pas le protocole OLSR.

3.1.2.2 FONCTIONNEMENT DU PROTOCOLE OLSR

Le fonctionnement du protocole OLSR peut sembler complexe. Aussi, dans un souci de clarté, il sera abordé par le biais d'un exemple dans lequel sera considéré le réseau présenté en Figure 3.7 et dans lequel tous les liens sont symétriques sauf le lien $N_3 \rightarrow N_6$ qui n'existe pas dans le sens opposé.

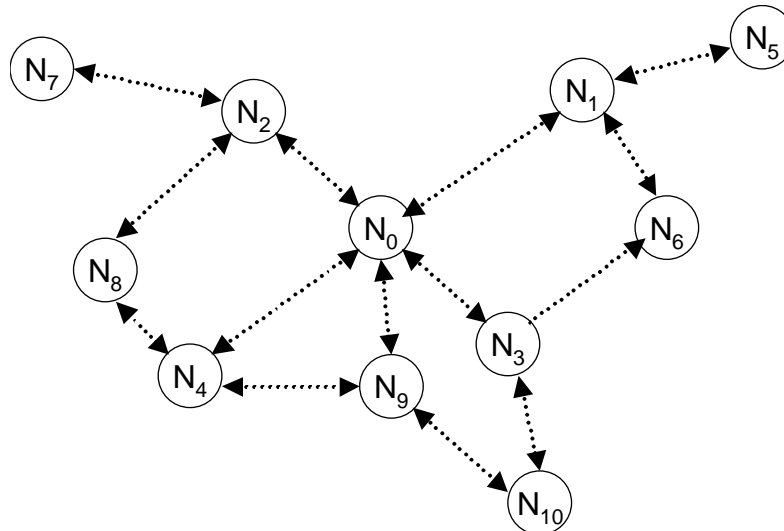


Figure 3.7 : Exemple de réseau ad-hoc

Dans un premier temps, les nœuds disposant de plusieurs interfaces doivent broadcaster leurs messages MID afin de donner aux autres nœuds des informations sur leur configuration. Pour plus de simplicité, nous estimerons ici que tous les nœuds n'ont qu'une seule et unique interface que nous identifierons par N_i , i étant l'indice du nœud.

3.1.2.2.1 Détection du voisinage

La détection du voisinage se fait par l'envoi périodique et l'écoute des messages *HELLO*. Rappelons que ces messages ne sont pas retransmis par les voisins. A l'aide de ces messages, un nœud peut donc apprendre :

- L'adresse principale de l'ensemble de ses voisins (à partir du champ *adresse d'origine* des messages reçus) ;
- La liste des voisins à deux bonds (voisin des voisins) et le type de liens entre ses voisins et les voisins à deux bonds (à partir du champ *code lien* et de la liste d'adresses associées).

Comme aucune information n'est supposée disponible sur l'état des liens entre le nœud N_0 et ses voisins, la méthode suivante est utilisée :

- N_0 , supposé arrivant dans le réseau, envoie un message *HELLO* vide pour l'instant.
- En entendant ce message, N_1 sait qu'il existe un lien $N_0 \rightarrow N_1$ qu'il insère dans sa *Link Set* et qui figure dans le prochain *HELLO* qu'envoie N_1 .
- En recevant ce message, N_0 sait qu'il existe un lien symétrique avec N_1 , information qu'il insère dans sa *Link Set* et qui figure dans le prochain *HELLO* qu'il envoie.
- En recevant ce message, N_1 sait qu'il existe un lien symétrique avec N_0 .

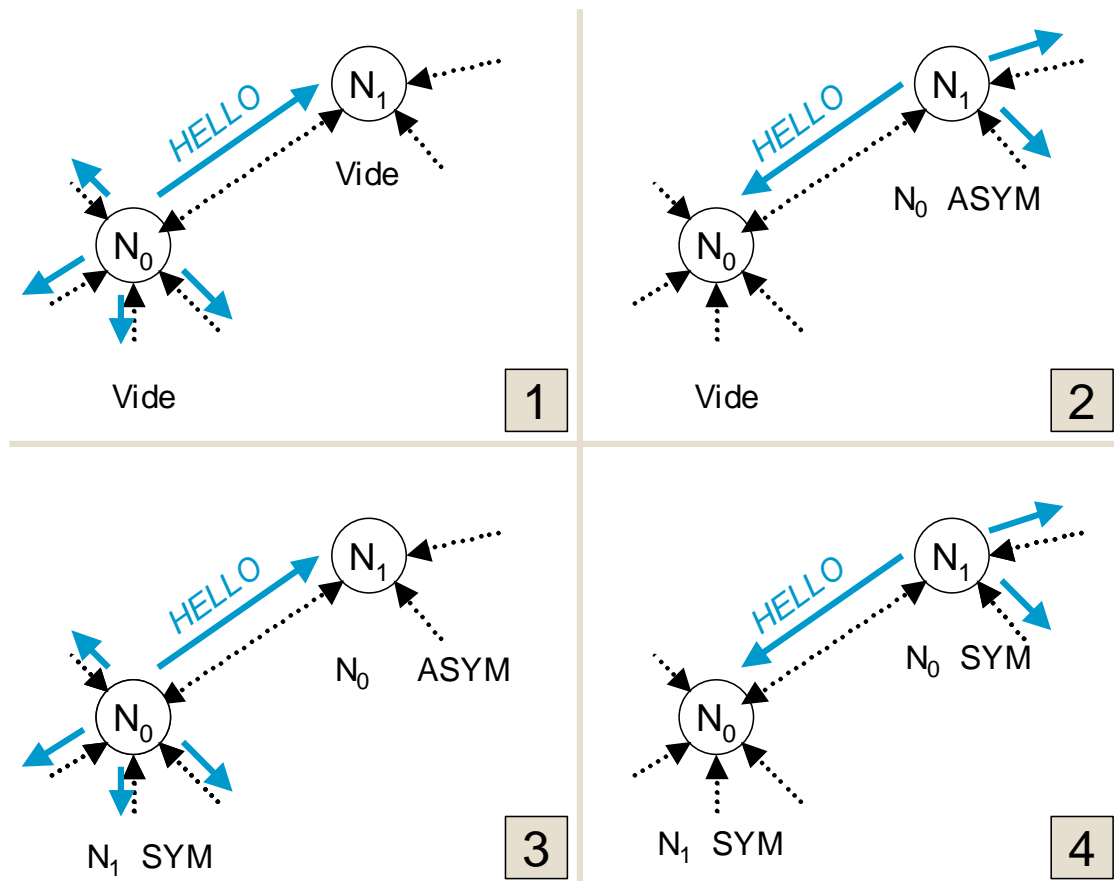


Figure 3.8 : Découverte de voisinage par échange de messages **HELLO**

Après plusieurs échanges, chaque nœud a une image correcte de son voisinage. Il peut donc procéder au choix de ses *MPRs*.

3.1.2.2.2 Élection des *MPRs*

Chaque nœud choisit, de manière indépendante son propre ensemble de *MPRs* (noté $MPR(N_i)$) parmi ses voisins avec lesquels il a un lien symétrique. Pour plus de commodité, on notera $Vois(N_i)$ cet ensemble.

L'ensemble $MPR(N_0)$ est calculé de manière à ce qu'à partir de cet ensemble, il soit possible d'atteindre tous les voisins à deux bonds de N_0 qui ont un lien symétrique avec un élément de $Vois(N_0)$. L'ensemble de ces voisins à deux bonds sera noté $Vois2(N_0)$.

Notons que $MPR(N_i)$ est calculé interface par interface et que l'ensemble des *MPRs* d'un nœud est l'union des ensembles des *MPRs* de chacune de ses interfaces. De plus, plus $MPR(N_i)$ est petit, plus l'*overhead* est minimisé.

Le choix des *MPRs* dépend également de la volonté des voisins à effectuer le routage. Ainsi, un nœud avec la volonté la plus basse (*WILL_NEVER*) ne peut pas devenir *MPR*.

L'algorithme de calcul des MPRs, plus amplement décrit dans la RFC 3626[2], repose sur les principes suivants :

- Tous les voisins symétriques d'un nœud dont la volonté à router est maximale sont choisis comme *MPR* ;
- Tous les voisins symétriques étant les seul à permettre d'atteindre par un lien symétrique un nœud de $Vois2(N_i)$ sont ajoutés à la liste des *MPRs* ;
- Tant que tous les nœuds de $Vois2(N_i)$ ne sont pas couverts par au moins un *MPR*, le voisin de plus haute volonté à router, ou couvrant le plus de nœuds de $Vois2(N_i)$, ou de plus grand degré³ est élu *MPR*.

L'application de cette méthode de calcul au nœud N_0 donne le résultat décrit en Figure 3.9.

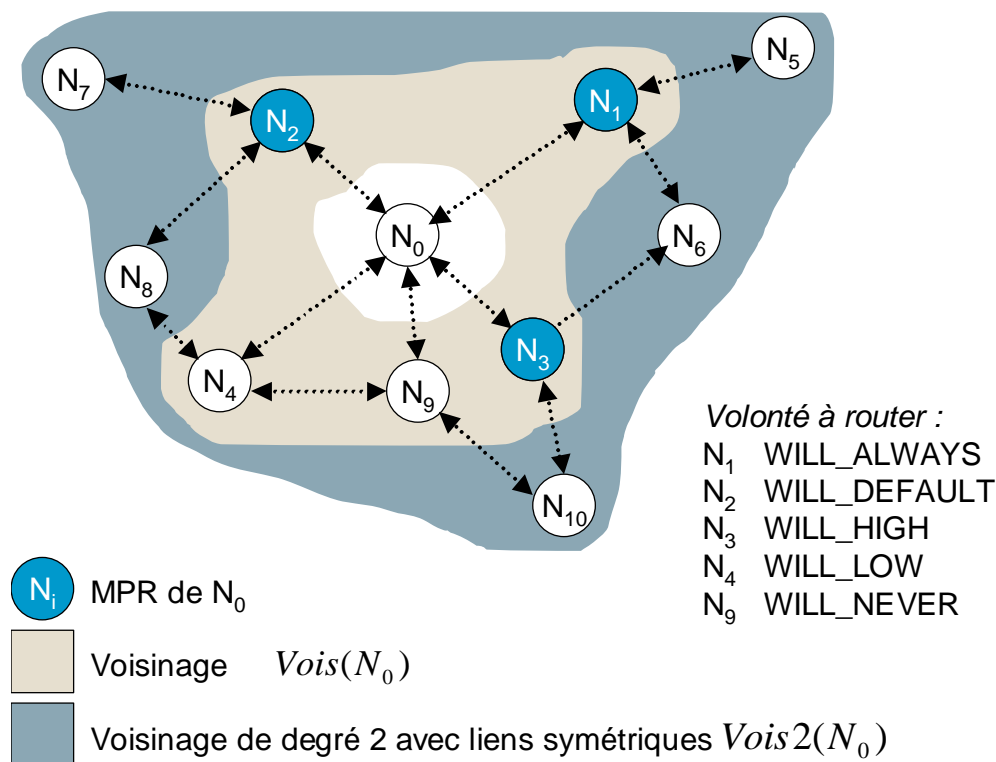


Figure 3.9 : Réseau après choix des MPRs par N_0

Ce mécanisme est appliqué par toutes les stations, ce qui permet au final d'obtenir un ensemble de *MPRs* connexe dans le réseau. Cet ensemble permet de fournir un service de broadcast réseau optimisé. Les paquets en unicast utilisent les tables de routage dont le calcul est décrit dans le paragraphe suivant.

³ Le degré d'un nœud est défini comme le nombre de ses voisins à un bond (voisins en portée directe).

3.1.2.2.3 Calcul des tables de routage

Le sous-réseau constitué de l'ensemble des *MPRs* élus par chacun des nœuds du réseau est un graphe connecté. Une « chaîne » de ces *MPRs* offre un chemin minimal entre n'importe quelle paire source/destination du réseau. Aussi, il n'est pas nécessaire d'annoncer tous les liens pour obtenir une table de routage complète (c'est à dire couvrant l'ensemble des paires source/destination du réseau).

Après le calcul de ses *MPRs*, chaque nœud peut envoyer, en utilisant le mécanisme de diffusion optimisé, une liste de voisins qu'il souhaite publier (au moins la liste de ses *multipoint relay selectors*) aux autres nœuds dans un message TC.

A la réception de ces messages, chaque nœud peut stocker un nœud et un ensemble de chemins y menant et ainsi construire, saut par saut, la route vers une destination.

3.1.2.3 SIMULATIONS ET EXPERIMENTATIONS D'OLSR

Cette partie décrit des résultats intéressants sur le fonctionnement, les performances et les optimisations possibles du protocole OLSR obtenus à partir de simulations ou d'expérimentations.

3.1.2.3.1 Simulations d'OLSR avec 802.11

L'équipe à l'origine du protocole OLSR a cherché à tester ses performances en utilisant un réseau 802.11 (standard de l'IEEE pour les réseaux sans-fil décrit dans la section 3.2.1)[3]. Le modèle de simulation décrit dans le paragraphe suivant.

3.1.2.3.1.1 Modèle de simulation

Pour simuler le comportement d'OLSR sur une couche physique et MAC conforme aux standards de l'IEEE, les auteurs ont retenu un système à séquence directe (D.S.) avec des paquets broadcastés à 1 Mbit/s et des connections point à point à 11 Mbits/s (802.11g). La couche MAC simule le comportement de CSMA/CA avec uniquement quelques simplifications quant à la confirmation des messages RTS/CTS (voir paragraphe 3.2.1.2.2).

Le modèle physique fait l'hypothèse d'une superposition linéaire des signaux envoyés par les émetteurs potentiels. Ainsi, la puissance du signal reçu à un instant donné par le nœud i est donné par la formule suivante :

$$Puiss(i) = \sum_{\substack{j=1 \\ j \neq i}}^n a_j cs_{i,j}, \text{ (eq. 3.1)}$$

où $a_j = 1$ si le nœud j transmet et $a_j = 0$ sinon,

$$cs_{i,j} = \frac{P_j}{r_{i,j}^\alpha} \text{ avec } P_j \text{ la puissance d'émission de } j$$

et $r_{i,j}$ la distance entre i et j , α étant un facteur compris entre 2 et 6.

Pour simuler les paramètres de la couche MAC, trois variables ont été introduites :

- *carriersenselevel*, qui est le niveau de signal à partir duquel le canal est considéré comme occupé ;
- *datalevel*, puissance nécessaire pour transmettre un signal avec succès ;
- *capturelevel*, rapport signal à bruit minimal pour pouvoir décoder une transmission avec succès.

Enfin, différentes topologies de réseau ont été simulées à l'aide de l'outil OPNET (connectivité presque parfaite, grille presque parfaite, alignement sur une large bande) pour des réseaux allant de 50 à 100 nœuds avec ou sans mobilité. Les variables ont été ajustées de manière à ce que la portée de détection du canal (*carrier sense*) soit égale à deux fois la portée de transmission.

Le trafic généré est un trafic de Poisson dont chaque paquet (de 8192 bits de long) a une destination aléatoire.

3.1.2.3.1.2 Résultats de simulation

Les différentes simulations montrent des résultats intéressants :

- L'*overhead* engendré par l'échange des messages *HELLO* et *TC* est assez pénalisant, notamment dans le cas de réseaux avec une topologie complètement connectée (l'*overhead* augmente alors respectivement en n et en n^2 , n étant le nombre de nœuds du réseau, car ils contiennent une liste plus importante de voisins et les messages *TC* sont retransmis plus souvent).
- L'optimisation de la diffusion des messages en utilisant les *MPRs* semble assez efficace et permet de réduire d'un facteur 5 à 10 l'*overhead* généré par ces messages.
- Dans les simulations effectuées, même avec une faible charge du réseau et sans mobilité, la disponibilité des routes n'atteint jamais 100% . Ceci est dû aux nombreuses collisions générées par le trafic en *broadcast* et par les insuffisances de la couche MAC de 802.11 en la matière (notamment dans le cas de configuration en *nœud masqué*).
- L'introduction de gigue dans l'envoi des paquets en *broadcast* (paquets *TC* notamment) permet de corriger cet effet.
- L'utilisation des *MPRs* engendre une disponibilité des routes légèrement inférieure à faible charge mais s'avère plus performante quand la charge du réseau devient importante, et ce avec ou sans mobilité des nœuds.
- La mobilité des nœuds entraîne une perte d'environ 10% de la bande passante. En particulier, dans le cas de la disparition d'une route (due au mouvement d'un ou de plusieurs nœuds), la couche MAC tente de renvoyer le paquet un certain nombre de fois (16 de manière standard). Réduire ce nombre d'essais inutiles permet d'améliorer les performances.

3.1.2.4 OPTIMISATIONS D'OLSR

En se basant à la fois sur des expérimentations et des résultats issus de la simulation, T. Clausen et son équipe proposent diverses améliorations possibles au protocole OLSR[4].

3.1.2.4.1 Conditions d'expérimentation et de simulation

Les conclusions des auteurs s'appuient à la fois sur des données expérimentales (obtenue en utilisant des ordinateurs portables équipés de cartes 802.11 dans différents scénarios dans des réseaux allant de 3 à 10 nœuds) et des données issues de simulations (obtenues en utilisant NS2 pour générer un grand nombre de scénarios aléatoires mais répondant à certaines caractéristiques en termes de dimensions, 1000x1000 mètres, de nombre de nœuds, 50, de vitesse de mouvement, de 1 à 5 m/s).

3.1.2.4.2 Résultats expérimentaux et issus de la simulation

Au vu des résultats expérimentaux et issus de la simulation, les auteurs proposent diverses améliorations quant au fonctionnement du protocole OLSR :

L'introduction de gigue dans l'émission des paquets TC permet de réduire fortement la collision des paquets (qui étaient alors envoyés de manière synchrone par tous les nœuds) et d'augmenter la stabilité des routes. Ce résultat est valable aussi bien en simulation que de manière expérimentale.

L'agrégation de paquets (*piggybacking*) permet de réduire le nombre de cycles d'accès au média et donc d'augmenter la capacité du réseau et d'en diminuer la charge. Cette propriété semble avoir peu d'impact lors des simulations

Augmenter les exigences pour déclarer un lien symétrique, revient à mieux choisir les *MPRs*, en privilégiant les routes les plus stables. Les auteurs proposent d'attendre l'échange de plusieurs messages *HELLO* avant de déclarer un lien symétrique, ce qui rend le réseau plus résistant au prix d'une réponse plus lente aux changements dans les liens et donc à la mobilité des nœuds.

3.1.2.5 FAST-OLSR

Comme semble l'indiquer la description du protocole OLSR, la mobilité des nœuds est grandement limitée par la fréquence d'envoi des messages de contrôle. Et, ainsi que nous l'avons vu dans le paragraphe précédent, il pourrait être profitable d'attendre l'échange de plusieurs messages de contrôle (messages *HELLO* notamment) avant de conclure quant à la validité d'un lien.

Tableau 3.2 : Intervalles d'émission standard du protocole OLSR

| Désignation | Durée |
|------------------|-------|
| HELLO_INTERVAL | 2 s |
| REFRESH_INTERVAL | 2 s |
| TC_INTERVAL | 5 s |

Aussi, pour maintenir des informations correctes avec des nœuds se mouvant à grande vitesse, comme des véhicules motorisés, il semble nécessaire de raccourcir les délais d'émission de ces messages au prix d'un *overhead* plus important.

Aussi, une modification du protocole OLSR permettant un routage correct vers des nœuds très mobile a été proposé[7][8]. Il propose d'adapter la fréquence des messages de contrôle en fonction de la mobilité du nœud tout en limitant la bande passante consommée.

L'intérêt de ce protocole est qu'il reste compatible avec la version standard d'OLSR, puisque des nœuds implémentant OLSR et Fast-OLSR peuvent communiquer. De plus, tout comme OLSR, il ne fait aucune hypothèse sur la couche MAC choisie et les informations qu'elle pourrait transmettre.

3.1.2.5.1 Description du protocole

Fast-OLSR permet à un nœud très mobile de découvrir rapidement ses voisins et de choisir rapidement un nombre limité de *MPRs* en établissant un nombre limité de liens symétriques, mais rafraîchis plus souvent en envoyant des messages appelés *Fast-HELLO* (identiques aux messages *HELLO* mais avec un nombre limité d'adresses) à une fréquence élevée ($\text{FAST_HELLO_INTERVAL} = \text{HELLO_INTERVAL}/10$).

Pour ce faire, Fast-OLSR propose trois mécanismes différents :

- *Passage en mode normal/rapide* : un nœud détectant de nombreux changements dans le réseau (dû à sa grande mobilité) passe du mode normal au mode rapide et envoie des messages *Fast-HELLO* ;
- *Etablissement de liens rapides* : Les nœuds voisins répondent au nœud très mobile avec des messages *Fast-HELLO* similaires. Le nœud mobile choisit, parmi ses voisins symétriques **en mode normal**, un **nombre limité** de *MPRs* (dont la valeur est fixée par le paramètre *Max-Default-MPR* du protocole) en utilisant le même algorithme que le protocole OLSR, les autres sont stockés comme *candidats* ;
- *Rafraîchissement des liens rapides/détection de liens brisés* : les *MPRs* du nœud très mobile répondent avec des *Fast-HELLO* vides (ce qui est suffisant pour maintenir l'état du lien et économise de la bande passante). Si aucun *Fast-HELLO* n'est reçu pendant trois $\text{FAST_HELLO_INTERVAL}$, le lien est déclaré comme perdu et le nœud très mobile élit en remplacement le candidat le plus récent comme *MPR*, ce qui simplifie et rend plus rapide le rétablissement du lien.

3.1.2.5.2 Analyse des performances de Fast-OLSR

De manière théorique, puisque l'intervalle d'émission des messages de contrôle est plus court et que le processus de remplacement des *MPRs* est simplifié, Fast-OLSR devrait fournir des performances au moins équivalentes .

3.1.2.5.2.1 Conditions de simulation

Pour la simulation du fonctionnement de Fast-OLSR, s'appuyant sur la couche physique et MAC de 802.11, le scénario suivant a été retenu.

Un nœud central C communique par des liens symétriques avec six nœuds notés de m_1 à m_6 . Le nœud m se déplace à vitesse constante (fixée selon l'expérimentation) et de manière circulaire autour de m_1 et m_6 et effectue neuf aller-retour. Les conditions sont telles qu'il n'y a pas de superposition de la couverture radio entre deux nœuds adjacents. Le nœud m ne pourra donc pas être simultanément à portée des différents nœuds.

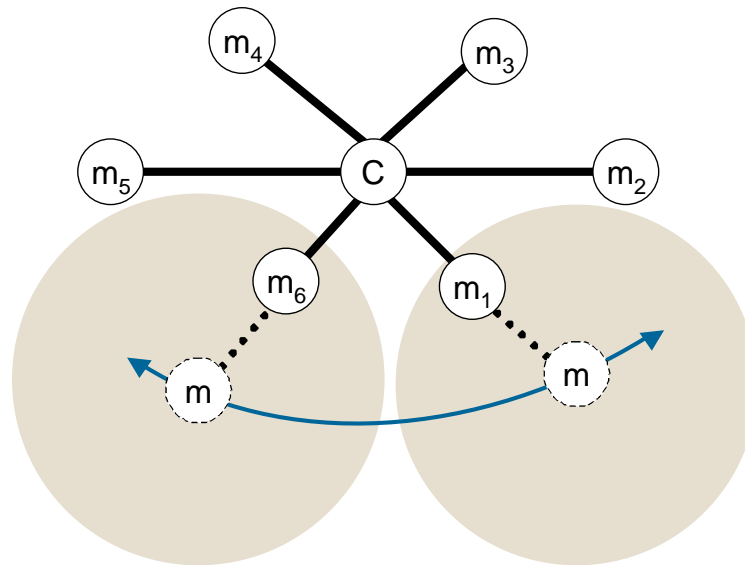


Figure 3.10 : Modèle de simulation de Fast-OLSR

Un trafic à débit constant (*Constant Bit Rate*, ou *CBR*) part de *C* à destination de *m*. Sont mesurés la quantité de paquets perdus durant l'expérience, l'*overhead* introduit par les messages *Fast-HELLO* et *TC*. Les différents paramètres de l'expérience sont la vitesse du nœud *m* et la fréquence d'envoi des messages *Fast-HELLO*.

3.1.2.5.2.2 Résultats de simulation

L'analyse des résultats porte sur des vitesses allant de 20km/h à 150km/h (soit de 5m/s à 40m/s à comparer avec les vitesses des expériences sur OLSR).

Grâce à Fast-OLSR, il est possible de limiter la perte de paquet lors d'un changement de nœud à portée à une valeur d'environ 10% pour un *overhead* d'environ 7.7kbits avec un intervalle d'émission de 100ms. Plus la vitesse du nœud augmente, plus le taux de perte de paquet augmente, et donc plus il est nécessaire de réduire l'intervalle d'émission des messages de contrôle, au prix d'un surcoût en *overhead*.

Une donnée qui aurait pu être intéressante mais qui ne figure pas est la comparaison entre les performances en terme de débit de Fast-OLSR et celle du protocole original.

Il reste aussi à voir dans quelle mesure les résultats présentés ici peuvent être transposés dans des cas où la portée de transmission des nœuds est accrue (par exemple plusieurs kilomètres contre les 60 mètres utilisés dans cette simulation), ce qui a un fort impact sur les temps de propagation et rend difficiles l'utilisation de temporisations aussi petites entre différents messages de contrôle.

3.2 IEEE 802.11 DANS LES RESEAUX MOBILES AD-HOC

La partie précédente a décrit les enjeux du routage de paquets dans le cadre de réseaux sans-fil ad-hoc à nœuds mobiles et a présenté en détails OLSR un des protocoles permettant de réussir cette tâche (parmi les nombreuses solutions possibles).

Lors des différents tests et simulations, qui reposaient tous sur l'utilisation de la norme 802.11 qui est aujourd'hui la plus répandue en matière de réseau sans-fil en mode paquet dans le monde civil, il a été montré que de nombreuses limitations dans les performances des algorithmes de routage venaient en fait de limitations des couches inférieures (notamment de la couche MAC) de 802.11.

Aussi, cette partie, après avoir décrit les différents principes de fonctionnement ce standard, détaille ses insuffisances lors de l'utilisation avec des réseaux sans-fil multibonds et propose diverses améliorations afin d'y palier.

3.2.1 LE STANDARD 802.11 DE L'IEEE

Le protocole 802.11 a été présenté par l'IEEE en 1999 et s'est imposé comme le standard en matière de communications sans-fil en mode paquet dans le monde civil.

Ce standard couvre uniquement la couche physique (PHY) et la couche d'accès au média (MAC, pour *medium access control*). Comme les autres standards 802.X.

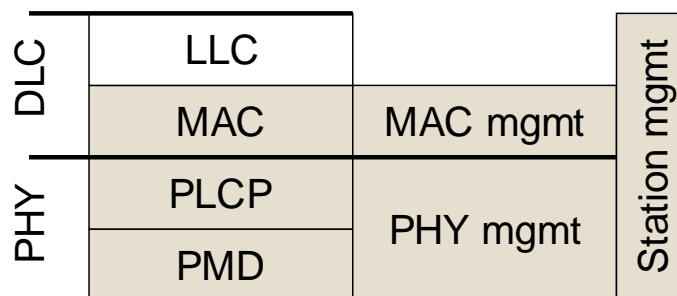


Figure 3.11 : Architecture protocolaire de 802.11

La couche physique est subdivisée en deux parties :

- La *couche protocolaire de convergence physique* (PLCP) qui fournit un signal de contrôle du média (CCA, pour *clear channel assesment*) et fournit des points d'accès physiques indépendants du médium utilisé ;
- La *sous-couche dépendante du médium physique* (PMD) qui gère la modulation et l'encodage/décodage du signal.

La couche d'accès au média gère quant à elle les mécanismes d'association et de ré-association des stations et contrôle l'authentification, le cryptage, la synchronisation et la gestion de l'énergie des différentes stations.

3.2.1.1 LA COUCHE PHYSIQUE

Le standard original de 802.11 propose trois différentes couches physiques :

- *Frequency Hopping Spread Spectrum (FHSS)* : qui utilise différentes fréquences pour les différents canaux. Le standard spécifie l'utilisation de GFSK (*Gaussian Frequency Shift Keying*) de niveau 2 (à 1Mbit/s) ou 4 (à 2 Mbits/s)
- *Direct sequence spread spectrum (DSSS)* : où les canaux sont différenciés par l'utilisation de différents codes utilisant la séquence de Barker. Différents algorithmes sont utilisés pour les sauts de phase, binaire DBPSK (à 1Mbit/s) ou en quadrature DQPSK (à 2Mbits/s).
- *Infra rouge* : qui devait permettre l'utilisation de liaisons infra rouge même sans ligne de vue et en lumière diffuse. Cette partie du standard n'a jamais été utilisée.

Notons que depuis la publication de ce standard sont apparues des variantes de la couche physique, notamment afin d'accroître la portée et le débit des communications :

- *802.11a* : basé sur la couche DSSS qui utilise aussi OFDM pour accroître le débit (54 Mbits/s);
- *802.11b* : basé sur la couche DSSS et qui introduit un troisième algorithme pour le code des sauts de phases, CCK pour *complementary code keying*, pour les débits de 5,5 et 11Mbits/s ;
- *802.11g* : qui utilise également OFDM mais dans la même bande de fréquence que 802.11b pour assurer une retro compatibilité tout en fournissant un débit de 54 Mbits/s ;
- *802.11n* : qui permet d'atteindre des débits supérieurs à 100 Mbits/s en améliorant le débit de crête (notamment grâce à l'utilisation de canaux de 40 Mhz au lieu de 20 Mhz actuels), en utilisant des flux spatiaux multiplexés sur la liaison radio (*MIMO : multiple input – multiple output*) et en améliorant le fonctionnement de la couche MAC (notamment par l'introduction d'agrégation de trames et accusés de réceptions de blocs de trames)[10].

3.2.1.2 LA COUCHE D'ACCES AU MEDIA

La couche d'accès au média (aussi appelée DFWMAC pour *distributed foundation wireless medium access control*) rend non-seulement des services d'accès au médium, mais aussi assure le support du *roaming* (passage d'un point d'accès à un autre), de l'authentification, et de la gestion de l'énergie.

Pour assurer la distribution de l'accès au médium, le standard 802.11 définit deux méthodes. La première, appelée *asynchronous data service*, est obligatoirement implémentée et assure un accès au média avec contention. La seconde, *time-bounded service*, est facultative et propose un accès sans contention.

L'*asynchronous data service* est implémenté en utilisant CSMA/CA (*carrier sense multiple access with collision avoidance*) ou MACA (*multiple access with collision avoidance*) qui est une extension optionnelle de CSMA/CA. Ces deux méthodes sont des fonctions de coordination distribuées (DCF, pour *distributed coordination function*).

Le *time-bounded service* est quant à lui implémenté avec une fonction de coordination centralisée (PCF, pour *point coordination function*) par interrogation des nœuds.

Pour fonctionner, la couche MAC de 802.11 introduit une durée de base, variable suivant la technologie utilisée, appelée *slot* et dont dépendent les autres temps caractéristiques des différents algorithmes.

Parmi ces paramètres figurent :

- *SIFS (short interframe spacing)* : qui est le temps d'attente entre la fin de la réception d'un paquet et l'envoi d'une réponse à ce dernier;
- *PIFS (PCF interframe sapcing)* : qui est le temps maximal que peut mettre un nœud à répondre à une interrogation du nœud maître, dans l'algorithme PCF ;
- *DIFS (DCF interframe spacing)* : qui est le temps minimal pendant lequel le canal doit être libre avant qu'un nœud n'initie un envoi de paquet.

De la taille de ces différents temps caractéristiques dépend la priorité des messages associés. Plus le temps est court, plus le message est prioritaire. Les valeurs standards de ces différents paramètres sont précisées dans le Tableau 3.3.

Tableau 3.3 : Paramètres du standard 802.11

| Paramètre | 802.11 (FHSS) | 802.11 (DSSS) | 802.11 (IR) | 802.11b | 802.11a (g) |
|----------------------------|------------------------------|---------------|--|--------------|------------------|
| t_{slot} | 50 μ sec | 20 μ sec | 8 μ sec | 20 μ sec | 9 μ sec |
| <i>SIFS</i> | 28 μ sec | 10 μ sec | 10 μ sec | 10 μ sec | 16 μ sec |
| <i>PIFS</i> | $SIFS + t_{slot}$ | | | | |
| <i>DIFS</i> | $SIFS + (2 \times t_{slot})$ | | | | |
| <i>Operating frequency</i> | 2,4Ghz | 2,4Ghz | $3,5 \cdot 10^{14}$ Hz (850-950 nm) | 2,4Ghz | 5Ghz (2,4Ghz) |
| <i>Maximum data rate</i> | 2Mbit/s | 2Mbits/s | 2Mbits/s | 11Mbits/s | 54Mbit/s |
| CW_{min} | 15 | 31 | 63 | 31 | 15 |
| CW_{max} | 1023 | 1023 | 1023 | 1023 | 1023 |

3.2.1.2.1 CSMA/CA

L'algorithme CSMA/CA est obligatoirement implémenté et il permet un mode d'accès avec contention au média. Le fonctionnement de l'algorithme est décrit dans ce paragraphe et s'appuie sur la Figure 3.12 qui illustre un scénario possible.

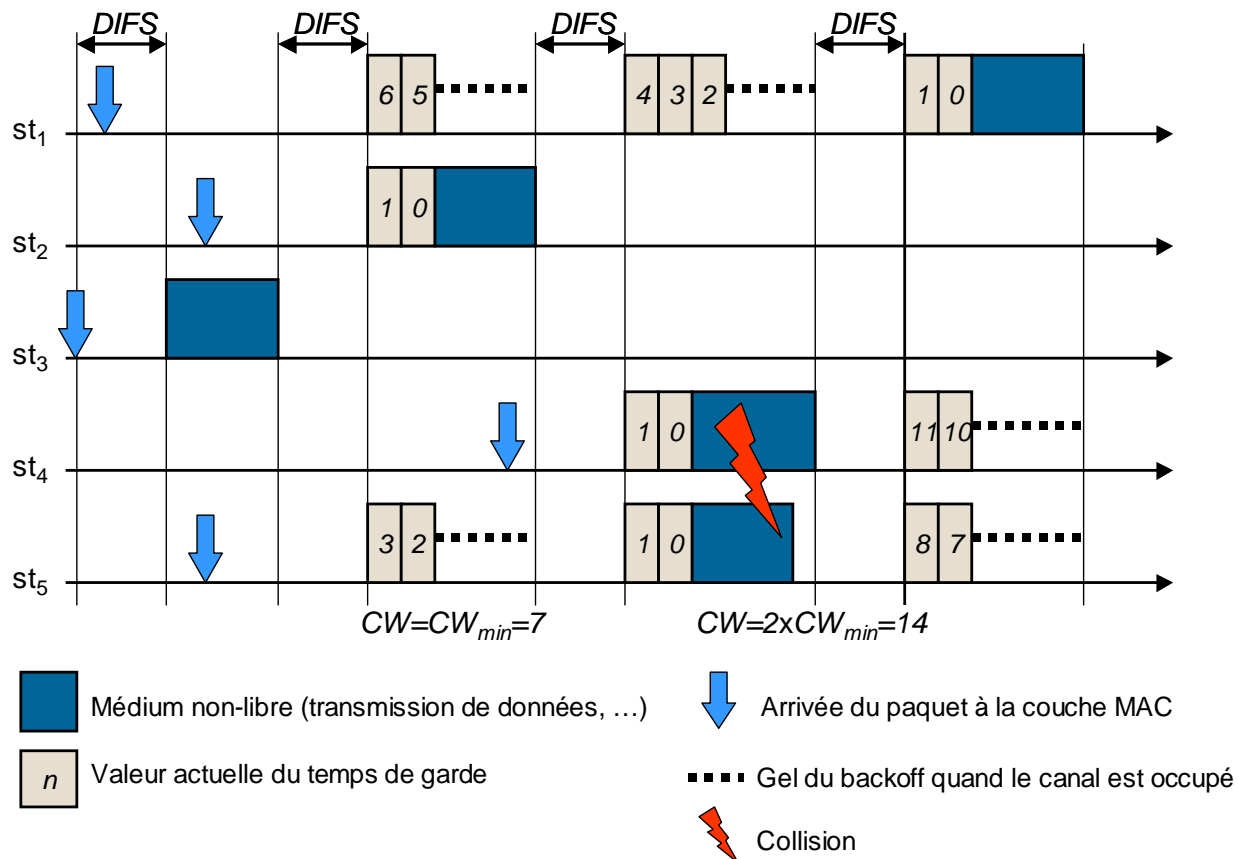


Figure 3.12 : Scénario de fonctionnement de CSMA/CA

Un nœud qui souhaite émettre doit d'abord attendre pendant *DIFS* après que le canal ait été libéré. Si le canal est toujours libre à cet instant, il peut envoyer son paquet, sinon, il devra essayer à la prochaine libération du canal après avoir choisi un temps de garde.

Dans le scénario proposé, le nœud 3 reçoit en premier un paquet à émettre. Au bout de *DIFS* le canal est toujours libre, donc 3 émet son paquet. Les autres nœuds qui ont un paquet à émettre (1, 2 et 5) voient un canal occupé et entrent dans une phase de contention.

Chacun de ces trois nœuds choisit un **temps de garde aléatoire** compris dans une fenêtre de contention (*CW*, pour *contention window*) $[0, CW]$ avec au départ $CW = CW_{min} = 7$. Tant que le canal est libre, chaque nœud décrémente simultanément son compteur. Le premier nœud arrivant à zéro (ici le nœud 2) émet son paquet. Les autres nœuds **gèlent leur compteur tant que le canal est occupé**. Ainsi, à la prochaine fenêtre d'émission, leur temps de garde sera statistiquement plus faible que ceux des nœuds qui n'étaient pas déjà en train d'attendre pour émettre un paquet (ce qui assure une meilleure équité et tente d'éviter la *famine* pour certains nœuds).

L'intervalle de choix pour le temps de garde étant réduit, il est probable que deux stations auront le même. Par exemple, 4 choisit un temps de garde de 1, et 5 a le même temps résiduel. Les deux stations envoient leur paquet simultanément, ce qui cause une collision.

Cette collision est détectée et les deux nœuds **doublent la taille de leur fenêtre de contention** (elle est doublée à chaque collision jusqu'à une valeur maximale et est remise à la valeur minimale après chaque transmission réussie d'un paquet), ce mécanisme est appelé **BEB**, pour *binary exponential backoff*. Les deux nœuds impliqués dans la collision choisissent un nouveau temps de garde dans la nouvelle fenêtre de contention et attendent pour émettre de nouveau leur paquet.

Cette vision de l'échange de données représente le mécanisme complet pour les messages diffusés en *broadcast*. Dans le cas d'envois point à point, un mécanisme d'acquittement du message est mis en place. Le récepteur du paquet de données répond à l'émetteur par l'envoi d'un message d'acquittement (ACK) après avoir attendu la durée *SIFS*. L'acquittement du paquet est donc prioritaire sur l'envoi de données par d'autres nœuds du réseau.

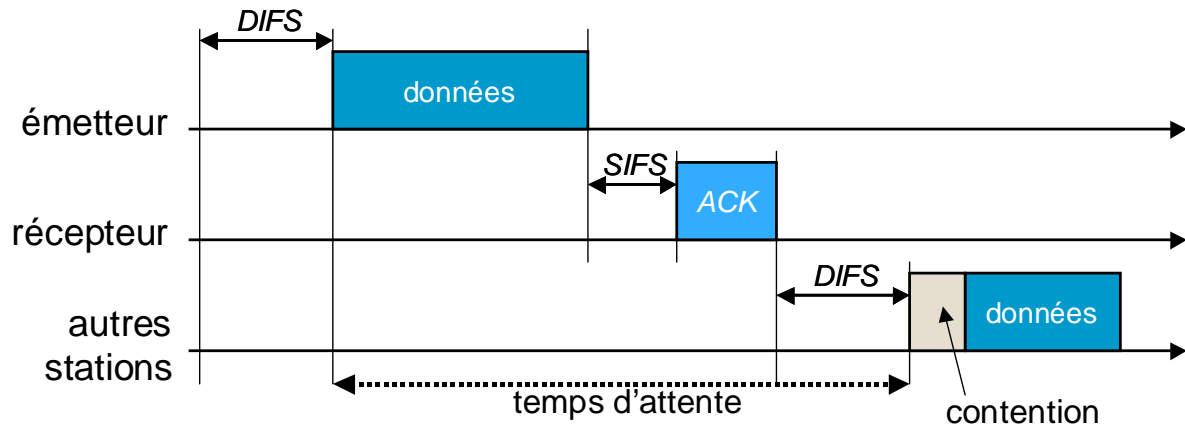


Figure 3.13 : Transfert de données standard de 802.11

3.2.1.2.2 MACA

A première vue, le mécanisme de CSMA/CA semble suffisant pour assurer dans la plupart des cas l'utilisation du canal par un seul nœud en émission et donc éviter de manière quasi certaine la collision de paquets.

Pourtant, dans les faits il existe une configuration, souvent décrite comme le *problème du nœud caché*, et qui met en évidence les limitations de l'algorithme décrit précédemment.

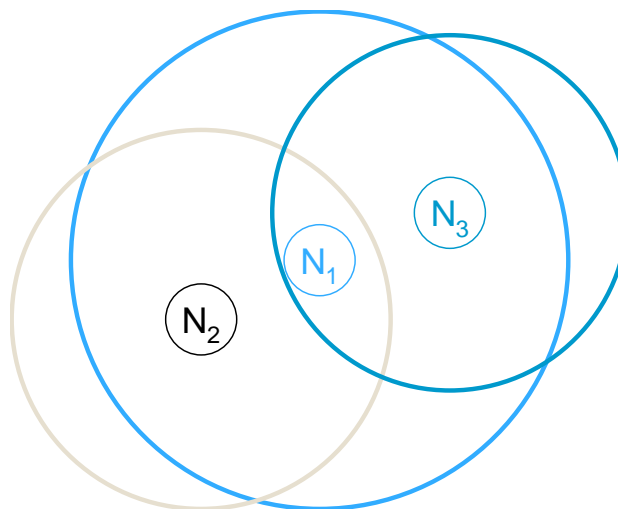


Figure 3.14 : Configuration du nœud caché

La situation correspondante, décrite dans la Figure 3.14, fait que comme le nœud N_2 n'est pas dans la zone de portée du nœud N_3 , si ce dernier est en train d'envoyer un paquet au nœud N_1 , N_2 n'a aucun moyen, même en écoutant le canal, de détecter l'occupation de ce dernier et donc d'éviter une éventuelle collision de paquets. Notons que le modèle de portée retenu dans cet exemple est extrêmement simpliste. Un modèle plus complet fait apparaître d'autres sources de collision et sera décrit dans la section 3.2.2.1 de ce document.

3.2.1.2.2.1 Mécanisme RTS/CTS

Pour pallier ce problème, il est possible d'activer un mécanisme de réservation virtuelle du canal qui assure qu'un seul et unique nœud pourra émettre à la fois. Ce mécanisme est appelé *extension RTS/CTS* (pour *request to send*, *clear to send*, nom des messages de contrôle supplémentaires introduits par cette technique).

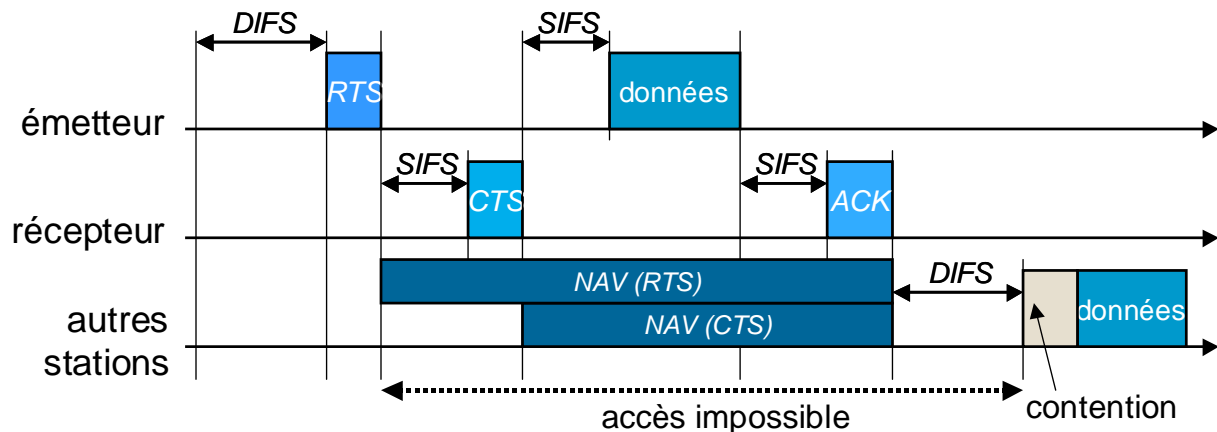


Figure 3.15 : Transfert de données avec extension RTS/CTS

Dans cette configuration de la couche MAC, chaque nœud qui souhaite envoyer un message à un récepteur précis envoie, après avoir attendu *DIFS*, plus un éventuel temps de garde si le médium était occupé, émet un message *RTS* qui contient entre autres :

- L'adresse du destinataire de la transmission (donc le destinataire du paquet *RTS*);
- L'adresse de l'émetteur de la transmission (donc l'émetteur du paquet *RTS*);
- La durée attendue de la transmission complète (*CTS* + données + *ACK*).

Chaque nœud qui entend ce message sans être son destinataire, met à jour son vecteur d'allocation du réseau (ou *NAV* pour *network allocation vector*) à la durée de transmission indiquée. Pendant toute cette durée, il ne tentera pas d'accéder au canal.

Le destinataire du paquet de contrôle répond avec un message *CTS* qui contient entre autres :

- L'adresse de l'émetteur du paquet *RTS* (donc destinataire du *CTS*);
- La durée attendue de la transmission résiduelle (données + *ACK*).

Les nœuds qui entendent ce message sans en être le destinataire mettent à jour leur vecteur d'allocation du réseau (ou *NAV* pour *network allocation vector*) à la durée de transmission résiduelle indiquée. Pendant toute cette durée, ils ne tenteront pas d'accéder au canal.

3.2.1.2.2.2 RTS/CTS et fragmentation de paquets

Comme les réseaux sans-fil sont sujets à de nombreuses perturbations et interférences, le taux de bits d'erreur est généralement assez élevé et nombreux sont les paquets qui doivent être retransmis car ils ont été reçus avec des erreurs (chaque paquet contient un *CRC*, *check redundancy code* qui permet de vérifier son intégrité).

Pour éviter d'avoir à retransmettre des paquets trop longs, il est intéressant de fragmenter les paquets en entités plus petites. Chaque fragment erroné peut alors être retransmis indépendamment des autres, ce qui limite l'*overhead* lié aux retransmissions dans un canal sujet aux erreurs.

Le mécanisme *RTS/CTS* permet d'envoyer facilement des paquets fragmentés. Après réception du message *CTS*, l'émetteur envoie, au lieu des données complètes, le premier fragment de données avec un nouvel indicateur de temps (pour la durée de l'acquittement de ce fragment et l'envoi du fragment suivant). Le récepteur répond alors, au bout de *SIFS*, avec l'acquittement du premier fragment auquel il joint la durée de la transmission suivante. Sur réception de cet acquittement modifié, l'émetteur envoie le second fragment et ainsi de suite.

Les nœuds qui entendent ces messages sans en être destinataire adaptent leur *NAV* en fonction des durées contenues dans les paquets pour ne pas utiliser le canal pendant la durée de transmission des fragments et acquittements successifs.

3.2.1.2.2.3 Données de contrôle engendrées

Si le mécanisme *RTS/CTS* pallie le problème du nœud caché en assurant une meilleure réservation du canal, il n'est pas sans désavantage. En particulier, il augmente le nombre de paquets de contrôle et donc réduit la bande passante utile du canal.

Soit m la taille des données utiles à transmettre (c'est la taille du paquet au niveau application). Soit n la taille de ce paquet au niveau physique ($n = m +$ taille des en-têtes des couches intermédiaires). Soit n_{ack} la taille du paquet d'acquittement au niveau physique. Soit d_0 le débit du canal pour les paquets de contrôle, d_1 le débit pour les données, t_{slot} la taille d'un slot et CW_{min} la taille minimale de la fenêtre de contention.

A partir de ces données, il est possible de calculer le débit utile du canal avec ou sans le mécanisme *RTS/CTS* :

$$D_{default} = \frac{m}{DIFS + \frac{n}{d_1} + SIFS + \frac{n_{ack}}{d_0} + \frac{CW_{min}}{2} t_{slot}} \quad (\text{eq. 3.2})$$

$$D_{RTS/CTS} = \frac{m}{DIFS + \frac{n_{RTS}}{d_0} + \frac{n_{CTS}}{d_0} + \frac{n}{d_1} + \frac{n_{ack}}{d_0} + 3SIFS + \frac{CW_{min}}{2} t_{slot}} \quad (\text{eq. 3.3})$$

En utilisant les valeurs des paramètres correspondant à la norme 802.11b (pour lequel le débit maximal théorique est de 11Mbps/s), les débits utiles sont égaux à 5,120Mbps/s pour le mode normal et à 4,386Mbps/s pour le mécanisme *RTS/CTS*, soit une utilisation relative respectivement inférieure à 45% et 40%.

3.2.1.2.3 PCF

La dernière alternative de contrôle d'accès au canal de transmission est une fonction sans contention qui permet à chaque nœud d'avoir un accès au canal dans un temps borné. Cette fonction, basé sur une interrogation des nœuds nécessite un point de coordination (d'où son nom de *point coordination function*) c'est-à-dire un nœud qui va, de manière centralisée, gérer l'accès au média. Pour cette raison, cette méthode est difficilement transposable aux réseaux ad-hoc et n'est que brièvement décrite ici.

Le coordinateur, après avoir attendu que le canal soit libre pendant *PIFS*, envoie à la première station les données qu'il avait en cache et qui lui étaient destinées. Si celle-ci a à son tour des paquets en cache à transmettre au coordinateur ou aux autres stations, elle les envoie après avoir attendu *SIFS* sinon, elle reste muette.

Au bout de *PIFS*, si aucun message n'a été reçu de la station, le coordinateur passe à la station suivante. Quand toutes les stations ont été interrogées, le coordinateur envoie un message de fin de période de non-contention (CF_{end}). Une fois cette période terminée, le système passe en mode contention pour une durée donnée avant de reprendre le mécanisme d'interrogation.

3.2.1.2.4 Autres fonctions de la couche MAC

Le rôle de la couche MAC de 802.11 ne se limite pas à allouer aux différents nœuds un accès au canal afin de limiter les collisions. Elle assure aussi des fonctions importantes de synchronisation, de gestion de l'énergie et de *roaming*. La réalisation de ces fonctions est particulièrement intéressante dans le cas de réseaux ad-hoc et est détaillée dans les paragraphes suivants.

3.2.1.2.4.1 Synchronisation de nœuds

La synchronisation des nœuds se fait par l'envoi de *timestamps* qui sont contenus dans les *frames* de *beacon* (paquets envoyés régulièrement par le point d'accès pour avertir entre autre du nom actuel du réseau sans-fil). Ces messages sont envoyés à intervalle régulier, appelé *intervalle de synchronisation*, ou à la première fenêtre libre du canal après le début de l'intervalle.

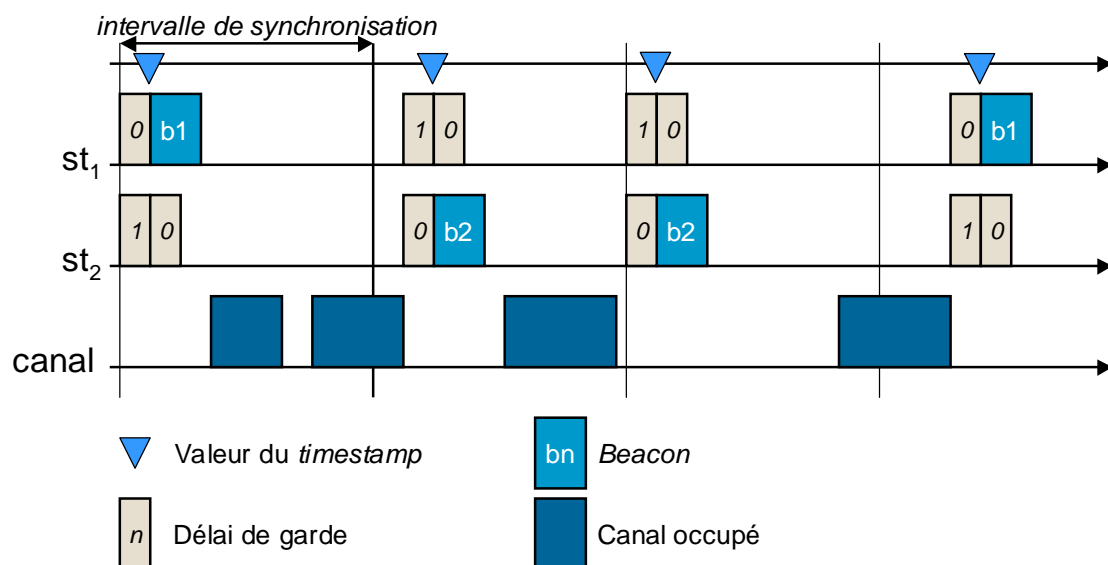


Figure 3.16 : Synchronisation en mode ad-hoc

En mode ad-hoc, où il n'y a pas de point d'accès, chaque station peut émettre un *beacon*. A chaque début d'intervalle les stations tentent d'envoyer un tel paquet avec un temps de garde aléatoire (comme pour toutes les autres données). Sur réception du *beacon* envoyé par la station la plus rapide, les autres mettent à jour leur horloge et suppriment leur paquet de *beacon* pour cet intervalle, si bien qu'un seul de ces paquets est effectivement envoyé par cycle. Si une collision survient, le *beacon* est perdu pour ce cycle et il faudra attendre le prochain intervalle pour se synchroniser.

3.2.1.2.4.2 Gestion de l'énergie

La gestion de l'énergie est un enjeu important des réseaux sans fil puisque, dans la plupart des cas, les nœuds du réseau sont des terminaux fonctionnant sur batterie. Pour augmenter leur autonomie, il est donc indispensable de limiter leur consommation, notamment en introduisant des périodes de sommeil pendant lesquelles les stations n'émettent pas et n'écoutent pas le canal.

La gestion d'énergie est complexe dans les réseaux ad-hoc (du fait qu'aucun nœud ne peut centraliser le processus de gestion) et des contraintes supplémentaires sont imposées aux stations. En particulier, ces dernières doivent être correctement synchronisées (d'où l'importance du paragraphe précédent) et implémenter un *buffer* pour stocker les messages à transmettre aux stations en sommeil. Ces messages sont annoncés en utilisant des paquets de contrôle appelés *ad-hoc traffic indication map* (ATIM).

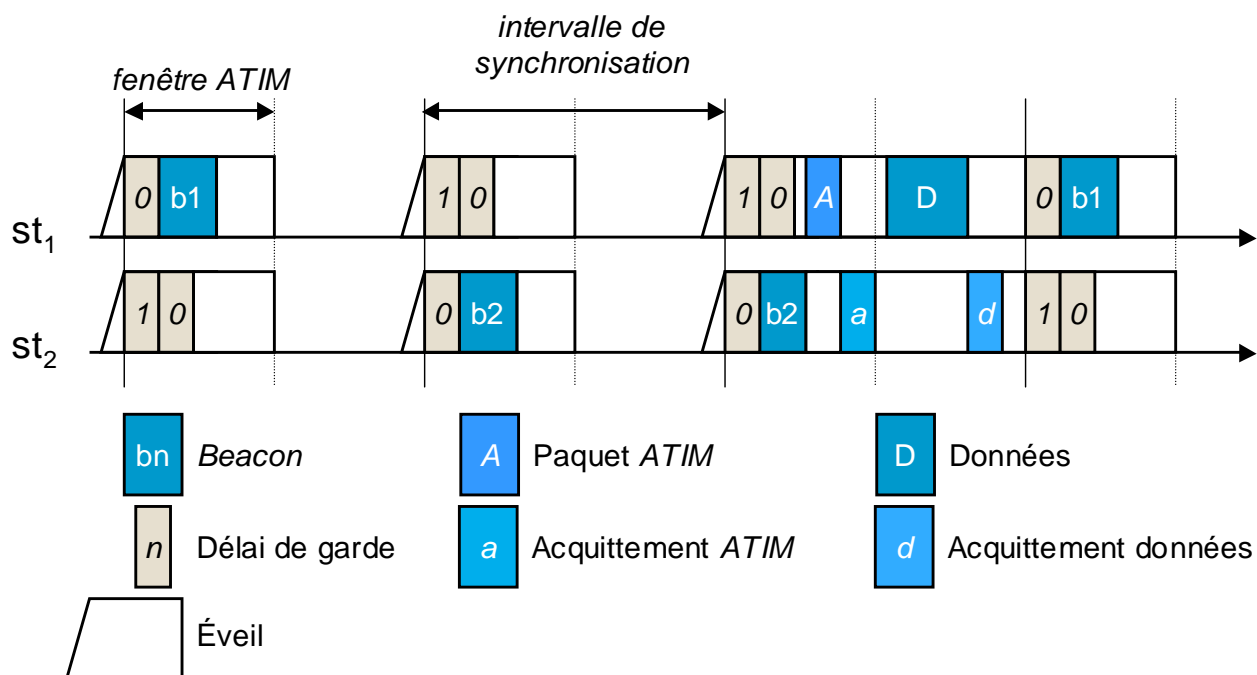


Figure 3.17 : Gestion de l'énergie en mode ad-hoc

Chaque intervalle de synchronisation (défini en 3.2.1.2.4.1) contient une fenêtré ATIM dont la taille est la moitié de l'intervalle. Toutes les stations sont éveillées pendant cette fenêtré. Si aucun paquet ATIM n'est transmis pendant cette fenêtré ou si aucun paquet ATIM émis n'est à destination d'une station celle-ci se rendort à la fin de la fenêtré jusqu'au prochain cycle.

Sinon, elle reste en éveil, envoie un acquittement pour le paquet ATIM la concernant, écoute le canal pour recevoir les données qui lui sont destinées, les acquitte et se rendort jusqu'à la fin du cycle. Selon les paramétrages, la station peut rester en éveil si le début du prochain cycle est proche (car la transition éveil/sommeil est également coûteuse en énergie et peut ne pas être bénéfique dans ce cas).

3.2.2 LES LIMITATIONS DE 802.11 DANS LES RESEAUX MOBILES AD-HOC

La section précédente a décrit les principes de fonctionnement du standard 802.11 et mis en avant certains problèmes inhérents à leur conception, notamment en soulignant le problème du nœud caché et l'impact des messages de contrôle sur la capacité du canal réellement disponible pour l'envoi de données.

Dans le cadre de l'utilisation de réseaux sans fil en mode ad-hoc, et notamment en mode multibond où les stations doivent effectuer des opérations de routage, le mécanisme de contrôle d'accès au canal de 802.11 pose différents problèmes, principalement liés à la vision trop simpliste adoptée pour définir ces protocoles et à l'interaction avec les couches réseau supérieures, TCP notamment.

3.2.2.1 L'IMPORTANCE DU MODELE DE PROPAGATION

Dans le paragraphe 3.2.1.2.2, a été introduit le problème du nœud caché, et avec lui le modèle le plus simple utilisé pour représenter la portée des différents nœuds du réseau : un simple cercle centré sur le nœud émetteur. La réalité de la propagation des ondes est bien différente puisque les conditions géographiques, météorologiques, la présence d'autres signaux parasites viennent grandement complexifier la zone de couverture d'un nœud du réseau.

Aussi, pour mieux appréhender les problèmes liés à la propagation radio sur le canal, il est important de grader à l'esprit que :

- Il n'y a pas de frontière absolue ou réellement observable au-delà de laquelle une station ne peut plus recevoir de message ;
- Le canal n'est pas protégé contre les signaux extérieurs qui peuvent venir le perturber ;
- Le médium radio est beaucoup moins fiable que les liaisons filaires ;
- Le canal a des propriétés de propagation asymétriques et variables dans le temps.

Ces assertions conduisent à remettre en cause le modèle de propagation simpliste adopté jusqu'ici pour essayer de mieux appréhender la complexité des mécanismes d'accès au canal.

3.2.2.1.1 Modèle de propagation raffiné

Bien sur, il est impossible de se passer d'un modèle et de travailler sur des données réelles pour concevoir ou expliquer le fonctionnement des mécanismes de contrôle d'accès au canal. Pourtant, les remarques précédentes nous conduisent à élaborer un modèle de propagation plus raffiné.

Dans un premier temps, remarquons que la capacité d'une station à écouter et à comprendre les messages du canal n'est pas binaire. L'amplitude du signal, et le rapport signal à bruit décroît avec la distance à l'émetteur. Aussi, il semble judicieux d'introduire trois distances caractéristiques (mesurées depuis la station qui émet)[11] :

- *Distance de transmission (TX_range)* : qui est la distance maximale à laquelle une station peut recevoir correctement (i.e. sans erreur) un paquet ;
- *Distance d'interférence (IF_range)* : qui est la distance en deçà de laquelle les stations en mode réception souffriront d'interférence avec un émetteur et donc expérimenteront des pertes de paquets ;
- *Distance d'écoute du canal (PCS_range)* : qui est la distance maximale à laquelle une station peut détecter l'occupation du canal pendant la transmission.

Ces définitions nous suggèrent la double-inégalité suivante :

$$TX_range \leq IF_range \leq PCS_range \quad (\text{eq. 3.4})$$

Par exemple, le simulateur *ns2* retient les valeurs de 250m pour *TX_range* et de 550m pour *IF_range* et *PCS_range* ($PCS_range \approx IF_range \geq 2 \times TX_range$).

Autre élément très important à prendre en compte, est le fait que la distance de transmission dépende du débit d'émission (à puissance d'émission constante). Or, pour des soucis de rétro compatibilité, les paquets de contrôle sont tous envoyés à un mégabit par seconde, même quand les paquets de données peuvent être envoyés à un débit beaucoup plus élevé.

Aussi, dans les standards les plus récents de la norme 802.11 comme 802.11g, paquets de données et paquets de contrôle ont des distances de transmission très différentes. Les valeurs, tirées de résultats expérimentaux[11], sont résumées dans le Tableau 3.4.

Tableau 3.4 : Distance de transmissions des paquets de données et de contrôle

| Type de paquets | 11 Mbits/s | 5.5 Mbits/s | 2 Mbits/s | 1 Mbit/s |
|-------------------------|------------|-------------|-----------|-----------|
| Data <i>TX_range</i> | 30 m | 70 m | 90-100 m | 110-130 m |
| Control <i>TX_range</i> | | | ≈ 90 m | ≈ 120 m |

Les zones d'interférence et d'écoute du canal quant à elles ne dépendent pas du débit de transmission mais uniquement de la puissance d'émission, supposée fixée pour un nœud donné. Aussi, le modèle d'émission pour un nœud donné peut être représenté ainsi que décrit dans la Figure 3.18.

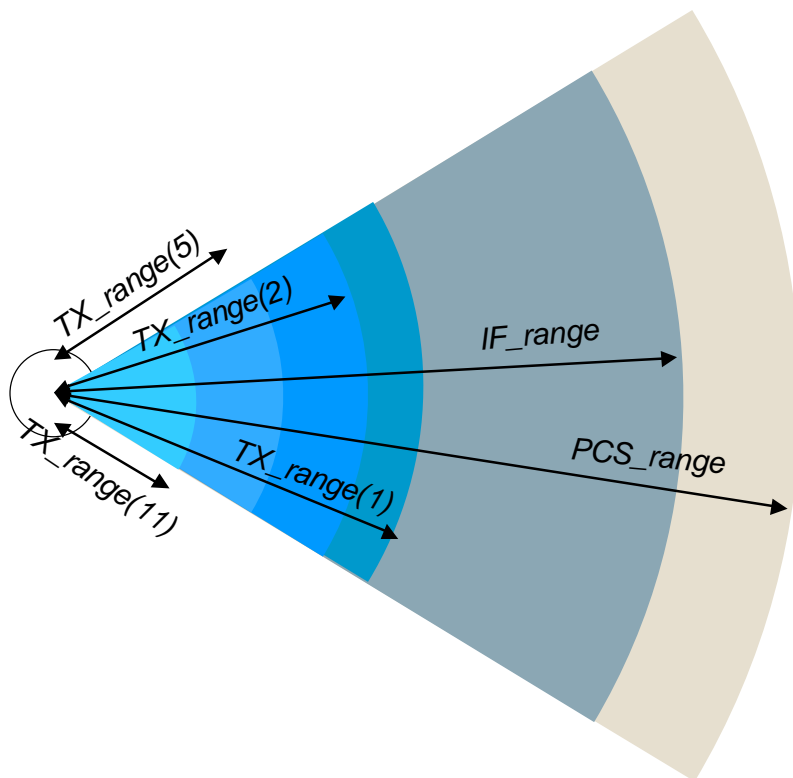


Figure 3.18 : Modèle radio raffiné

3.2.2.1.2 Nœuds cachés et nœuds exposés

L'utilisation de ce modèle de couverture raffiné conduit à une multiplication des problèmes de nœuds cachés et de nœuds exposés, dont des exemples caractéristiques sont donnés dans les paragraphes suivants.

3.2.2.1.2.1 Nœuds cachés

Soit un nœud R_1 en mode réception et situé dans la zone de transmission d'un nœud S_1 . Soit un nœud S_2 , dont la zone d'interférence couvre R_1 , mais pas S_1 . Si S_2 est en train de communiquer avec un second nœud en mode réception, R_2 par exemple, et que S_1 tente d'envoyer des données à R_1 , celui-ci ne pourra pas les recevoir correctement du fait de l'interférence avec S_2 .

Le plus dangereux, est que S_1 , qui n'est pas dans la zone d'écoute du canal de S_2 n'a aucun moyen de savoir que S_2 est en train de transmettre.

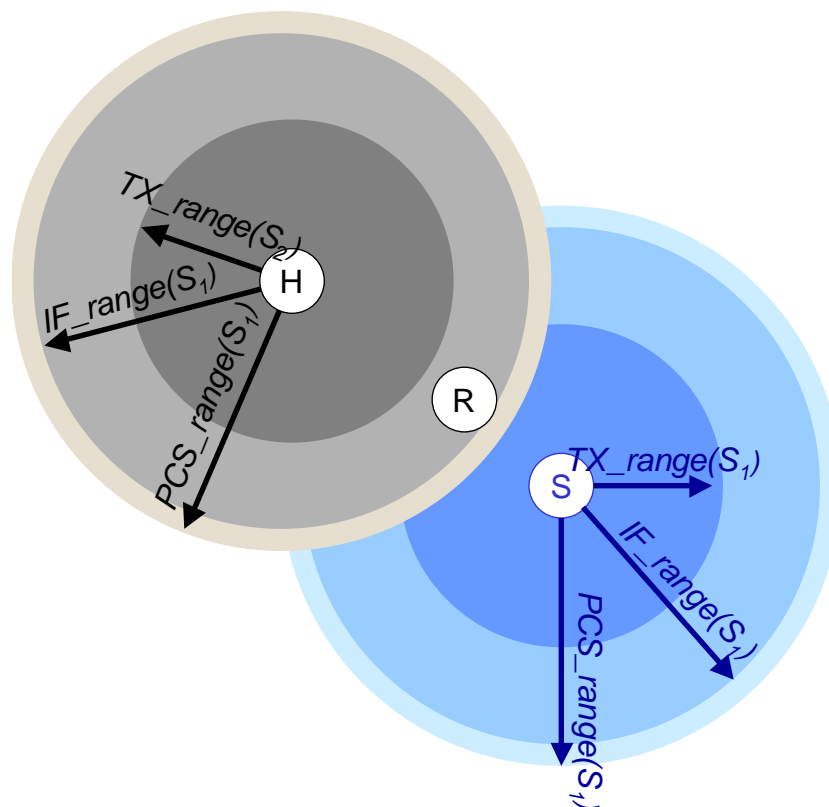


Figure 3.19: Le problème du nœud caché

En conclusion, on définira comme nœud caché H (ici S_2) perturbant la transmission entre un émetteur S et un récepteur R , tout nœud satisfaisant :

$$d(S, H) < PCS_Range(S) \cap TX_range(H) < d(H, R) < IF_range(H) \quad (\text{eq. 3.5})$$

Il existe bien entendu d'autres situations dans lesquelles un nœud caché peut venir perturber les communications d'un nœud ou l'empêcher d'émettre.

3.2.2.1.2.2 Nœuds exposés

Une situation de nœud exposé correspond à un nœud récepteur R_2 , placé dans la zone d'écoute du canal d'un nœud émetteur S_1 , qui transmet des données à un autre récepteur R_1 .

Quand R_2 reçoit des données d'un second émetteur S_2 , il ne peut y répondre (par exemple pour envoyer un acquittement) car il est exposé aux transmissions de S_1 . Ce qui conduit S_2 , après avoir augmenté sa fenêtre de contention, à choisir un nouveau temps de garde et à retransmettre le paquet jusqu'à déclarer une rupture de liaison avec le nœud R_2 qui est pourtant dans sa zone de réception.

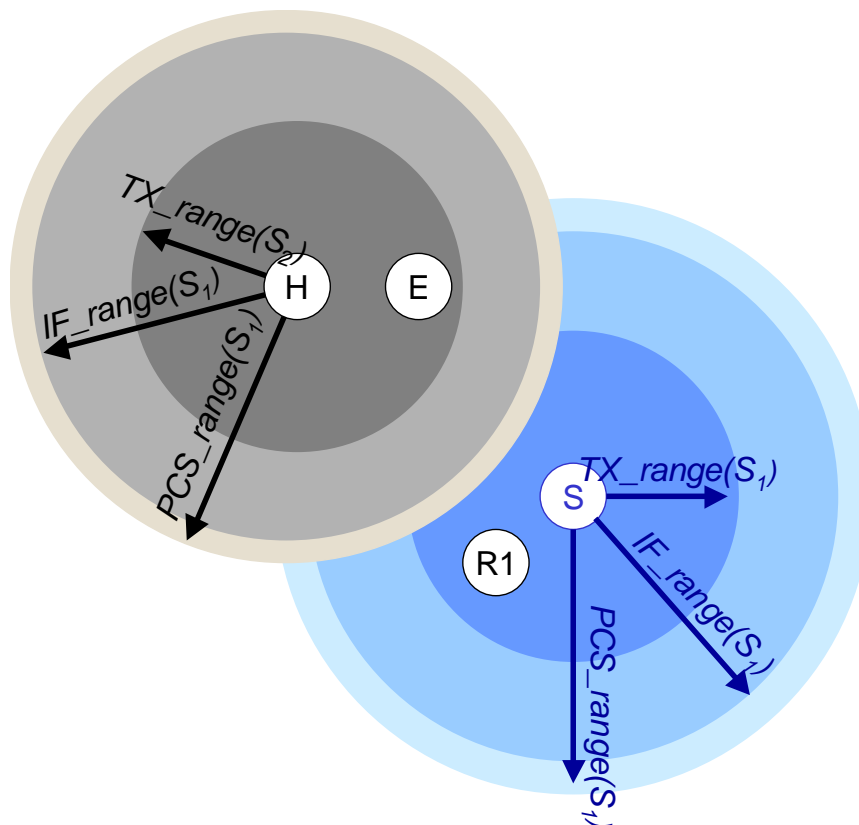


Figure 3.20 : Le problème du nœud exposé

On définira donc comme nœud E , exposé à un nœud émetteur S_1 et ne pouvant répondre au nœud S_2 qui souhaite transmettre avec lui, tout nœud satisfaisant :

$$d(S_2, E) < TX_Range(S_2) \wedge TX_range(S_1) < d(S_1, E) < PCS_range(S_1) \quad (\text{eq. 3.6})$$

3.2.2.1.3 Les zones grises

Un autre problème directement lié au modèle de propagation plus élaboré que nous venons de décrire concerne l'existence de zones grises, dans lesquelles la couche MAC et le protocole de routage supposent que le nœud peut recevoir, émettre et transmettre des paquets de données alors qu'en réalité, il parvient uniquement à entendre et répondre si nécessaire aux paquets de contrôle.

Ce cas de figure est valable pour les protocoles de routage qui utilisent des petits paquets de contrôle, le plus souvent *broadcastés* dans le réseau afin d'établir ou de maintenir à jour les informations relatives à la topologie. Ceci s'applique donc à *OLSR*, notamment lors de l'utilisation des messages *HELLO* et *TC* décrits dans la section 3.1.2.1.2.

Différents facteurs contribuent à l'apparition de ces zones grises, en particulier, les différentes propriétés des messages de contrôle comparées aux paquets de données jouent un rôle important.

Ces propriétés et leurs effets sont résumés ci-dessous :

- *Vitesse de transmission* : Les messages de signalisation sont transportés à une vitesse plus faible (2Mbits/s) alors que les paquets de données sont transmis à des vitesses supérieures (11Mbits/s à 54Mbits/s suivant le standard adopté). Par conséquent, les paquets de contrôle ont une distance de transmission beaucoup plus grande que les données ;
- *Accusé de réception* : Dans les standards 802.11b et 802.11g, les paquets *broadcastés* sont transmis sans accusé de réception. Par conséquent, une station qui reçoit un tel message d'un voisin ne peut faire aucune hypothèse sur la bidirectionnalité du lien qui les unit ;
- *Taille du paquet* : Les paquets de contrôle sont de taille généralement très inférieure à celle des paquets de données. Ce faisant, ils sont beaucoup moins affectés par les pertes de transmission et ont beaucoup moins de chance d'entrer en collision avec d'autres données sur le canal.

Parmi les différentes façons de résoudre le problème des zones grises, il est possible de renforcer le nombre de messages de contrôle échangés avec succès avant d'établir la validité d'un lien ou d'insérer un seuil de signal à bruit pour considérer ces messages de contrôle comme valides. Notons que ces deux options sont suggérées dans la description du protocole *OLSR*.

3.2.2.2 INTERACTIONS AVEC LA COUCHE TCP

D'autres problèmes, plus directement l'implémentation du contrôle d'accès au canal dans la norme 802.11, apparaissent lors de l'utilisation de liaisons sans fil respectant ce standard dans un réseau ad-hoc multibond. Dans la plupart des cas, ils sont liés à une interaction qui s'avère dommageable entre la couche MAC et la couche TCP.

3.2.2.2.1 TCP et les connexions sans-fil

TCP (*Transport Control Protocol*) a été conçu pour être une couche de transport fiable et orientée connexion. En particulier, il fournit des services de livraison en ordre des données et de détection de perte. Il est *full duplex* et, grâce à un mécanisme dit de *fenêtre glissante*, il contrôle le flux de données et tente d'éviter les congestions au sein du réseau.

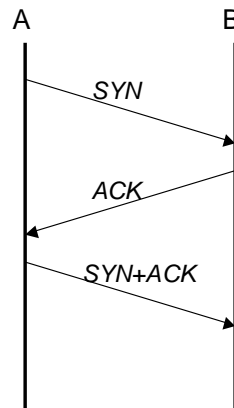


Figure 3.21 : Connexion en trois passes de TCP

En particulier, TCP est assez mal adapté à ce qui reste les principales limitations des réseaux ad-hoc sans-fil : la mobilité des nœuds, le fort taux d'erreur et la bande passante limitée.

3.2.2.2.2 L'instabilité de TCP

Cette inadaptation aux connexions sans-fil tient en particulier dans le mécanisme adopté pour ajuster la taille de la fenêtre de congestion à la charge du réseau. La fenêtre de congestion est le nombre de paquets successifs que TCP peut envoyer avant de recevoir l'acquittement du premier paquet envoyé.

Initialement, la fenêtre de congestion prend la valeur d'un *MSS* (*maximum segment size*). Après chaque paquet correctement envoyé (pour lequel un *ACK* a été reçu), la taille de cette fenêtre est doublée jusqu'à atteindre la valeur de seuil de départ lent *SST* (*slow start threshold*), c'est la phase dite de départ lent (*slow start*).

Passée cette phase, la taille de la fenêtre de congestion augmente d'un *MSS* à la réception de chaque acquittement d'un paquet envoyé (cette phase est appelée *contrôle de congestion*).

À la première congestion du réseau (pas d'*ACK* reçu avant l'expiration d'une temporisation), la taille de la fenêtre est ramenée à 1 *MSS* et la valeur de seuil de la phase de départ rapide est initialisée à la moitié de la taille maximale qu'avait prise la fenêtre. Puis la phase de départ recommence et la fenêtre double jusqu'à atteindre la nouvelle valeur de *SST*.

TCP considère également comme congestion la réception de trois *ACK* dupliqués et retransmet alors le dernier segment non acquitté (*fast retransmit*).

Pour éviter d'avoir à répéter trop souvent la phase de départ lent, notamment si un seul paquet manque ou si les paquets ont été transmis dans le désordre, des améliorations de TCP (comme TCP Reno) permettent un redémarrage rapide (*fast recovery*). Dans les deux cas de congestion cités précédemment, les données incriminées sont retransmises immédiatement et la taille de la fenêtre de congestion n'est que divisée par deux. Au prochain paquet correctement transmis, la taille de la fenêtre sera doublée ou incrémentée selon que l'on est en dessous ou au-dessus du seuil de démarrage lent.

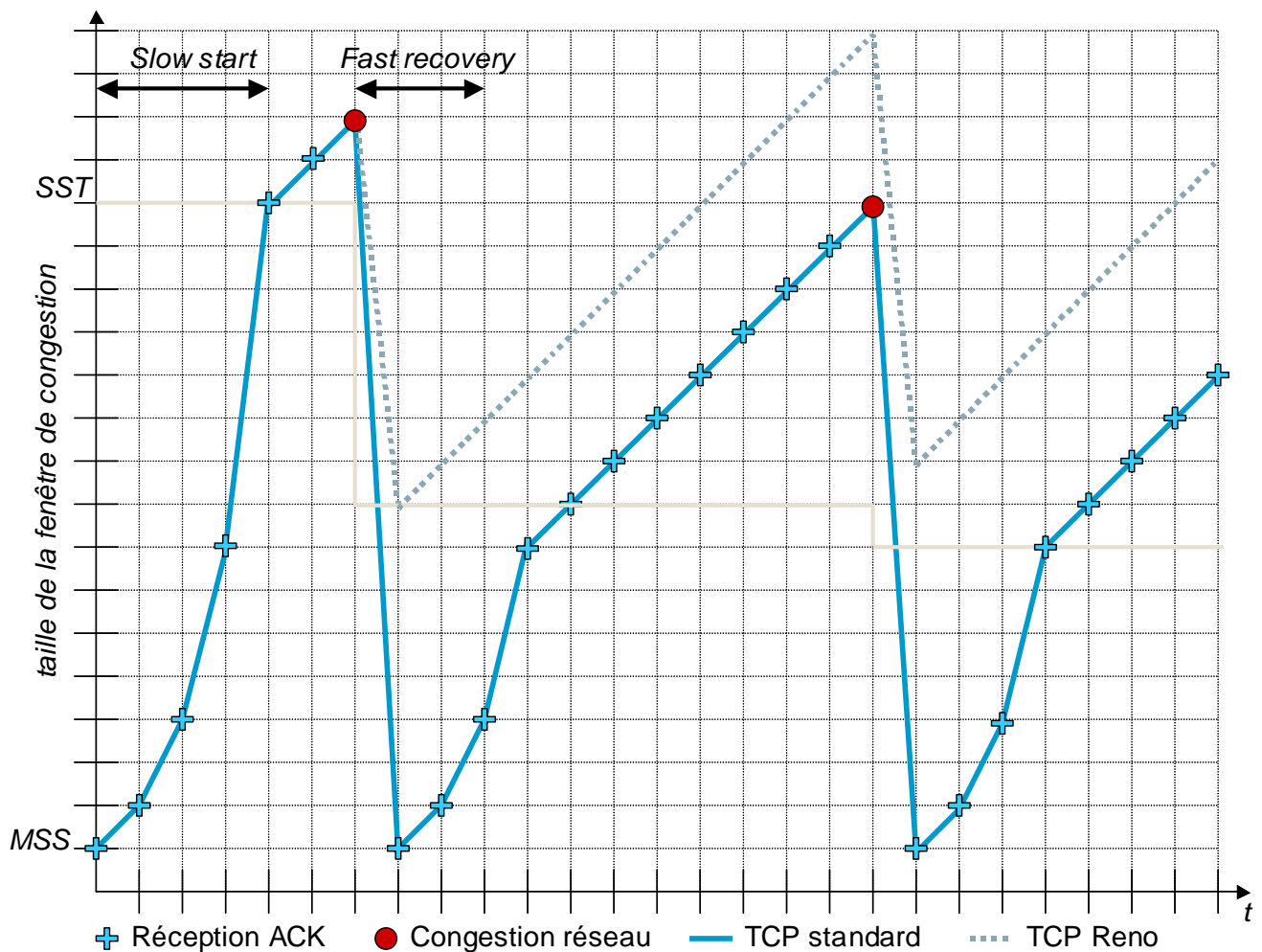


Figure 3.22 : Dynamique de la fenêtre de congestion TCP

Le principal problème de ce mécanisme est qu'il diminue fortement les performances de transmission à chaque perte de paquet. Or, dans le cadre de réseaux mobiles, la perte de paquets est bien plus souvent due à des échecs dans les liaisons qu'à une réelle congestion du réseau. Par conséquent, TCP opère bien en dessous de la valeur optimale de la fenêtre de congestion dans les réseaux ad-hoc multibonds où de nombreux liens entre nœuds apparaissent et disparaissent.

Notons également que le mécanisme *BEB* (*binary exponential backoff*) utilisé pour gérer la contention d'accès au canal dans CSMA/CA contribue à favoriser cette situation.

Ce comportement, décrit et étudié par S. Xu et T. Saadawi[12], cause une vraie diminution de la capacité du canal. Les auteurs montrent en particulier que lorsqu'un nœud ne parvient plus à joindre un nœud adjacent, c'est la couche MAC qui déclare le lien brisé, après de multiples essais de retransmission qui diminuent la taille de la fenêtre TCP. Les auteurs ont remarqué qu'avec une taille de fenêtre plus petite, le phénomène est moins marqué. Mais cette stabilité s'acquiert au prix d'une limitation de l'optimisation des transmissions (due à la limitation de la taille de la fenêtre de congestion).

Aussi, différentes approches pour résoudre ce problème et permettre à TCP de faire la distinction entre congestion du réseau et échec des liaisons ont été développées. Elles seront décrites dans la section 3.3.1.

3.2.2.2.3 Iniquité entre les nœuds

Un second phénomène, lié aux interactions entre la couche TCP et la couche MAC de 802.11, est l'incompatibilité entre deux connexions TCP entre des nœuds mobiles, qui cause une iniquité entre ces nœuds : une des deux connexions existantes capture l'ensemble de la capacité du canal et réduit les possibilités pour l'autre connexion de transmettre des données.

Les causes de cette capture, décrites plus amplement dans [12], peuvent être résumées aux facteurs suivants :

- Le problème des stations cachées/exposées qui peuvent empêcher deux communications parallèles d'avoir lieu simultanément ;
- Les tailles supérieures des zones d'interférence de sensibilité à la porteuse qui augmentent la probabilité d'apparition des situations précédentes ;
- Le mécanisme *BEB* qui favorise la dernière station ayant réussi à transmettre par rapport à celle ayant échoué dans leur transmission ;
- Le *three way handshake* de TCP (voir Figure 3.21 : Connexion en trois passes de TCP) qui nécessite l'échange bidirectionnel de nombreux petits paquets de contrôle pour initier/confirmer un échange de données et qui favorise les connexions établies au dépend des connexions cherchant à s'établir, dans le cadre des réseaux sans fil.

La résolution du problème de l'équité entre nœuds nécessite donc des modifications importantes des couches TCP et MAC pour les adapter aux conditions particulières de fonctionnement des réseaux sans fil multibonds.

3.3 AMELIORATIONS DES STANDARDS POUR LES RESEAUX MOBILES AD-HOC

Les paragraphes précédents nous ont montré les limitations des standards de la norme 802.11 lors de son utilisation dans des réseaux ad-hoc sans fil. Les problèmes engendrés proviennent principalement de l'inadaptation de la couche de contrôle d'accès au canal aux situations particulières liée à la mobilité des nœuds et d'interactions malheureuses entre cette couche et la couche de transport TCP.

Aussi, les paragraphes suivants décrivent succinctement des adaptations possibles de TCP avant de revenir plus en détail sur des mécanismes de contrôle d'accès au canal plus évolués et spécialement dédiés aux réseaux ad-hoc.

3.3.1 AMELIORATIONS DE TCP

Comme nous l'avons vu, le principal défaut de TCP est son inadaptation aux particularités des réseaux sans-fil, notamment la latence variable dans le transport des paquets et leur fort taux d'erreur à la réception. Ces deux phénomènes, faussement interprétés par TCP comme des cas de congestion du réseau, conduisent à de nombreuses phases de *slow start*, et diminuent d'autant le débit de données sur le réseau.

Ainsi, pour éviter ce problème, une des solutions possibles consiste à s'assurer, qu'au niveau de la couche de transport, toutes les erreurs de transmissions sont corrigées (il appartient donc aux

couches inférieures de corriger toutes les erreurs). Ceci peut être réalisé en ajoutant un code de correction d'erreur (*forward error correction*, ou *FEC*) avec chaque paquet afin de permettre à chaque récepteur du paquet de le corriger lui-même. Cependant, cette solution, qui oblige à envoyer des données redondantes pour la correction des paquets, engendre un *overhead* non négligeable, même si un mécanisme adaptable (en fonction de la qualité du canal) est mis en place.

D'autres solutions sont envisagées pour limiter l'impact des erreurs de transmission ou de la perte de paquet, comme par exemple la retransmission des paquets erronés directement par la couche lien ou le fait d'accepter de recevoir les paquets dans un ordre approximatif (*almost in order delivery*).

Pour réaliser ces améliorations, diverses approches sont adoptées. Elles reposent soit entièrement sur la couche lien (*Snoop TCP*, *TCP-unaware Link Layer*), soit entièrement sur des solutions *end-to-end* (*ELN*, *WTCP*, *TCP Sack*⁴, *TTCP*) ou encore sur des approches hybrides (*ICTP*, *M-TCP*). Ces solutions sont décrites plus en détail dans [6].

3.3.2 AMELIORATIONS DE LA COUCHE MAC

La couche MAC traditionnelle de 802.11, celle implémentant l'algorithme CSMA/CA, révélant certaines insuffisances lors de l'utilisation avec des réseaux mobiles ad-hoc, de nombreuses alternatives, dont certaines sont classées dans la taxonomie présentée en Figure 3.23, ont été développées.

Une description complète de ces techniques étant disponible[6], nous nous attacheront uniquement à la présentation des plus remarquables.

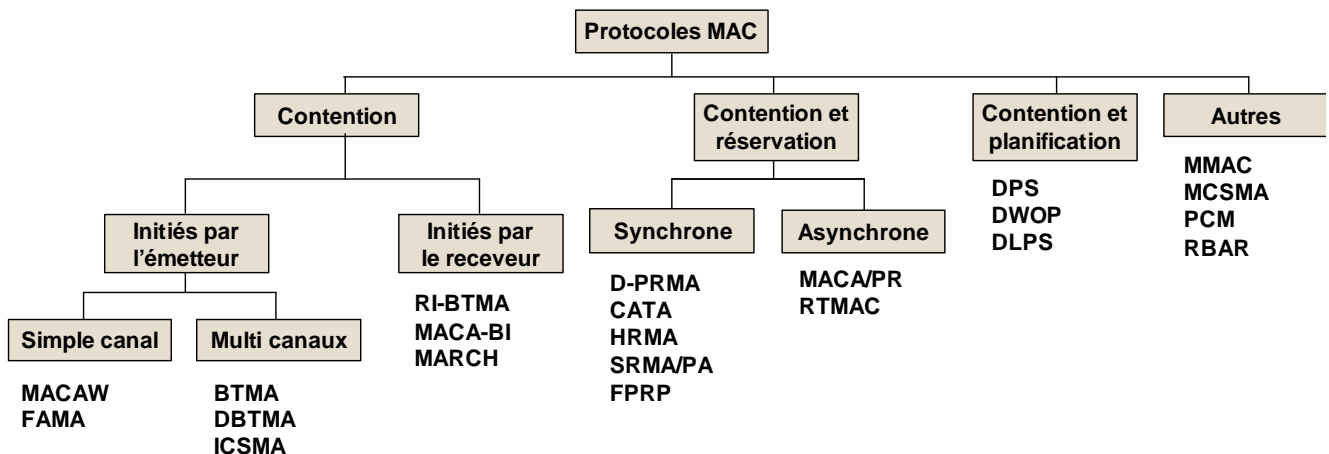


Figure 3.23 : Taxonomie des protocoles MAC ad-hoc

⁴ A noter que TCP-SACK est largement disponible (et utilisé) dans les couches TCP/IP standards qu'on trouve sur l'Internet.

Tableau 3.5: Solutions d'adaptation de TCP aux réseaux ad-hoc

| Problème | TCP-F | TCP-ELFN | TCP-BuS | ATCP | Split-TCP |
|--|------------------------------|-------------------------------|---|---|--|
| Perte de paquet (erreur ou collision) | Comme TCP | Comme TCP | Comme TCP | Retransmission sans contrôle de congestion | Comme TCP |
| Erreur de route | Mise en veille de l'émetteur | Mise en attente de l'émetteur | Mise en veille de l'émetteur | Comme TCP | Comme TCP |
| Paquets désordonnés | Comme TCP | Comme TCP | Paquets désordonnés atteints après rétablissement de la route | ATCP réordonne les paquets | Comme TCP |
| Congestion | Comme TCP | Comme TCP | Messages explicites (ICMP) utilisés | ECN utilisé pour notifier l'émetteur. Même contrôle de congestion que TCP | Contrôle de congestion au sein d'une zone (géré par des nœuds <i>proxy</i>) |
| Fenêtre de congestion après rétablissement de la route | Même qu'avant la rupture | Même qu'avant la rupture | Même qu'avant la rupture | Recalculée pour la nouvelle route | Les nœuds <i>proxy</i> maintiennent la fenêtre et gèrent la congestion |
| Notification explicite d'erreur de route | Oui | Oui | Oui | Oui | Non |
| Notification explicite du rétablissement de la route | Oui | Non | Oui | Non | Non |
| Dépendance vis-à-vis du protocole de routage | Oui | Oui | Oui | Oui | Non |
| Sémantique point à point | Oui | Oui | Oui | Oui | Non |
| Cache des paquets aux nœuds intermédiaires | Non | Non | Oui | Non | Oui |

3.3.2.1 AMELIORATIONS DU PROTOCOLE MACA

Une première série d'améliorations consiste en diverses techniques venant modifier le fonctionnement de MACA pour le rendre plus résistants aux problèmes liés à la topologie (MACAW) ou bien réduire l'*overhead* associé (MACA-BI, MARCH).

3.3.2.1.1 Multiple Access Collision Avoidance for Wireless (MACAW)

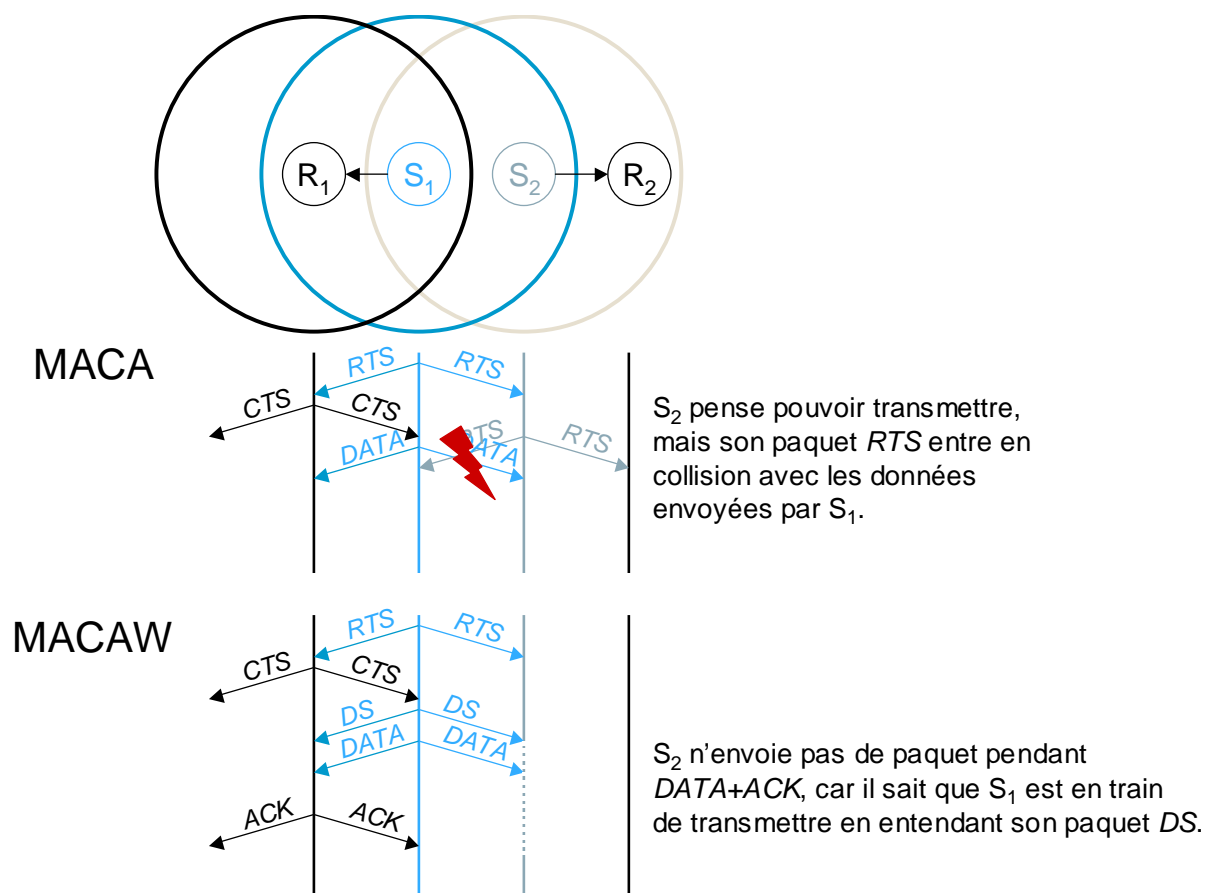
MACAW est une simple amélioration du protocole MACA décrit en 3.2.1.2.2. Il vise à résoudre le problème des nœuds exposés et cachés.

Les principales différences avec le mécanisme RTS/CTS introduit précédemment sont les suivantes :

- Une plus grande équité dans le mécanisme *BEB* introduit en associant un temps de garde par flot de données (chaque flot étant placé dans une queue différente du nœud) et en transmettant la valeur actuelle du compteur dans chaque paquet ;
- Meilleure gestion de la durée du temps de garde en ne remettant plus à la valeur initiale la valeur de la fenêtre de congestion après chaque transmission réussie, mais en utilisant une variation *MILD* (*multiple increase linear decrease*) ;
- Introduction de nouveaux paquets de contrôle : *ACK* modifié, permettant de gérer le recouvrement d'erreur au niveau de la couche lien (ce qui est plus rapide et supprime certains problèmes liés à TCP expliqués précédemment) ; *DS* qui acquitte l'échange RTS/CTS et précise la durée de transmission du paquet de données ; *RRTS* (*request for request to send*) qui permet de s'affranchir du problème du nœud exposé.

La

Figure 3.24 et la Figure 3.25 illustrent l'utilité de MACAW dans deux exemples de topologie.



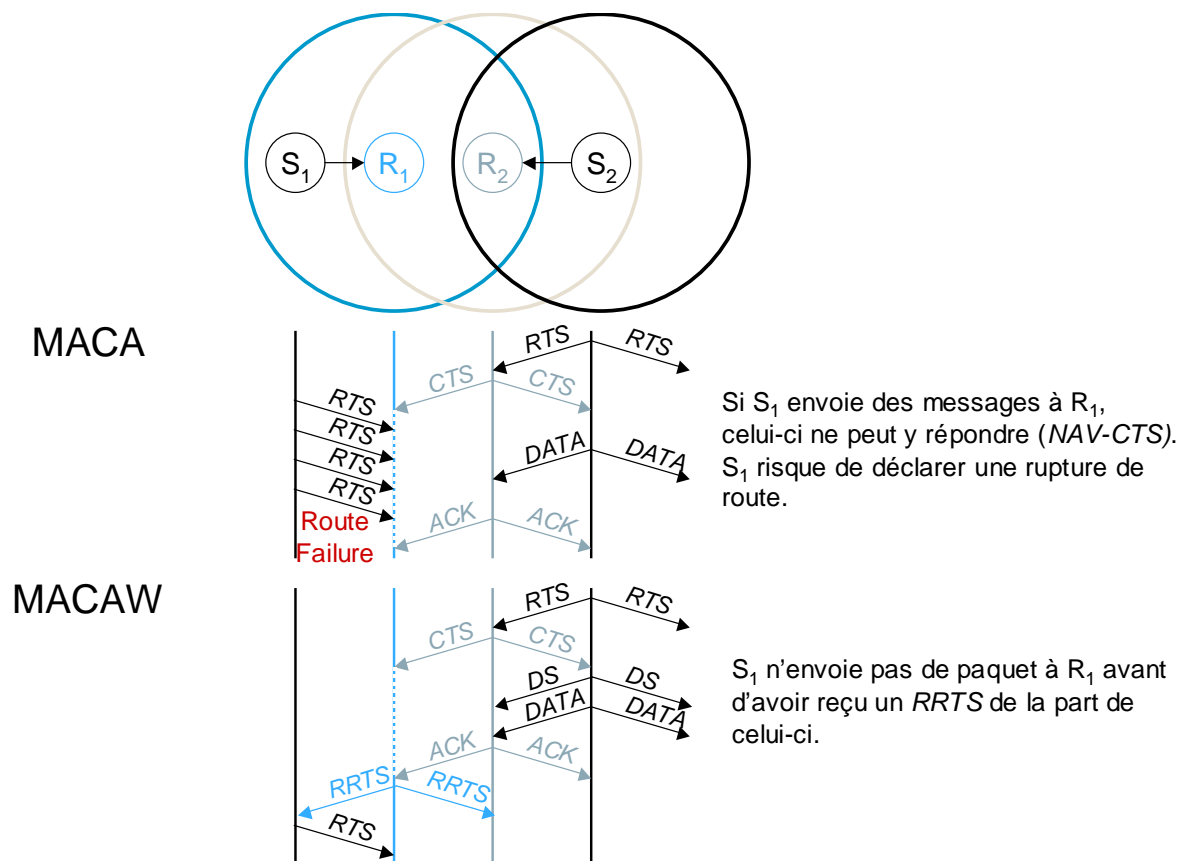


Figure 3.25 : Utilité du protocole MACAW (cas 2)

3.3.2.1.2 MACA-By Invitation (MACA-BI)

MACA-BI permet d'éviter l'utilisation des paquets *RTS* et *CTS* en laissant l'initiative de l'envoi au receveur qui initie la connexion avec un paquet *RTR* (*ready to receive*). Pour combattre le problème du nœud caché, le paquet *RTR* contient une estimation de la durée nécessaire à l'envoi du paquet de données, et donc pendant laquelle les stations voisines ne doivent pas émettre. Cette durée est calculée en utilisant des informations sur l'état du réseau contenues dans les paquets de données modifiés en conséquence.

3.3.2.1.3 Media Access with Reduced Handshake (MARCH)

Comme *MACA-BI*, *MARCH* laisse l'initiative de la transmission au receveur, mais il ne nécessite pas de mécanisme de prédiction sur la durée de transmission du paquet de données.

MARCH est vraiment dédié au routage ad-hoc puisqu'il tire parti du fait que les nœuds voisins entendent le message *CTS* pour n'utiliser le message de contrôle *RTS* que lors du premier saut, ce qui réduit grandement l'*overhead*.

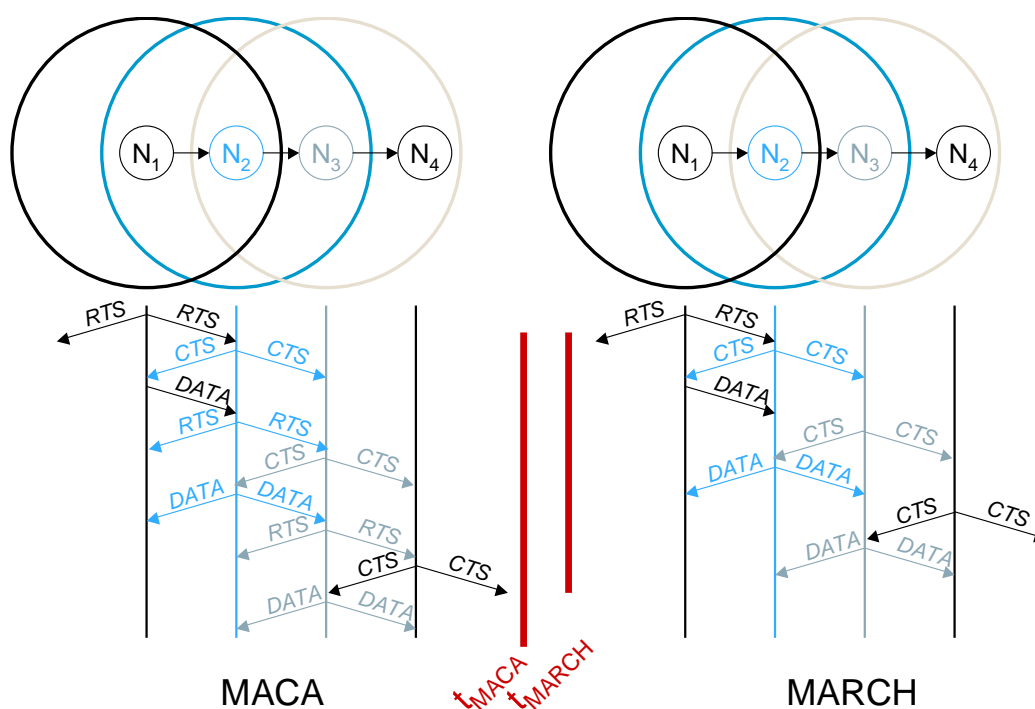


Figure 3.26 : Comparaison MACA/MARCH

3.3.2.1.4 MACA with Piggy-Backed Reservation (MACA/PR)

MACA/PR est une évolution du mécanisme *MACA* qui permet d'introduire une différenciation entre paquets temps réel (pour la voix ou la vidéo) et paquets *best effort* (pour les autres données). *MACA/PR* fonctionne sur le principe du *TDMA* (*time division multiple access*) en divisant le temps en différents slots.

Chaque slot est de taille variable et est réservé au niveau des nœuds de manière asynchrone. Chaque nœud tient à jour une table de réservation qu'il transmet et reçoit périodiquement de ses voisins. Ainsi, un nœud n'émet pas pendant un slot réservé par un voisin (ce qui évite le problème du nœud caché).

Les slots non réservés sont utilisés pour les données en *best effort* en utilisant le mécanisme *MACA* traditionnel.

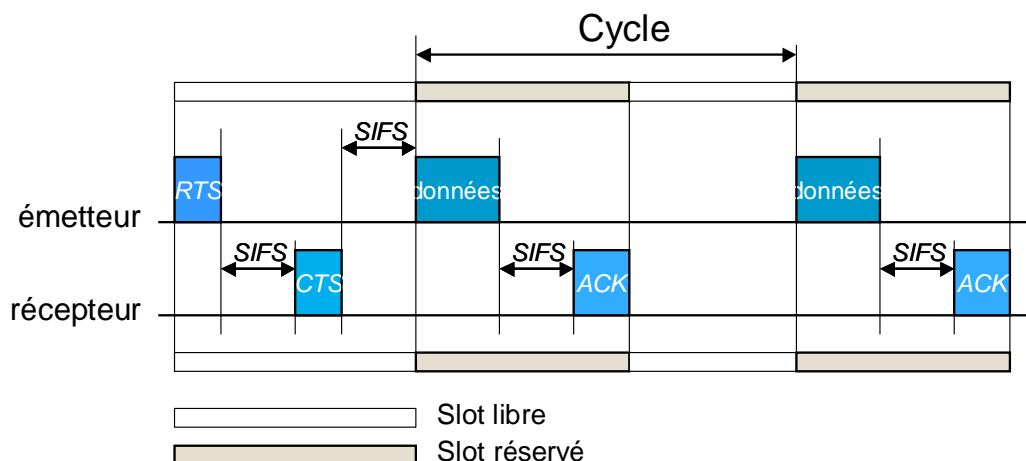


Figure 3.27 : Fonctionnement de MACA/PR

Pour les paquets temps réel, le fonctionnement est différent. Le premier paquet d'un flot est transmis en utilisant le mécanisme RTS/CTS traditionnel. Mais *MACA/PR* tient son nom du fait que les données envoyées contiennent, agrégées au message, les données de réservation pour le prochain slot. Aussi, l'envoi de données sert à réserver le slot pour le paquet suivant (qui commencera dans une durée arbitraire appelée *cycle*). Le paquet *ACK* contient également des données similaires et sert de confirmation de réservation de la part du récepteur.

3.3.2.2 DISTRIBUTED PACKET RESERVATION MULTIPLE ACCESS PROTOCOL (D-PRMA)

D-PRMA est une extension du protocole centralisé PRMA pour les réseaux ad-hoc. Il est basé sur un schéma TDMA et peut être utilisé, par exemple, pour favoriser le transport de la voix dans les réseaux sans fil multibonds.

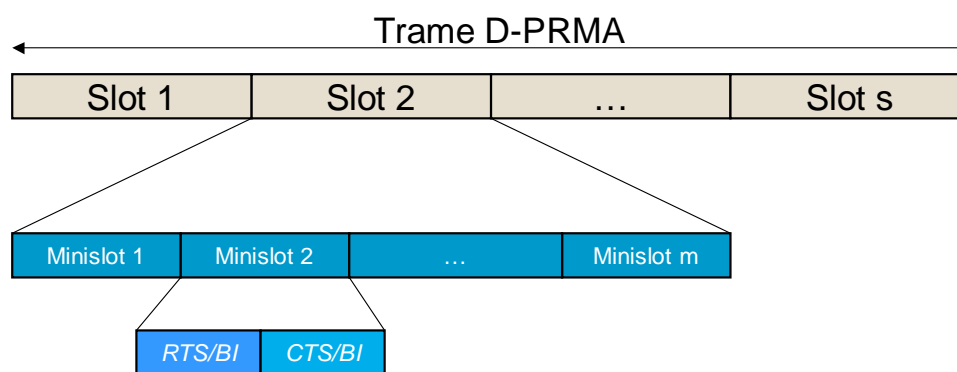


Figure 3.28 : Structure de trame D-PRMA

Dans *D-PRMA*, chaque trame est divisée en s slots, eux-mêmes divisés en m mini-slots, qui contiennent deux champs de contrôle (*RTS/BI* et *CTS/BI*, *BI* pour *busy indicator*). Au début de chaque slot, tous les nœuds souhaitant transmettre sont en contention. Dès qu'un nœud gagne, le reste des mini-slots du slot en cours lui est donné pour transmettre ses données.

Pour favoriser la voix par rapport aux autres données, les paquets de VoIP ont une probabilité 1 de gagner le premier slot (contre $p < 1$ pour les autres données). De plus, ils peuvent utiliser le même slot dans les trames suivantes pour transmettre tant que leur flot de données n'est pas terminé.

3.3.2.3 DISTRIBUTED PRIORITY SCHEDULING (DPS)

Le mécanisme DPS, qui repose également sur le contrôle d'accès au canal MACA, permet d'ajouter une notion de priorité entre les nœuds et de favoriser ainsi certaines transmissions sur d'autres (qualité de service).

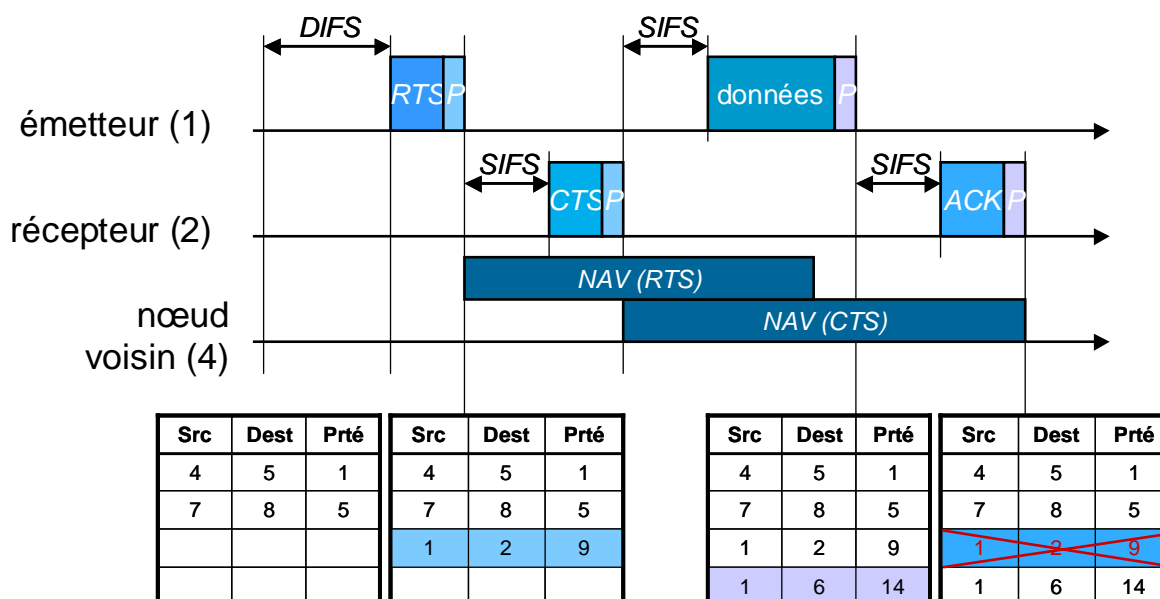


Figure 3.29 : Fonctionnement de DPS

Pour ce faire, chaque nœud agrège une mesure de priorité (qui peut être par exemple le délai maximal pour l'envoi) pour le paquet courant ou le prochain paquet, respectivement sur les messages de type *RTS* et *DATA*. Ces informations sont répétées par le nœud receveur (respectivement dans les messages *CTS* et *ACK*), ce qui permet aux nœuds cachés d'être également informé de la priorité des envois.

A partir des informations reçues depuis les voisins, chaque nœud tient donc à jour une table de planification des prochains paquets à envoyer avec la source, la destination et la mesure de priorité. Cette mesure est prise en compte lors du calcul du temps de garde pour l'accès au canal, ce qui assure statistiquement que les paquets les plus prioritaires seront envoyés en premier.

3.4 QUALITE DE SERVICE DANS LES RESEAUX MOBILES AD-HOC

Les différents protocoles de contrôle d'accès au canal définis dans les paragraphes précédents ont introduit la notion de flux de données prioritaires, voix et vidéo notamment, pour lesquels une partie de la bande passante devait être réservée, de manière garantie ou statistique.

Cette volonté va de mise avec le développement d'application de voix sur IP et de vidéoconférence utilisant des terminaux sans fil. Dans le cas des réseaux ad-hoc, ces applications pourraient remplacer avantageusement les actuels standards de communication (radio, téléphonie cellulaire) car elles se passeraient alors d'infrastructure tout en offrant des services supplémentaires.

Devant ces enjeux, l'*ietf* a mis au point la norme 802.11e qui modifie la couche MAC du standard 802.11 afin d'y introduire de la qualité de service.

3.4.1 LA NORME 802.11E

Cette norme, récemment adoptée par l'IEEE, permet de différencier plusieurs flots au sein de chaque nœud tant en terme de débit qu'en terme de délai.

Pour ce faire, 802.11e introduit un nouveau standard pour la couche MAC appelé HCF (*hybrid coordination function*) qui définit lui-même deux mécanismes de contrôle d'accès : *HCF controlled channel access (HCCA)* qui permet un accès sans contention et *enhanced distributed channel access (EDCA)* qui est basé sur un accès en contention.

Aussi, une trame 802.11e alterne périodes sans contention (où *HCCA* est utilisé seul) et périodes avec contention où *HCCA* et *EDCA* coexistent.

Comme le standard n'est pas dédié aux réseaux ad-hoc, *HCCA* (qui est une version de *PCF* améliorée pour gérer la QoS) nécessite un coordinateur et ne peut être utilisé dans les réseaux d'égal à égal. Aussi, tous les nœuds d'un réseau ad-hoc doivent implémenter et ne peuvent utiliser que *EDCA*.

Pour gérer la qualité de service, l'algorithme *EDCA* définit différentes catégories d'accès (ou *AC*, pour *access categories*) qui ont une file d'attente différente au sein de chaque nœud et différents paramètres qui leur assurent une priorité différente.

Tableau 3.6: Paramètres de priorité dans 802.11e (cf. [13])

| Priorité | Désignation | Catégorie d'accès | AIFSN | CW _{min} | CW _{max} | Limite TXOP (ms) |
|----------|-------------------------------------|-------------------|-------|-------------------|-------------------|------------------|
| 1 | BK (Arrière-plan) | AC_BK | 7 | 15 | 1023 | 0 |
| 2 | BK (Arrière-plan) | AC_BK | 7 | 15 | 1023 | 0 |
| 0 | BE (<i>Best effort</i>) | AC_BE | 3 | 15 | 1023 | 0 |
| 3 | EE (Vidéo <i>excellent effort</i>) | AC_BE | 3 | 15 | 1023 | 0 |
| 4 | CL (Vidéo charge contrôlée) | AC_VI | 2 | 7 | 15 | 3,008 |
| 5 | VI (Vidéo) | AC_VI | 2 | 7 | 15 | 3,008 |
| 6 | VO (Voix) | AC_VO | 2 | 3 | 7 | 1,504 |
| 7 | NC (Contrôle du réseau) | AC_VO | 2 | 3 | 7 | 1,504 |

Ainsi, le temps d'espace entre trame (*DIFS*) n'est plus constant mais est fonction de la priorité donnée au flux auquel appartient le prochain paquet à transmettre. Cette valeur, appelée *AIFS* (*access interframe spacing*), est calculée comme suit :

$$AIFS(ac) = SIFS + AIFSN(ac) \times t_{slot} \quad (\text{eq. 3.7})$$

Aussi, plus le flux est prioritaire, plus l'*AIFS* correspondant est petit, ce qui donne statistiquement une plus grande chance d'accès au canal.

Un autre aspect permettant de réserver une plus grande bande passante aux flux les plus prioritaires est l'introduction du *contention free bursting*, période pendant laquelle un flux qui a acquis le canal par le mécanisme *EDCA* peut continuer à émettre sans contention. Cette période, également appelée *opportunité de transmission (TXOP, pour transmission opportunity)* est limitée selon le niveau de priorité attribué au flux.

Afin d'augmenter encore le débit utile du canal, 802.11e définit également une politique d'acquittement *NO ACK* où les messages de données ne sont plus confirmés par l'envoi d'un paquet *ACK*. Cette technique est particulièrement avantageuse dans le cas de canaux avec un très faible taux d'erreur.

3.4.2 PERFORMANCES DE 802.11E

Comme précisé auparavant, le protocole 802.11e n'a pas été exclusivement développé pour les réseaux mobiles ad-hoc. Aussi, il est intéressant de voir comment se comporte ce protocole dans différentes situations, plus ou moins favorables. Dans [14], les auteurs évaluent les variations du débit en fonction de paramètres comme la taille des paquets de données, le nombre de nœuds et la qualité de la liaison. Dans [13], les auteurs s'attachent à étudier le comportement du protocole dans divers scénarios de réseau ad-hoc multibonds.

3.4.2.1 ANALYSE DES PERFORMANCES

Le premier document ([14]) procède à une analyse des performances de 802.11e en terme de débit utile dans divers scénarios qui sont simulés.

3.4.2.1.1 Conditions de simulation

Pour réaliser les simulations, les auteurs ont utilisé une implémentation du 802.11e sur le simulateur événementiel OPNET qui utilise un modèle de 802.11b pour la couche physique. Dans tous les tests, des messages en *constant bit rate (CBR)* sont envoyés par *UDP*. La taille de ces messages dépend de l'expérience menée. Sauf indication contraire, les liens entre les stations sont considérés comme parfaits.

Notons que dans le document présenté, assez peu d'information sur la topologie des nœuds du réseau est donnée, ce qui limite grandement la portée des résultats.

3.4.2.1.2 Résultats de simulation

Les résultats présentés évaluent respectivement l'impact de la taille des paquets de données, du nombre de stations, des différents niveaux de priorité et de la qualité du lien sur le débit de données.

3.4.2.1.2.1 Impact de la taille des paquets de données

Les mesures portent sur le débit en amont de la couche MAC (la taille des en-têtes des couches MAC et PHY ne sont donc pas prises en compte). L'*overhead mesuré* ne tient donc compte que des messages de contrôle échangés et des différents temps d'attente lors de l'accès au canal. Pour *EDCA*, les messages ont été envoyés avec la priorité *AC_VO*.

Bien entendu, le débit des données augmente quand la taille des paquets transmis augmente. En terme de débit, *HCCA* est légèrement supérieur à *EDCA*, tous deux se montrant plus performants que le traditionnel *DCF*.

D'autre part, la suppression des acquittements des messages de données (politique *NO ACK*) permet de gagner au maximum entre 15% et 20% de débit (respectivement pour *HCCA* et *EDCA*) pour une taille de paquet entre 250 et 400 octets. Cette différence est grandement réduite pour *EDCA* avec de grands paquets (2500 octets), puisqu'elle tombe à 3% environ.

Cette chute est due au fait qu'en mode *EDCA*, le paquet *ACK* ne représente qu'une partie de l'*overhead* de contrôle (avec les paquets *RTS* et *CTS*). Quand les paquets de données sont petits

(de la taille des paquets de contrôle), sa disparition est notable (environ 20% de différence). Quand la taille des paquets de données devient grande devant celle des paquets de contrôle, la disparition des paquets *ACK* ne représente qu'une très faible variation de l'*overhead*.

3.4.2.1.2.2 Impact du nombre de stations

L'impact du nombre de stations est mesuré en simulant l'envoi d'un trafic de 2Mbits/s en *CBR* avec des paquets de 1500 octets par un nombre croissant de stations, toujours en utilisant la couche physique de 802.11b (soit 11Mbits/s pour les données et 2Mbits/s pour les paquets de contrôle). Les mesures ont été réalisées pour différents types de contrôle d'accès et différents niveaux de priorité (un niveau de priorité par test).

Les résultats présentés par cette simulation sont assez intéressants puisque, dans le cas de l'utilisation de l'algorithme *EDCA*, le niveau de priorité le plus haut n'est pas forcément celui qui obtient les meilleures performances à nombre de stations égal.

En particulier, pour un grand nombre de stations (supérieur 7 ou 8 stations en interférence) le débit réel lors de l'utilisation du niveau de priorité *AC_VO* (pourtant le plus prioritaire) chute dramatiquement jusqu'à atteindre moins de 20% pour 30 stations en interférence. Ce phénomène est dû aux **bornes de la fenêtre de contention qui sont extrêmement proches à ce niveau de priorité, ce qui ne permet plus d'éviter les collisions** (même statistiquement) quand le nombre de stations en portée dépasse 5.

3.4.2.1.2.3 Impact des niveaux de priorité

Dans ce scénario un nombre croissant de stations émet du trafic pour chacun des quatre niveaux de priorité d'*EDCA*. Chaque flot a un débit de 250Kbits/s (soit 1Mbit/s au total émis par station). Cette étude va permettre de voir l'évolution du débit pour chaque niveau de priorité en fonction du nombre de stations (et donc de la charge totale du canal).

La simulation fait apparaître trois zones de fonctionnement :

- *Zone de faible charge (< 5 nœuds)* : toutes les priorités ont le même débit (qui suit la croissance normale du débit par *AC* avec le nombre de stations). Le canal étant faiblement chargé, il n'est pas nécessaire de privilégier les flux prioritaires aux dépens des flux d'arrière plan ;
- *Zone de moyenne charge (entre 5 et 10 nœuds)* : le débit des flux prioritaires continue d'augmenter mais, pour privilégier cette augmentation par rapport aux flux moins importants, ces derniers voient leur débit global diminuer. Il est clair que le protocole 802.11e parvient à maintenir la qualité de service des flux prioritaires aux dépens des flux d'arrière-plan ;
- *Zone de forte charge (> 10 nœuds)* : Dans cette zone de fonctionnement, le débit des flux d'arrière plan est quasiment nul, voir nul au-delà de 15 stations. Le débit des flux prioritaires pourrait augmenter jusqu'à saturation du canal. Au contraire, il diminue assez rapidement (pour retrouver par exemple à 15 stations le débit atteint avec 5 stations émettrices seulement). Le protocole *EDCA*, et notamment la petite fenêtre de contention attribuée aux flux prioritaires, montre ses limites : **il n'est performant que si la topologie n'est pas très dense (au maximum 5 à 10 stations dans une zone d'interférence donnée).**

3.4.2.1.2.4 Impact d'un lien faible

Ce scénario simule l'éloignement d'une des stations qui en conséquence change sa modulation et son débit d'émission (de 11Mbits/s à 5,5Mbits/s, 2Mbits/s ou 1Mbit/s). Les données sont envoyées avec la priorité *AC_VO*.

La simulation montre que la dégradation du débit lié aux changements de modulation sur le lien faible n'impacte le débit réel que de 10% à 20% (selon l'ampleur de la baisse du débit d'émission). Cette faible variation s'explique par le fait que le mécanisme *EDCA* donne le même accès au canal indépendamment du schéma de modulation. En conséquence, la station émettant à un débit plus faible a la même probabilité d'accès au canal que les autres stations. Mais comme elle émet moins rapidement mais possède la même valeur de *TXOP*, elle doit simplement fragmenter ses données en éléments plus petits afin de respecter cette durée limite de monopolisation du canal.

Notons qu'aucun résultat n'est fourni sur l'effet d'un lien faible lors de l'utilisation d'autres niveaux de priorité.

3.4.2.2 COMPOTEMENT AVEC DES RESEAUX AD-HOC MULTIBONDS

Ce document ([13]) s'intéresse plus particulièrement aux performances de 802.11e dans des réseaux mobiles multibonds. Aussi, il ne s'intéresse qu'au mécanisme *EDCA* dont il évalue les performances en terme de débit réel et délai d'acheminement des paquets par simulation.

3.4.2.2.1 Conditions de simulation

Dans cette étude, deux scénarios sont mis au point et implémentés sous le simulateur ns-2 : un scénario dit « statique » (dont la topologie est décrite en Figure 3.30) et un scénario dit « mobile » où 50 stations se déplacent de manière aléatoire à une vitesse de 5m/s dans un rectangle de 1900x400 mètres. Dans tous les cas, le protocole AODV est utilisé pour router des messages de 512 octets transmis par UDP sur un médium physique respectant la norme 802.11g.

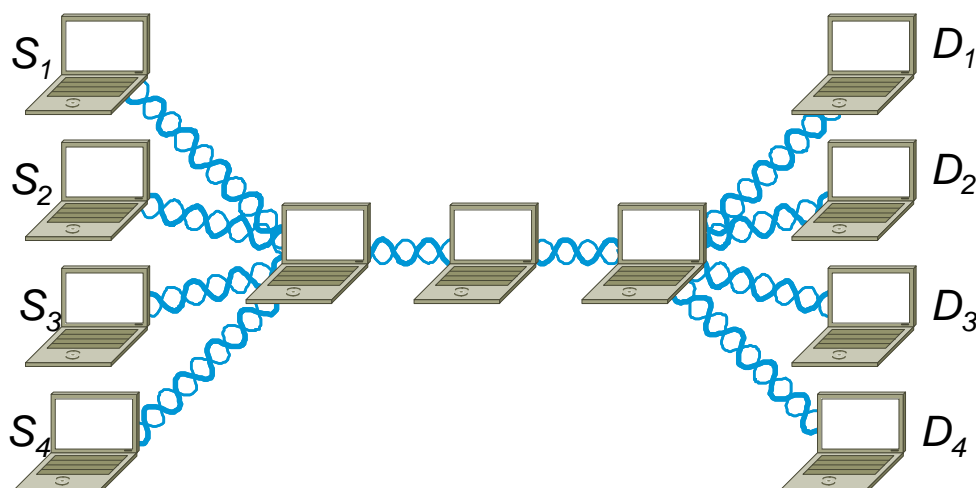


Figure 3.30 : Scénario statique de test d'efficacité de 802.11e

3.4.2.2.2 Résultats de simulation

Les principaux résultats de simulation concernent la détermination des limites de saturation, l'impact du nombre de nœuds, l'impact de la mobilité sur les performances et l'influence de la présence de stations n'implémentant pas 802.11e parmi les nœuds du réseau ad-hoc.

3.4.2.2.2.1 Limite de saturation

Cette simulation a été réalisée en utilisant le scénario statique avec et sans *contention free bursting* (CFB). Chacune des stations émettrices (S_i) émet un trafic de plus en plus important et qui est réparti équitablement entre les différents niveaux de priorité (de 0 à 8Mbps/s par AC, soit de 0 à 32 Mbps/s par nœud).

En terme de débit de données, on observe toujours une première phase (*zone de faible charge*) où les différents flux ont tous le même débit réel et une seconde phase (*zone de moyenne charge*) où les flux prioritaires prennent le dessus et où les flux d'arrière-plan ont peu voir plus accès au canal. Par contre, on n'observe plus d'affaissement du débit quand le trafic augmente. Ceci est du à la topologie retenue (beaucoup moins dense qu'en 3.4.2.1.2.2) qui permet au mécanisme EDCA d'éviter un grand nombre de collisions, même avec des bornes de fenêtre de contention très rapprochées (comme pour les flux AC_VO).

Les données concernant les délais de transmissions coïncident avec les résultats concernant les débits. Au-delà de 4Mbps/s par AC (soit 12Mbps/s par nœud), les délais des flux secondaires sont rédhibitoires (100 s) et correspondent à des *timeouts* alors que les délais des flux prioritaires se stabilisent autour de 0,1s

L'introduction de CFB améliore légèrement les performances en terme débit pour les flux prioritaires, mais n'empêche pas la « famine » des flux d'arrière-plan et n'a pas d'influence notable sur les délais.

3.4.2.2.2.2 Impact du nombre de nœuds

L'influence du nombre de nœuds a été mesurée en modifiant le nombre de stations dans la « ligne centrale » entre les émetteurs et les récepteurs. Les messages ont été émis avec un débit de 4Mbps/s par AC.

Le trafic total, en augmentant le nombre de nœuds de la ligne de 1 à 8, passe de 16,60Mbps/s à 3,32Mbps/s (sans CFB) et de 21,74Mbps/s à 3,34Mbps/s (avec CFB). Donc, plus le nombre de sauts augmente, plus les performances s'en ressentent et plus l'effet du mécanisme CFB s'estompe. Il est donc impératif de rechercher les routes minimisant les sauts afin d'optimiser les performances.

De plus, alors que la répartition reste stable sans CFB, l'utilisation de ce mécanisme conduit à une augmentation de la proportion de trafic allouée aux flux secondaires (AC_BK et AC_BE) aux dépens des flux prioritaires (AC_VI et AC_VO).

3.4.2.2.3 Impact de la mobilité

L'influence de la mobilité a été étudiée en comparant, sur les bases du scénario « mobile » les débits et les délais de transmission avec des nœuds mobiles et des nœuds immobiles envoyant des paquets à un débit de 0,2Mbit/s par AC.

L'introduction de la mobilité conduit à un débit plus faible que le débit maximal théorique, même dans la zone de faible charge. Ceci est dû à la plus grande diversité dans les chemins à suivre et aux erreurs de routage dues à des chemins dont la validité a expiré et qui n'ont pas encore été mis à jour par le protocole.

Là encore, le comportement est très différent avant et après saturation du réseau. En deçà de la limite de saturation, on constate que les performances (surtout en terme de délai d'acheminement) sont les mêmes quelle que soit la priorité donnée au flux. Cette limitation dans la gestion de la qualité de service peut être imputée au protocole de routage utilisé dans la simulation, AODV, qui est protocole réactif et qui nécessite donc la redécouverte du chemin de routage avant chaque envoi et la mise en cache des paquets au niveau des nœuds, ce qui uniformise vers le haut les délais d'acheminement.

Par contre, au-delà de la limite de saturation, **l'introduction de la mobilité permet une meilleure équité entre les flux**. Alors que les simulations de réseaux statiques ont montré qu'au-delà de la limite de saturation du réseau les flux d'arrière-plan étaient « écrasés » par les flux prioritaires et ne pouvaient plus émettre, **la mobilité permet de conserver un débit limité mais non nul pour les données secondaires**.

Là encore, les résultats semblent dépendants du protocole de routage utilisé, et il serait intéressant de mener des simulations similaires en utilisant un protocole proactif comme OLSR.

3.4.2.2.4 Impact de stations hétérogènes

Cette simulation reprend le scénario « mobile » en ajoutant un nombre croissant de stations n'implémentant pas 802.11e mais uniquement 802.11 (appelées *stations simples*) et analyse l'évolution du débit par AC et du délai d'acheminement par AC en fonction de ce facteur.

Les résultats montrent que l'introduction de 10% de stations simples suffit à faire dramatiquement chuter les performances, notamment en terme de délai. La pire configuration est atteinte quand 80% des stations n'implémentent pas 802.11e (les performances sont alors pires qu'avec uniquement des stations simples). **En conséquence, l'introduction de la norme 802.11e doit se faire manière coordonnée et doit être envisagée dès le début du déploiement des équipements comme la couche MAC utilisée par défaut.**

4. ENVIRONNEMENT DE SIMULATION

Cette section décrit l'outil utilisé pour la mise en place des scénarios de test et des mesures de performances ainsi que l'architecture des nœuds utilisés pour la simulation.

4.1 OUTILS DE SIMULATION

L'outil de simulation retenu est OMNeT++. Il s'agit d'un simulateur de réseau à événements discrets dans lequel les différents éléments du réseau communiquent par envoi de messages.

Il est développé en C++ et se distingue par son orientation objet et l'utilisation de modules hiérarchisés qui permet une grande flexibilité dans la création de nœuds complexes au sein du réseau.

OMNeT++ peut donc être utilisé pour :

- La modélisation de trafic de réseaux de communication
- La modélisation de protocoles
- La modélisation de réseaux de files d'attente

Afin de gérer la mobilité des nœuds et le routage ad-hoc, OMNeT++ est utilisé avec le *Mobility Framework* dont l'implémentation fournit une gestion dynamique de la connectivité et de la mobilité des nœuds, ainsi que le support d'un canal de connexion sans fil.

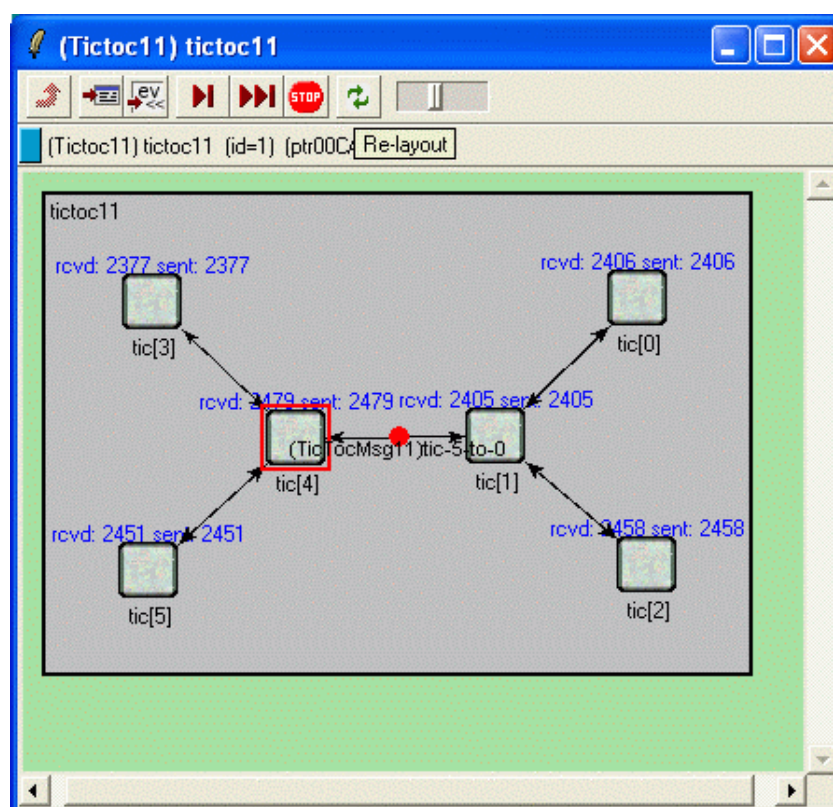


Figure 4.1 : L'environnement de simulation *tkenv*

4.1.1 OMNeT++

Un modèle de simulation OMNeT++ consiste en une série de modules imbriqués hiérarchiquement. La profondeur d'imbrication n'est pas limitée, ce qui permet de refléter, dans la structure du modèle de simulation, la structure logique du système réel.

Les différents modules communiquent par envoi de messages qui peuvent contenir des structures de données complexes. Ces messages peuvent être envoyés directement d'un module à un autre (pour des messages de commande par exemple) ou suivre un chemin défini passant par des portes (l'équivalent des ports de communications) connectées à un autre module (pour simuler par exemple le parcours d'un paquet de données à travers les différentes couches d'un nœud du réseau).

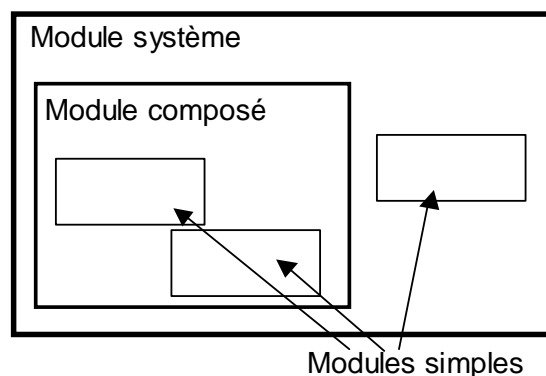


Figure 4.2 : Architecture modulaire de OMNeT++

Chaque module possède ses propres paramètres, qui peuvent être hérités des modules supérieurs. Les modules de plus bas niveau, appelés « modules simples », définissent le comportement du module composé auquel ils appartiennent. Ils sont programmés en C++ en utilisant la librairie de simulation.

Le rendu des simulations peut utiliser soit la ligne de commande (où on utilisera alors un outil pour enregistrer la trace laissée par les différents événements comme les envois de message) soit une interface graphique permettant de visualiser l'évolution du réseau au cours de la simulation (changements de topologie, d'état des liens, parcours des messages).

OMNeT++ implémente donc les outils de base pour la réalisation de simulations, mais il ne fournit aucun composant spécifique à la simulation d'un réseau de communications. Pour ce faire, il est donc intéressant d'utiliser un des différents *frameworks* disponibles, par exemple le *mobility framework*, afin de faciliter la réalisation d'un environnement de simulation complet.

4.1.2 MOBILITY FRAMEWORK

Le *mobility framework* permet de développer des simulations de réseaux sans-fil et mobiles sous OMNeT++. En plus du support de la mobilité des nœuds et d'une gestion dynamique des connections sur un canal sans-fil, il propose une série de modules *basiques* qui peuvent être dérivés afin de réaliser facilement des modules spécifiques réalisant les différentes fonctions d'une communication mobile (routage, gestion de l'accès au canal, couche physique, etc...)

Les deux principaux composants du *mobility framework* sont une architecture pour le support de la mobilité et des connections dynamiques et un modèle de nœud mobile pour OMNeT++.

Le module *ChannelControl* maintient toutes les connexions potentielles entre les nœuds (c'est à dire des nœuds qui sont dans leur zone d'interférence respectives). A la différence des liaisons traditionnelles d'OMNeT++, un lien entre deux nœuds du *mobility framework* ne signifie pas que ces derniers pourront correctement échanger des données (c'est la couche physique de chaque nœud qui décide si un paquet est correct, erroné ou considéré comme du bruit et ce en fonction du modèle physique retenu). En fait, ne sont pas connectés les nœuds qui n'interfèrent pas l'un avec l'autre.

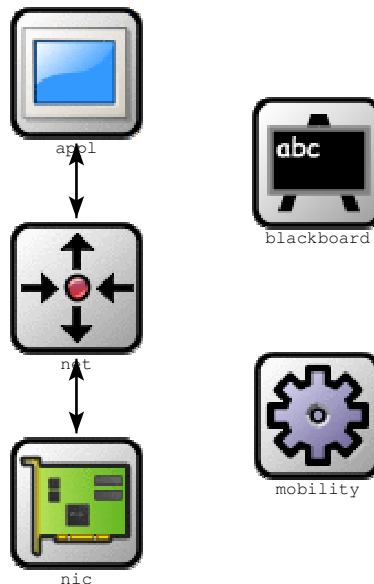


Figure 4.3 : Architecture d'un nœud du mobility framework

La structure interne d'un nœud du *mobility framework* est décrite en Figure 4.3. En plus des modules représentant les différentes couches ISO/OSI, il existe un module de mobilité (qui fournit la position courante du nœud et gère sa mobilité) et un module appelé *blackboard* qui est utilisé pour la communication de paramètres entre les couches (comme par exemple le niveau d'énergie actuel du nœud ou bien encore son état radio : émission, réception).

La version retenue du *mobility framework* (1.0a6) fournit une implémentation de la couche MAC et physique de 802.11 utilisée, avec diverses modifications, dans l'environnement de simulation. Par contre, aucun protocole de routage n'existant, une implémentation d'OLSR a été spécifiquement développée.

4.2 ARCHITECTURE DE LA SIMULATION

L'architecture des nœuds utilisés dans les différents scénarios de simulation repose sur l'utilisation de différents modules qui émanent directement des outils de simulation utilisés (comme le *mobility framework*), qui ont été élaborés pour des simulations précédentes ou qui ont été spécialement développés.

L'architecture de simulation est présentée en Figure 4.4 et le fonctionnement des différents modules est décrit dans les parties suivantes du présent document.

RAPPORT DE STAGE DE TROISIEME ANNEE

Forme d'onde Wi-Fi/OLSR pour réseaux ad-hoc tactiques

4.2.1 GENERATEUR DE TRAFIC

La génération du trafic est assurée par le module *USER* qui est implémenté par chaque nœud du réseau. Il permet d'insérer dans le réseau différents flux décrits dans un fichier de configuration et d'ainsi simuler un trafic généré par diverses applications réseau.

Le générateur de trafic a été initialement développé pour d'autres environnements de simulation et a donc nécessité plusieurs adaptations afin de s'intégrer au mieux dans l'environnement utilisant le *mobility framework*.

4.2.1.1 PRINCIPE DE FONCTIONNEMENT

Le module *UsagerModule*, qui fonctionne comme la source de trafic pour le nœud, est associé au fichier de commande de trafic par une classe imbriquée dans *UsagerModule* nommée *CommandFileHandler*.

Parmi tous les nœuds du réseau, le premier créé lit le fichier de commande précisé dans les paramètres de la simulation et envoie à chaque nœud, au moyen de messages de type *CommandLine*, des informations concernant le trafic qu'il auront à émettre (instant de départ, nœud de destination, type de trafic, périodicité, taille des paquets, etc...). Ces messages de commandes sont envoyés avant le lancement de la simulation proprement dite grâce à la commande *sendDirect* qui permet d'adresser directement un message d'un module à un autre sans passer par la suite des portes reliant éventuellement ces deux modules.

Le détail du fichier de commande et les différents paramètres du trafic pouvant être émis sont décrits en section 4.2.1.2.

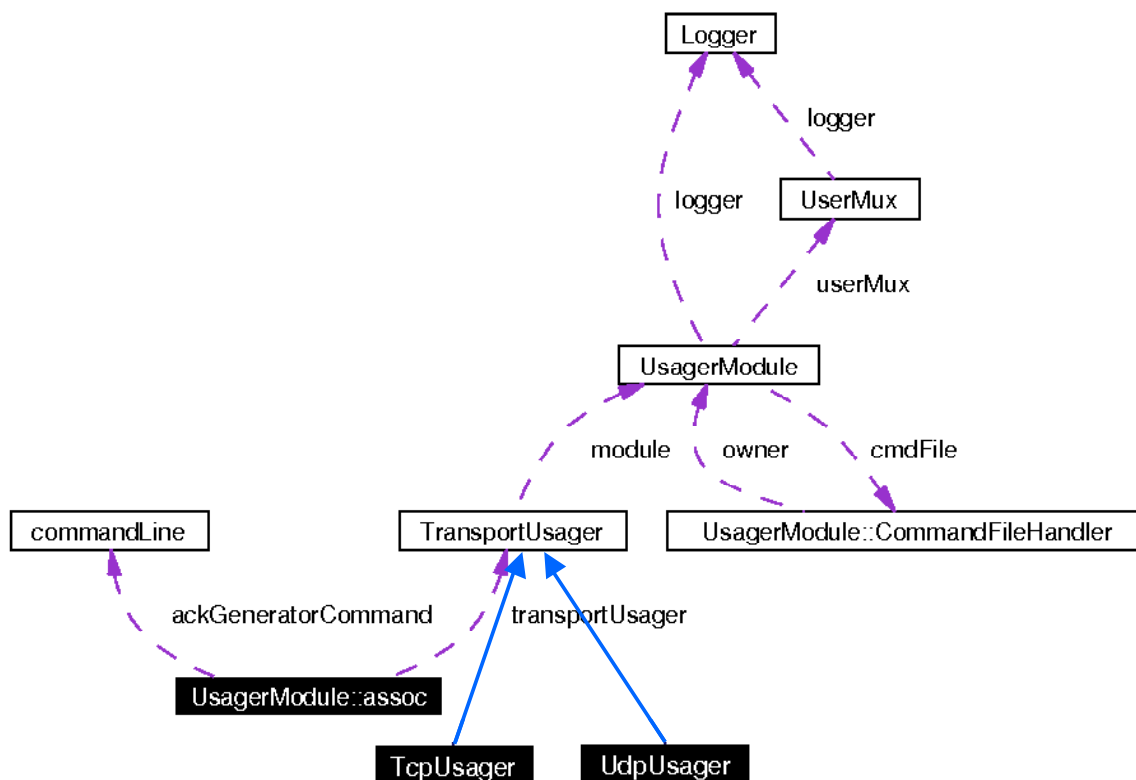


Figure 4.5 : Diagramme de collaboration du générateur de trafic

Le module *UserMux*, sous-jacent au précédent, permet de gérer le multiplexage entre les différents flux qui peuvent être émis depuis ce nœud ou à destination de celui-ci. Pour ce faire, *UsagerModule* et *UserMux* sont reliés par un couple de vecteurs de 16 portes, qui sont dynamiquement créées, allouées à des flux ou libérées en fonction de l'état du trafic émis ou reçu par un nœud.

Puis les modules de type *TransportUsager* (*TcpUsager* ou *UdpUsager*) créent, en fonction du type de trafic, des modules client ou serveur, TCP ou UDP, afin de générer un trafic correspondant au type d'application souhaitée. En instanciant différemment ces modules client ou serveur, il est donc possible de simuler le comportement de diverses applications réseau.

Les paquets ainsi émis sont ensuite transmis aux modules représentant les couches inférieures du modèle ISO/OSI (couche transport : *TcpModule* ou *UdpModule*, couche réseau-IP : *IpModule*) où ces paquets sont notamment encapsulés (avec des tailles d'en-tête correspondant aux valeurs réelles des en-têtes TCP, UDP et IP dont les valeurs sont détaillées dans le Tableau 4.1).

Tableau 4.1 : En-têtes ajoutés dans les différents modules d'un nœud de simulation

| En-tête | Module associé | En-tête (bits) |
|---------|-----------------------|----------------|
| UDP | <i>UdpModule</i> | 64 |
| TCP | <i>TcpModule</i> | 160 |
| IP | <i>IpModule</i> | 160 |
| - | <i>MFCLayer</i> | 0 |
| - | <i>*NetwLayer</i> | 0 |
| MAC | <i>Mac80211</i> | 272 |
| PLCP | <i>DebugEval80211</i> | 192 |

Le module *DropDelay* permet d'ajouter un délai lors de l'arrivée de paquets (pour simuler un temps de propagation ou une durée de traitement des données) et une probabilité de perte de paquet. Ces mêmes fonctions étant assurées par d'autres modules issus du *mobility framework*, les paramètres de simulation font que ce module transmet simplement les paquets au module *IpModule* sans interférer.

Enfin, le module *MFCLayer* pour *Mobility Framework Compatibility Layer* permet d'assurer la compatibilité entre le générateur de trafic et le *mobility framework*. Les détails de ce module sont décrits dans la section 0.

Tous ces modules écrivent, via l'utilisation d'un objet de type *Logger*, les traces de leur activité (réception, traitement, envoi de messages) dans un fichier de log séparé ou commun à tous les modules et dont les noms respectifs sont précisés dans la configuration de l'environnement de simulation.

4.2.1.2 FICHIER DE DESCRIPTION DU TRAFIC

L'objectif de la campagne de simulation est de comparer les performances d'une forme d'onde basée sur un accès au canal en contention (CSMA/CA) et sur un routage OLSR à celles d'une autre approche, développée par ailleurs.

Aussi, afin de reproduire les mêmes scénarios, il était essentiel d'utiliser les mêmes fichiers décrivant le trafic au sein du réseau. Un exemple d'un tel fichier est donné en Figure 4.6 et le Tableau 4.2 précise la signification, le type et l'utilisation faite de chaque variable.

| | | | | | | | | | | | |
|---------------------------|--------|---------|-----|------------|---------|-------|------|----|-------|-------|------|
| #Fichier de trafic usager | | | | | | | | | | | |
| #Date : 13/06/2006 | | | | | | | | | | | |
| #time | MAC | srcport | cmd | FEC | dstport | proto | mode | nb | T(ms) | bits | prio |
| 20000 | 0x000A | 0x0001 | 1 | 0x10240000 | 0x0002 | 17 | 2 | 4 | 100 | 12500 | 5 |
| 30000 | 0x0006 | 0x0003 | 1 | 0x10240008 | 0x0004 | 17 | 2 | 3 | 100 | 12500 | 5 |
| 35000 | 0x0009 | 0x0001 | 1 | 0x10240000 | 0x0002 | 17 | 2 | 1 | 100 | 12500 | 5 |

Figure 4.6 : Exemple de fichier de trafic

Tableau 4.2 : Interprétation du fichier de trafic

| Variable | Type | Valeurs | Utilisation |
|----------------|----------------|---|---|
| <i>time</i> | Entier décimal | $n \geq 0$ | Instant en nombre de slots écoulés auquel la commande s'exécute (NB : cette valeur doit être convertie en secondes dans le cadre de la simulation) |
| <i>MAC</i> | Entier hexa. | $0 < n < \text{nb. nœuds}$ | Adresse MAC du nœud émetteur du trafic (voir l'adressage du réseau 4.2.1.3.1) |
| <i>srcport</i> | Entier hexa. | $n > 0$ | Numéro de port utilisé par la source du flux |
| <i>cmd</i> | Entier décimal | 0=ARRET_FLUX 1=DEMARRER_FLUX 2=EMISSION_DONNEES | |
| <i>FEC</i> | Entier hexa. | $0 < n < \text{nb. nœuds}$ | Seuls les deux derniers octets sont utilisés dans la simulation : Adresse MAC du nœud destinataire |
| <i>dstport</i> | Entier hexa. | $n > 0$ | Numéro de port utilisé par la destination du flux |
| <i>proto</i> | Entier décimal | 6=TCP 17=UDP | Couche transport utilisée pour le flux |
| <i>mode</i> | Entier décimal | 0=PASSIF 1=PERIODIQUE 2=PONCTUEL 3=POISSON_TIME 4=PAIR_POISSON_TIME 33=ACK_GENERATOR | Mode d'émission des données : passif (écoute), périodique, ponctuel (une émission), ou suivant une loi de Poisson |
| <i>Nb</i> | Entier décimal | $n > 0$ | Nombre de paquets à envoyer (PONCTUEL uniquement) |
| <i>T</i> | Entier décimal | $n > 0$ | Période d'envoi en ms (PERIODIQUE uniquement) |
| <i>Bits</i> | Entier décimal | $n > 0$ | Taille en bits du message à envoyer |
| <i>Prio</i> | Entier décimal | $n \geq 0$ | Priorité du flux. Non utilisé dans la simulation. |

4.2.1.3 MODIFICATIONS APORTEES AU GENERATEUR DE TRAFIC

Afin d'adapter le générateur de trafic à l'environnement de simulation basé sur le *mobility framework*, il a été nécessaire d'y apporter certaines modifications. Ces dernières sont décrites dans les paragraphes suivants.

4.2.1.3.1 Mode d'adressage

Comme précisé dans le Tableau 4.2, l'adresse MAC des nœuds est utilisée pour l'adressage dans le réseau. Le *mobility framework* faisant apparaître à la fois l'adresse MAC et l'adresse réseau (IP) des nœuds, il a été décidé :

- **De rendre égales adresse MAC et adresse IP des nœuds** (voir le paragraphe 4.2.2.3 pour les modifications engendrées au *mobility framework*) ;
- **D'utiliser comme référence des adresses MAC le paramètre *address* des modules de type *Host*.**

Le fonctionnement du mode d'adressage est décrit plus en détail dans l'Annexe A .

4.2.1.3.2 Taille des paquets

Dans les fichiers de description du trafic, la taille des paquets à envoyer est exprimée en octets. De même, dans tout le générateur de trafic, la taille des paquets est exprimée en octets (même si la documentation OMNeT++ spécifie que les fonctions permettant de fixer et d'obtenir la taille des paquets doivent utiliser des valeurs exprimées en bits).

Aussi, pour une meilleure cohérence avec l'API OMNeT++ et le reste de l'environnement de simulation, la taille des paquets est convertie en bits directement à la lecture du fichier de trafic (qui lui reste inchangé). Les modifications conséquentes ont été effectuées dans le générateur de trafic afin de maintenir la cohérence sur les tailles des messages.

Ainsi, **dans l'ensemble de l'environnement de la simulation toutes les tailles de paquets sont exprimées en bits.**

4.2.1.3.3 Mesure du temps

Le générateur de trafic ayant été conçu pour simuler une forme d'onde implémentant une autre forme de gestion d'accès au canal, il utilisait une mesure de temps particulière qui n'avait plus de sens dans le cadre d'une forme d'onde avec accès en contention.

Aussi, toutes les données temporelles ont été ramenées à une expression en secondes, avec l'introduction d'une variable de conversion dont la valeur est fixée de manière à ramener l'ensemble des durées dans l'unité légale de temps.

4.2.1.3.4 Convergence avec le mobility framework

Afin d'assurer l'intégration du générateur de trafic en tant que module applicatif de la structure d'un nœud tel que requise par le *mobility framework*, il a été nécessaire de développer un module de convergence, appelé *Mobility Framework Convergence Layer*, et qui réalise les fonctions suivantes :

- Encapsulation des messages de type *IpHeader* reçus depuis le module *IpModule* en messages de type *AppIPkt* définis dans le *mobility framework* ;
- Conversion des messages *AppIPkt* reçus depuis le module **NetwLayer* en *IpHeader* et transmission au module de type *DropDelay* ;
- Conversion des adresses de type FEC en adresses MAC ;
- Envoi d'un message d'erreur à la réception d'un message sur la porte *from_mngt* puisque la simulation ne gère pas les priorités des flux.

4.2.2 GESTION DE LA MOBILITE

La gestion de la mobilité s'appuie sur les principes introduits par le *mobility framework*. Dans ce dernier, la mobilité est gérée de manière distribuée : chaque nœud peut avoir son propre modèle de mobilité et bouge indépendamment des autres nœuds, sans connaître la position de ses homologues.

En contrepartie, après chaque mouvement, il met à jour ses coordonnées auprès du module *ChannelControl* qui lui connaît la position de tous les nœuds et, à l'aide du modèle physique utilisé dans la simulation, met à jour les connexions entre les nœuds (qui correspondent au fait que les nœuds soient en position d'interférence).

4.2.2.1 PRINCIPE DE FONCTIONNEMENT

La position « omnisciente » du module *ChannelControl* le rend propice au rôle de commande pour la mobilité des autres nœuds. Aussi, un module *TopologyChannelControl*, qui hérite du module *ChannelControl* a été développé.

Au début de la simulation, ce module ouvre et lit un fichier décrivant la topologie initiale et la mobilité des nœuds durant la suite du scénario. L'emplacement de ce fichier est un paramètre de l'environnement. A la lecture du fichier, il effectue les opérations suivantes :

- Calcul de la taille du *playground size* (taille de la zone dans laquelle les nœuds vont évoluer) qui doit être précisée au *mobility framework* avant le début de la simulation : la taille est fixée à la position maximale que prendra le nœud le plus distant de l'origine pendant la simulation ;
- Création des nœuds du réseau définis dans la partie « initialisation » du fichier de topologie et assignation des paramètres correspondants (adresse MAC, position,...) ;

- Création et envoi retardé de messages de type *ConstrainedMove* à chaque lecture d'une nouvelle position pour un nœud. Ces messages sont envoyés suffisamment tôt pour laisser au nœud le temps de traiter le message avant l'instant où la nouvelle position doit être atteinte (en fait, chaque message est envoyé au moment où est atteinte la position précédente).

Pour interagir avec le module de type *TopologyChannelControl*, un module *ConstrainedMobility*, héritant du module *BasicMobility*, a été créé. Ce module vient quelque peu rompre les paradigmes du *mobility framework* puisque le module *ConstrainedMobility* gère non-seulement les messages auto-adressés, mais aussi les messages du type *ConstrainedMove* en provenance du module *TopologyChannelControl*.

A la réception d'un message de type *ConstrainedMove* (qui indique l'ajout d'une nouvelle position à atteindre), le module de mobilité copie les informations contenues dans ce message (position et temps d'arrivée à cette position notamment) dans une structure de type *move*, insère cette structure dans une liste et prévoit l'envoi d'un message auto-adressé à l'instant où la position doit être atteinte.

A la réception d'un message auto-adressé (signifiant que le nœud doit bouger), le premier élément de la liste des mouvements est lu, le nœud est alors placé directement à cette position (par un « saut » instantané) et le mouvement lu est retiré de la liste.

Notons que le stockage des prochains mouvements dans une liste n'est pas nécessaire avec le calendrier retenu pour l'envoi des messages *ConstrainedMove*, mais a été introduit pour prévoir une compatibilité avec d'autres politiques éventuelles d'envoi de ces messages.

De plus, un autre module de mobilité, appelé *InterpolatedConstrainedMobility*, a été développé. Demandant un paramètre supplémentaire dans la configuration de la simulation, qui représente l'intervalle de mise à jour de la position, il propose d'interpoler le mouvement du nœud entre deux positions à deux instants distincts sur le modèle d'un déplacement à vitesse constante.

4.2.2.2 FICHIER DE DESCRIPTION DE LA MOBILITE

Comme pour le générateur de trafic, le module de gestion de la mobilité des nœuds lit des fichiers de description formatés, utilisés dans d'autres simulations réalisées chez Thales. Un exemple d'un tel fichier est donné en Figure 4.7 et le détail des différentes informations qu'il contient est donné dans le Tableau 4.3.

```
# SCALE == 100.0000
# n° slot @MAC  E      X      Y      Autres paramètres ...
# Partie initialisation
0          1      0x1  3333  3333  1=0 5=1 4=1 3=50 7=3 6=3
0          10     0x1  3333  4833  1=0 5=1 4=1 3=50 7=3 6=3
...
# Partie mobilité
20930      1      0x1  3333  3333
20930      10     0x1  8333  3333
```

Figure 4.7 : Exemple de fichier de mobilité

Tableau 4.3 : Interprétation du fichier de mobilité

| Variable | Type | Valeurs | Utilisation |
|----------|----------------|------------------------------|---|
| SCALE | Réel décimal | $x \geq 0$ | Echelle de la simulation. Les distances (et toutes les grandeurs dépendantes) sont divisées par cette échelle pour une meilleure représentation du réseau sous <i>tkenv</i> |
| N°slot | Entier décimal | $n \geq 0$ | Instant en nombre de slots écoulés où la position doit être atteinte (ou 0 pour les positions initiales). Cette valeur est convertie en secondes par le module <i>TopologyChannelControl</i> |
| @MAC | Entier héra. | $0 < n < \text{nb. noeuds}$ | Adresse MAC du nœud concerné par la commande (voir l'adressage du réseau 4.2.1.3.1) |
| E | Entier héra. | | Etat du nœud (ignoré dans la simulation) |
| X | Entier décimal | $n > 0$ | Position (en abscisse) à atteindre (voir Figure 4.8) |
| Y | Entier décimal | $n > 0$ | Position (en ordonnée) à atteindre (voir Figure 4.8) |
| Autres | Entier décimal | $n_{id} = n_{\text{valeur}}$ | Paramètres supplémentaires pour le nœud. Ignorés dans la simulation. |

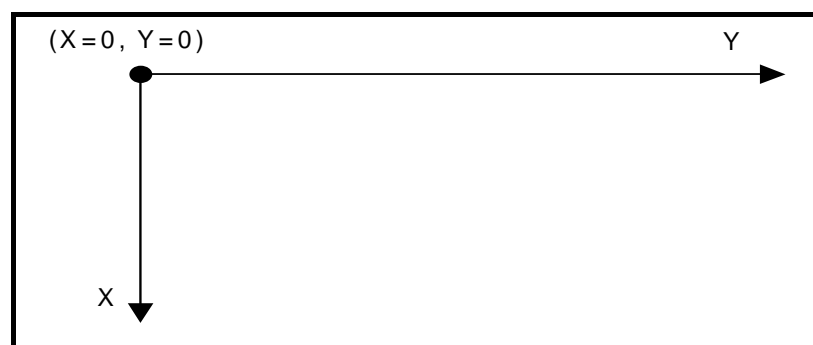


Figure 4.8 : Représentation des coordonnées

4.2.2.3 MODIFICATIONS APPORTEES AU MOBILITY FRAMEWORK

Les paragraphes suivants décrivent les modifications apportées au *mobility framework*, en plus de la création de nouveaux modules spécifiquement adaptés à l'environnement de simulation.

4.2.2.3.1 Mode d'adressage

Comme précisé dans le Tableau 4.2, l'adresse MAC des nœuds est utilisée pour l'adressage dans le réseau. De plus, adresse IP et adresse MAC ont été rendues égales pour plus de commodité.

Par défaut, le *mobility framework* utilise l'*id* du module **NetwLayer* comme adresse réseau et l'*id* du module *Nic* comme adresse mac. Aussi, pour garder une compatibilité avec le générateur de trafic, les modifications suivantes ont été apportées dans la classe *BasicNetwLayer* du *mobility framework* :

- La fonction *myNetwAddr()* renvoie maintenant la valeur du paramètre *address* du nœud (cette valeur est définie par la valeur du champ MAC lu dans la partie initialisation du fichier de topologie) ;
- La fonction *getMacAddr(netwAddr)* qui permettait de réaliser une translation adresse réseau vers adresse MAC (fonction d'ARP), renvoie simplement l'adresse réseau *netAddr* passée en paramètre (puisque adresse réseau et adresse MAC sont égales afin de maintenir la compatibilité avec le générateur de trafic).

Notons que la gestion de la mobilité fait appel aux adresses MAC au sens du *mobility framework*. Les modifications décrites précédemment n'ont pas modifié le fonctionnement de la mobilité des nœuds.

Le fonctionnement du mode d'adressage est décrit plus en détail dans l'Annexe A .

4.2.2.3.2 Taille des paquets

Afin de ne pas fausser les résultats de la simulation, il convient de conserver la valeur exacte de l'overhead engendré par les encapsulations successives liées à la traversée des diverses couches protocolaires (TCP/UDP, IP, MAC, PLCP) dont la taille des en-têtes est donnée dans le tableau Tableau 4.1.

Aussi, les tailles des en-têtes des modules *MFCLayer* (*appl.headerLength*) et **NetwLayer* (*net.headerLength*) ont été choisies et **doivent rester égales à zéro**. Les tailles respectives des en-têtes sont résumées dans le Tableau 4.1.

4.2.3 ROUTAGE DES DONNEES

L'établissement et la mise à jour des tables de routage des différents nœuds de la simulation sont assurés par le protocole de routage OLSR dont le fonctionnement est décrit en partie 3.1.2.2 de ce document.

Dans le contexte de la simulation, le fonctionnement du protocole est assuré par un module de routage *OlsrRouting* qui met à jour la table de routage *RoutingTable* en utilisant le protocole OLSR. L'interface avec le mobility framework est réalisé à l'aide du module *RoutingNetwLayer*.

4.2.3.1 PRINCIPE DE FONCTIONNEMENT

Le fonctionnement d'OLSR est assuré par le module *OlsrRouting* présent dans chacun des nœuds de la simulation. Ce module lance pour chaque nœud un « démon » OLSR qui génère les messages associés et calcule les différents ensembles nécessaires au fonctionnement d'OLSR (MPR set, MPR Selectors set, ...) qui permettent de mettre à jour la table de routage contenue dans le module *RoutingTable*.

Les trois modules cités précédemment interagissent de la manière suivante :

- Le module *RoutingNetwLayer* permet d'envoyer les messages de contrôle OLSR vers le module de routage *OlsrRouting* pour que ces derniers soient traités. Il permet aussi de transmettre les messages de contrôle OLSR générés par le module de routage aux

couches inférieures du nœud après encapsulation. Ce module assure la compatibilité du module de routage avec le formalisme du mobility framework ;

- Le module *OlsrRouting* traite les messages de contrôle OLSR arrivant et génère des messages sortants en utilisant les données des différentes tables relatives au fonctionnement du protocole. L'ensemble des calculs nécessaires à ce processus est assuré par le démon OLSR, le module ne servant qu'à faire l'interface entre le code OMNeT++ et le code C du démon. Il sert également à mettre à jour la table de routage contenue dans le module *RoutingTable* en fonctions des données calculées par le démon OLSR ;
- Le module *RoutingTable* est issu du INET framework (autre framework développé pour OMNeT++) et simule le fonctionnement d'une table de routage. Il est utilisé par le module *OlsrRouting* qui lui fournit les informations de topologie et de routage calculées par le protocole OLSR, et par le module *RoutingNetwLayer* qui utilise les données contenues dans la table de routage pour savoir vers quel nœud diriger les paquets sortants.

L'utilisation de ces modules, parce qu'ils proviennent d'autres environnements de simulation et utilisent d'autres *frameworks*, a nécessité diverses adaptations décrites dans les parties suivantes.

4.2.3.2 ADAPTATION A L'ENVIRONNEMENT DE SIMULATION

Comme décrit précédemment, la table de routage est issue du INET framework et son utilisation dans l'environnement de simulation a donc nécessité plusieurs adaptations, notamment en ce qui concerne le mode d'adressage des nœuds et le processus de routage.

4.2.3.2.1 Adressage des nœuds

Pour chaque nœud du réseau, le *INET framework* nécessite que soit définie une variable *routerId* qui sert à identifier de manière unique le nœud sur le réseau. Cette variable, qui est de la classe *IPAddress*, correspond en fait à l'adresse IP (ou adresse réseau) du nœud dans la simulation. Cette adresse est lue depuis le paramètre *routerId* du module *RoutingTable* lors de l'initialisation de celui-ci (le format doit être « A.B.C.D », où A, B, C et D sont des entiers entre 1 et 254).

Pour plus commodité et afin de maintenir une cohérence dans le mode d'adressage des nœuds entre le générateur de trafic et la table de routage, la translation suivante est mise en œuvre :

- Un paramètre *addressPrefix* est créé pour la simulation, il est utilisé pour convertir les adresses des nœuds en adresse IP (ex : « 192.168.0 ») ;
- Lors de son initialisation (par le module *TopologyChannelControl*) chaque nœud se voit attribuer un paramètre *ipAddress*, égal à son adresse (au sens du générateur de trafic) préfixée par le préfixe *addressPrefix* (ex : le nœud 1, prend l'adresse IP « 192.168.0.1 ») ;
- La variable *routerId* du module *RoutingTable* est définie comme étant égal au paramètre *ipAddress* du nœud parent.

La translation d'adresse est implémentée dans le seul module assurant l'interface entre les adresses au sens du générateur de trafic et les adresses au sens du *INET framework*. Aussi, cette fonction est assurée uniquement dans le module *RoutingNetwLayer* au moyen des fonctions *addrToIpAddr()* et *ipAddrToAddr()*.

Il existe maintenant trois adressages coexistants dans la simulation :

- L'adressage du générateur de trafic (où adresse MAC et adresse réseau égalent un paramètre *address* du nœud) ;
- L'adressage du mobility framework (où l'adresse MAC est l'id du module NIC et l'adresse réseau l'id du module net) utilisée pour la mobilité des nœuds ;
- L'adressage du INET framework (où l'adresse IP d'un nœud est obtenue en préfixant l'adresse du nœud au sens du générateur de trafic).

Les zones de validité ainsi que les interfaces entre ces différents modes d'adressage sont décrites dans la Figure A.1.

4.2.3.2.2 Utilisation de la table de routage

Dans le *INET framework*, chaque nœud est supposé contenir un module jouant le rôle de routeur, c'est à dire possédant plusieurs interfaces. Aussi, chaque nœud doit, à l'initialisation de la simulation enregistrer ses différentes interfaces (et obtenir une adresse MAC unique pour chacune d'entre elles). De même, la *RoutingTable* fournit non-seulement, pour chaque paquet qu'un nœud doit retransmettre, l'adresse réseau du prochain nœud mais aussi l'interface sur laquelle le paquet doit être envoyé.

Cette conception, assez peu pertinente dans le contexte de notre simulation où chaque nœud ne possède qu'une interface unique, risque également de poser un problème de compatibilité avec le mode d'adressage du générateur de trafic. Aussi, toutes les références aux interfaces telles que les définit le *INET Framework* ainsi que les fonctions de translation MAC/IP qu'il propose ont été supprimées.

Par conséquent, l'utilisation de la *RoutingTable* est légèrement modifiée (en particulier, vu du *INET framework* tous les paquets sont envoyés sur l'interface locale des nœuds) . Puisque seules les informations concernant l'adresse du prochain nœud conservent donc un sens, le routage d'un paquet s'effectuera de la manière suivante :

- La fonction *nextGatewayAddress(destAddress)* qui renvoie l'*IPAddress* du prochain nœud sur le plus court chemin vers *destAddress* devra être utilisée pour déterminer l'adresse réseau du voisin à qui transmettre le paquet ;
- La fonction *sendDown()* du mobility framework nécessitant que l'adresse au sens du générateur de trafic soit fournie, la valeur de retour de la fonction précédente devra être convertie avec la fonction *ipAddrToAddr()* et utilisée pour l'envoi.
- Si l'adresse de destination n'était pas dans la table de routage, l'adresse IP « nulle » (*IPAddress()*) a été retournée et convertie en la valeur *NULL*. Il est donc nécessaire de tester la valeur de retour de cette conversion et de supprimer le paquet impossible à router le cas échéant.

Notons enfin qu'à l'adresse de broadcast « 255.255.255.255 » du *INET Framework* correspond l'adresse de broadcast « -1 » commune au *mobility framework* et au générateur de trafic.

Le broadcast des paquets est considéré comme étant local (un paquet broadcast n'étant jamais retransmis par un nœud).

4.2.3.2.3 Adaptation au mobility framework

L'adaptation au *mobility framework* est réalisée par le module *RoutingNetwLayer* qui permet, entre autre de séparer correctement les paquets entre les différents modules :

- Les paquets arrivant de la couche supérieure (*appl*) sont encapsulés en *NetwPkt* et routés à l'aide des informations de la *RoutingTable* puis transmis aux couches inférieures;
- Les paquets arrivant des couches inférieures sont :
 - S'il s'agit de paquets de données, soit remontés aux couches supérieures s'ils sont arrivés à destination, soit routés vers le prochain nœud sur leur chemin ;
 - Soit transmis au module *OlsrRouting* s'il s'agit de paquets de contrôle du protocole de routage ;
- Les paquets provenant de la couche de routage sont encapsulés en *NetwPkt* puis transmis en broadcast à tous les voisins du nœud.

Notons que, puisque le module *RoutingNetwLayer* est situé sous les couches TCP/UDP et IP, la taille des paquets de contrôle qui y parviennent doit tenir compte la taille des en-têtes de transport (UDP dans le cas d'OLSR) et réseau (en-tête IP) dont la taille est précisée dans le Tableau 4.1.

4.2.4 GESTION DE L'ACCES AU CANAL

La couche de contrôle de l'accès au canal est basée sur CSMA/CA avec l'option RTS/CTS qui fait partie du standard 802.11 de l'IEEE. Le fonctionnement, les principaux atouts et inconvénients de cette méthode de contrôle d'accès sont détaillés dans les parties 3.2.1.2.1 et 3.2.1.2.2 de ce document.

4.2.4.1 PRINCIPE DE FONCTIONNEMENT

L'implémentation utilisée est celle fournie par le *mobility framework*, par le biais du module *Mac80211*.

S'il permet de simuler efficacement le mécanisme de CSMA/CA, ce module souffre de certaines limitations. Notamment, la gestion de la fragmentation (qui fait pourtant partie intégrante du standard et dont le fonctionnement est décrit en 3.2.1.2.2.2) n'est pas implémentée.

D'autre part, la gestion de la file d'attente des paquets en provenance des couches supérieures est des plus simples puisqu'il s'agit d'une unique file FIFO (*first-in-first-out*) dans laquelle sont stockées les données à destination de tous les nœuds et les messages de contrôle sans différenciation.

La modification de ce module afin d'introduire la fragmentation des paquets et d'améliorer la gestion des messages entrants peut conduire à d'importantes évolutions favorables des performances et est envisagée comme une évolution du modèle de simulation.

4.2.4.2 ADAPTATION AUX CONTRAINTES DES COMMUNICATIONS TACTIQUES

La spécification de la couche MAC de Wi-Fi décrite dans [22], correspond à la portée et au débit requis pour une utilisation civile. En particulier les durées des différents *interframe spacings* (durées inter-frames), également présentés dans le Tableau 3.3, correspondent à des portées de transmissions n'excédant pas quelques centaines de mètres.

Dans le cadre de communications tactiques, les stations doivent être capable de communiquer directement (un seul bond radio) jusqu'à une distance de 15 à 20 km (voire 30 km si elles sont placées sur des « points hauts » ce qui permet d'augmenter leur couverture). On comprend bien que ce paramètre, impactant le délai de propagation va nécessiter une modification de ces délais.

Cet effet n'a jamais été étudié dans la littérature car la plupart des publications universitaires utilisent le plus souvent, pour des raisons pratiques et matérielles, du matériel Wi-Fi standard pour mener leurs expériences.

4.2.4.2.1 Détail de calcul des durées inter-frames

Les durées inter-frames sont le plus souvent énoncées sans justification et seule une lecture du standard 802.11 de l'IEEE[22], permet d'obtenir le détail des calculs.

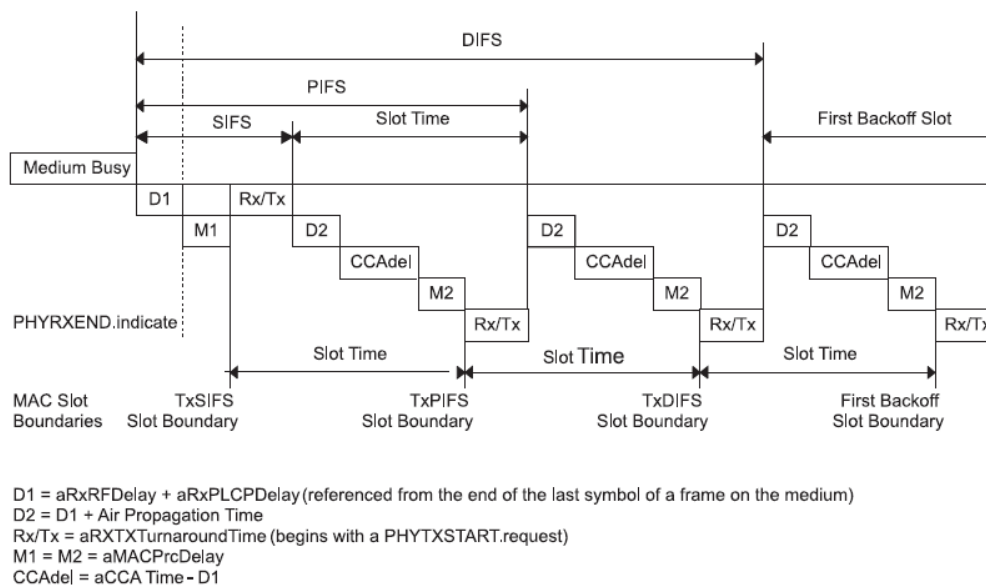


Figure 4.9 : Détails du calcul des durées inter-frames

En résumé, les durées nous intéressant particulièrement (t_{slot} et *SIFS*) sont définies de la manière suivante :

$$t_{slot} = aCCATime + aRxTxTurnaroundTime + aAirpropagationTime + aMACprocessDelay \quad (\text{eq. 4.1})$$

$$SIFS = aRxRFDelay + aRxPLCPDelay + aAirpropagationTime + aMACprocessDelay \quad (\text{eq. 4.2})$$

Le même document fournit une description des différents paramètres, ce qui va nous permettre de les adapter aux contraintes de notre forme d'onde tactique. Les paramètres pertinents sont regroupés dans le Tableau 4.4.

Tableau 4.4 : Paramètres de calcul des durées inter-trames

| Paramètre | Description |
|----------------------------|---|
| <i>aCCATime</i> | Temps minimum (en μs) nécessaire au mécanisme de détection de porteuse pour déterminer si le médium est libre ou occupé |
| <i>aRxTxTurnaroundTime</i> | Temps minimum (en μs) nécessaire à la chaîne de réception physique pour passer du mode réception à l'émission du premier symbole |
| <i>aRxPLCPDelay</i> | Temps nominal (en μs) nécessaire à la couche PLCP pour transmettre un bit de la couche PMD (dépendant de la couche PHY) à la couche MAC (indépendante de la couche PHY) |
| <i>aRxRFDelay</i> | Temps nominal (en μs) entre la fin de la réception d'un symbole à l'interface radio et l'envoi d'une notification de réception à la couche PLCP |
| <i>aAirPropagationTime</i> | Temps nominal (en μs) nécessaire à un signal transmis pour aller de la station émettrice à la station réceptrice |
| <i>aMACProcessingDelay</i> | Temps nominal (en μs) utilise par la couche MAC pour analyser une trame reçue et préparer une réponse à cette trame. |

4.2.4.2.2 Calcul des durées inter-trames

Ces données vont permettre de calculer les paramètres t_{slot} , *SIFS* et *DIFS* relatifs à notre forme d'onde. Les données retenues sont présentées dans le tableau

Tableau 4.5 : Durées inter-trames pour les standards 802.11 et la forme d'onde tactique

| | 802.11b | 802.11g | Tactique Hdw | Tactique Sftw |
|---------------------------|-----------|-----------|---------------------|---------------------|
| tslot | 2,000E-05 | 9,000E-06 | 1,552E-04 | 1,153E-03 |
| SIFS | 1,000E-05 | 1,600E-05 | 4,700E-05 | 1,045E-03 |
| DIFS | 5,000E-05 | 3,400E-05 | 3,575E-04 | 3,351E-03 |
| TX_range | 250 | 250 | 15000 | 15000 |
| PCS_range | 550 | 550 | 30000 | 30000 |
| AirPropagTime | 1,835E-06 | 1,835E-06 | 1,001E-04 | 1,001E-04 |
| RxRFDelay | 2,500E-06 | 2,500E-06 | 2,500E-06 | 2,500E-06 |
| RxPLCPDelay | 2,500E-06 | 2,500E-06 | 2,500E-06 | 2,500E-06 |
| MACProcessingDelay | 0,000E+00 | 2,000E-06 | 2,000E-06 | 1,000E-03 |
| RxTxTurnaroundTime | 5,000E-06 | 5,000E-06 | <i>Confidentiel</i> | <i>Confidentiel</i> |
| CCATime | 1,317E-05 | 4,000E-06 | <i>Confidentiel</i> | <i>Confidentiel</i> |

Les deux variantes des données de la de la forme d'onde tactique considèrent soit que le traitement MAC est fait en matériel (*chipset* ou FPGA dédié) ce qui l'accélère grandement, soit (et c'est alors plus conforme au concept de software radio) par une couche logicielle dont le temps de traitement est beaucoup plus long.

L'adaptation des durées inter-trames se montrant assez pénalisant lors de l'utilisation de l'accès en contention dans des contraintes tactiques, par la suite, **l'hypothèse d'une implémentation sur du matériel dédié de la couche MAC a été retenue pour les simulations.**

4.2.5 MODELISATION DU CANAL RADIO

Le modèle radio utilisé pour la simulation doit permettre de mettre en évidence les limitations de la méthode de contrôle d'accès au canal implémentée dans 802.11 tout en étant suffisamment simple pour ne pas créer « d'artefacts » supplémentaires qui pourraient rendre les résultats obtenus difficilement interprétables lors d'une première approche.

Dans l'environnement de simulation, la description du canal radio intervient dans le module *ChannelControl* et les modules des couches basses de chacun des nœuds (*Mac*, *Decider* et *SnrEval*).

4.2.5.1 DESCRIPTION DU MODELE RETENU

Le modèle radio retenu est donc un compromis entre un modèle fin (prenant en compte l'atténuation, l'augmentation du taux de bits d'erreur, les interférences entre nœuds, ...) et un modèle simpliste, beaucoup trop favorable à CSMA/CA, dans lequel chaque nœud n'a qu'une portée unique et est insensible à tout ce qui se passe au-delà.

De plus, la modélisation retenue doit prendre en compte les impératifs physiques (débit, portée, bande de fréquence) relatifs à une utilisation dans le domaine tactique et qui sont différents des paramètres standard du Wi-Fi « domestique ».

4.2.5.1.1 Modèle de propagation

Le modèle de propagation retenu est un modèle « tout ou rien » (parfois également qualifié de modèle « debug ») puisqu'une distance donnée est utilisée comme référence pour dire si oui ou non les paquets reçus ont une influence sur le récepteur.

Mais, afin d'être sensibles aux limitations de CSMA, deux distances caractéristiques ont été retenues :

- Une distance de transmission (*TX_range*), en deçà de laquelle les paquets émis sont reçus sans perte, sans erreur et sans affaiblissement de puissance. Plusieurs paquets émis en deçà de cette zone peuvent créer une collision ;
- Une distance d'interférence (*IF_range*), telle que les paquets émis en deçà de cette distance ne peuvent pas créer de collision mais occupent tout de même le canal (il est donc impossible pour un nœud d'émettre si une émission a déjà lieu en deçà de cette distance). Cette distance correspond donc à une distance de sensibilité à la porteuse (*carrier sensing*), mais est nommée distance d'interférence car elle utilise la notion de distance d'interférence du mobility framework (voir paragraphe 4.1.2). En fait, dans le modèle retenu distances d'interférence et de *carrier sensing* ont été choisies égales.

La forme d'onde utilisée ayant des applications tactiques, elle est notamment renforcée contre le brouillage et la collision des paquets. Aussi, on estime qu'un paquet dont moins de la moitié des

données est brouillée par une collision est récupérable (sauf si la collision a eu lieu sur les bits de synchronisation, auquel cas le paquet est perdu).

Les hypothèses faites sur ces distances conduisent à dire que le modèle radio retenu est un modèle « tout ou rien », sans atténuation, prenant en compte les collisions de paquets et la sensibilité à la porteuse mais pas les interférences.

Le modèle retenu est illustré par la Figure 4.10 ci-dessous.

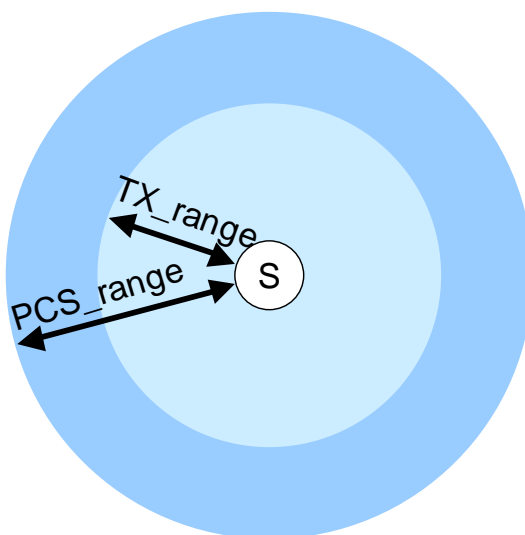


Figure 4.10 : Modèle radio de la simulation

4.2.5.1.2 Paramètres physiques

Les paramètres physiques utilisés sont hérités des domaines de fonctionnement des communications tactiques et à laquelle la solution Wi-Fi/OLSR se conformer. L'ensemble de ces paramètres est détaillé dans le Tableau 4.6 ci-dessous.

Tableau 4.6 : Paramètres physiques de la simulation

| Paramètre | Valeur |
|-------------------------------------|----------------------------|
| Fréquence porteuse* | 2,4 Ghz |
| Zone d'interférence max. | 30 km |
| Zone de transmission max. | 15 km |
| Débit | 547 kbits/s |
| Bits de synchronisation | 80 bits |
| Seuil de récupération sur collision | 50% de la taille du paquet |

* Paramètre non-modifié mais sans incidence sur le déroulement de la simulation

4.2.5.2 IMPLEMENTATION DU MODELE

La section suivante décrit le mode d'implémentation du modèle, notamment les différents modules qui ont été développés et les modifications apportées au *mobility framework* afin d'y intégrer la gestion du canal radio propre à la simulation.

4.2.5.2.1 Modules de gestion du modèle radio

Dans le *mobility framework*, la modélisation du canal radio pour 802.11 est implémentée dans les modules *SnrEval80211* (qui calcule la puissance de réception des paquets, gère le bruit, la collision et les interférences) et *Decider80211* (qui utilise un modèle probabiliste fonction du rapport signal à bruit des paquets afin de générer un taux de bits d'erreur).

Pour les besoins de la simulations, deux modules, *DebugEval80211* et *DebugDecider80211*, héritant respectivement des modules précédemment décrits, ont été développés.

4.2.5.2.1.1 Module DebugEval80211

Conformément au modèle radio choisi, le module *DebugEval80211*, se base uniquement sur la distance à l'émetteur (et non plus de la puissance de réception du paquet) pour considérer le fait qu'un paquet est interprétable (et peut éventuellement créer des collisions) ou s'il n'est uniquement perçu que comme un signal occupant le canal.

A l'arrivée d'un message, un pointeur vers celui-ci le marque comme actuellement reçu. Lui est également associé une liste des événements survenus pendant la réception.

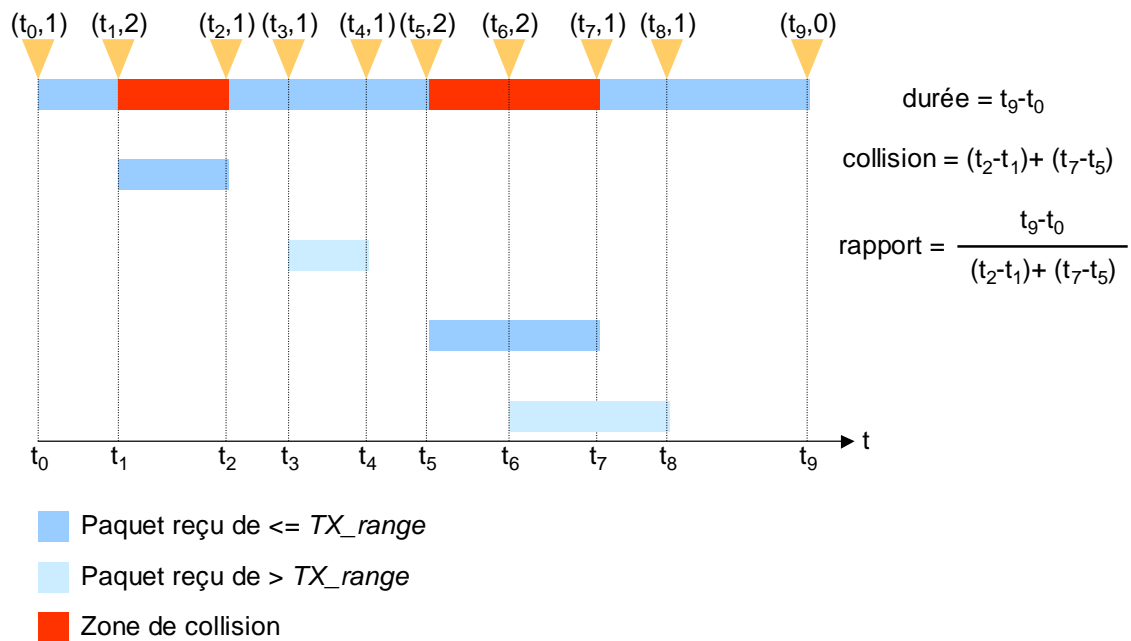
La structure de cette liste est la suivante :

- Un pointeur vers le message ;
- Une liste d'évènements de la forme : (instant de simulation, nombre de paquets en collision).

Lorsqu'un autre message arrive, un nouvel événement, avec comme instant de simulation la date d'arrivée du nouveau message, est ajouté de la manière suivante :

- Si le message a été envoyé depuis une distance inférieure à *TX_range*, le nombre de paquet en collision est incrémenté, et l'événement est ajouté ;
- Si le message a été envoyé depuis une distance supérieure à *TX_range*, le nombre de paquet potentiellement en collision n'est pas incrémenté, et l'événement est ajouté.

Ainsi, la liste des événements associée à un message permet de suivre l'évolution du nombre de messages en collision avec celui-ci et de calculer la durée de réception du paquet et la durée totale des périodes de collision.



$(t_0, 1)$ (instant, nb de paquets potentiellement en collision)

Figure 4.11 : Calcul du temps de collision d'un message

4.2.5.2.1.2 Module DebugDecider80211

Sur réception des messages envoyés par le module *DebugEval80211*, le module *DebugDecider80211* calcule, en utilisant la liste d'événements survenus pendant la réception transmise avec le message :

- La durée totale de la réception du message par le module *DebugEval80211* ;
- La durée totale des collisions éventuelles survenues pendant cette réception ;
- Les instants auxquels ces collisions sont survenues.

Si la durée de collision ramenée à la durée de réception du message dépasse un seuil de tolérance (défini par la variable *maxInterfRatio*) ou si une collision est survenue pendant les bits de synchronisation (dont le nombre est fixé par *syncBits*), le message est supprimé. Sinon, il est transmis à la couche supérieure.

4.2.5.2.2 Modifications du mobility framework

Dans sa version originale, le *mobility framework* calcule la distance d'interférence entre les nœuds en fonction de divers paramètres physiques définis dans le fichier de configuration.

Aussi, un module dérivant de *ChannelControl* (module qui met à jour les connections entre les nœuds du *mobility framework*) a été implémenté, notamment afin de redéfinir le calcul de la zone d'interférence des nœuds. Ce dernier est extrêmement simplifié, puisqu'il retourne simplement la valeur du paramètre *IF_range* du fichier de configuration.

Deux méthodes, permettant d'accéder à la valeur des paramètres *IF_range* et *TX_range* ont également été implémentées.

5. SIMULATIONS ET ANALYSE DES RESULTATS

La partie suivante du document décrit différents scénarios de simulations puis présente et analyse les résultats obtenus.

L'ensemble des scénarios utilisés permet soit de tester l'impact de certains paramètres (débits, topologie, mobilité des nœuds) sur le comportement du réseau, soit d'avoir un aperçu des performances de la forme d'onde dans des conditions opérationnelles.

5.1 REPONSE EN DEBIT SANS MOBILITE

Le premier scénario réalisé vise à tester la capacité du réseau à absorber un débit de plus en plus important afin d'en connaître la charge maximale. L'intérêt est également de comparer la valeur limite et l'évolution de cette réponse en fonction de l'étendue de la zone de sensibilité à la porteuse.

Pour l'instant, même si le protocole de routage est activé (des messages OLSR transiteront dans le réseau en plus des données « utiles »), les stations sont considérées comme fixes.

5.1.1 DESCRIPTION DU SCENARIO

Dans ce scénario, 16 stations sont disposées sur grille de 4x4 stations. Chaque nœud est distant de ses voisins de 4km de telle sorte que chaque station ne peut recevoir et émettre correctement vers les stations situées directement au-dessus, en dessous, à gauche et à droite. La topologie est décrite dans la Figure 5.1.

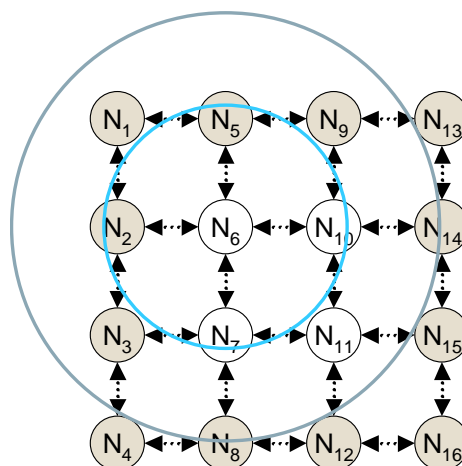


Figure 5.1 : Grille de 16 stations

Le trafic généré est un flux de poisson dont la paire source-destination est choisie de manière aléatoire parmi les nœuds périphériques. Des messages de 1000 octets sont émis avec un intervalle moyen croissant de 80 à 6 ms, de telle sorte que le débit moyen soumis au réseau varie entre 100 kbits/s et 1300 kbits/s.

De plus, deux campagnes de mesures sont menées, en faisant varier la taille de la zone de sensibilité à la porteuse (*IF_range*) qui peut être étendue (*IF_range* = 10 km) quand elle couvre les voisins, les voisins à deux bonds et les nœuds directement situés en diagonale, ou restreinte (*IF_range* = *TX_range*). Aucune sensibilité aux collisions n'est tolérée.

5.1.2 PRESENTATION ET ANALYSE DES RESULTATS

5.1.2.1 RESULTATS OBTENUS

Le tableau suivant récapitule la valeur de différentes métriques caractéristiques du comportement du réseau pour différentes valeurs de débit et différentes tailles de la zone de sensibilité à la porteuse.

Tableau 5.1 : Résultats de la simulation de réponse en débit sans mobilité

| Débit | 100 kbits/s | | 200 kbits/s | | 400 kbits/s | | 500 kbits/s | | 800 kbits/s | | 1000 kbits/s | | 1300 kbits/s | |
|------------------------|-------------|----------|-------------|----------|-------------|----------|-------------|----------|-------------|----------|--------------|----------|--------------|----------|
| IF_range | réduit | étendu | réduit | étendu | réduit | étendu | réduit | étendu | réduit | étendu | réduit | étendu | réduit | étendu |
| Débit soumis (kbits/s) | 101,5 | 97,44 | 190,6 | 206,5 | 394,8 | 394,9 | 483,2 | 487,9 | 824,6 | 778,6 | 1010,5 | 942,1 | 1325,1 | 1339,6 |
| Débit reçu (kbits/s) | 76,6 | 96,96 | 126,7 | 179,4 | 202,4 | 264,4 | 225,6 | 271,3 | 299,9 | 284,3 | 328,5 | 285,1 | 373,6 | 320,6 |
| Rapport | 0,754 | 0,995 | 0,664 | 0,87 | 0,51 | 0,67 | 0,47 | 0,56 | 0,36 | 0,36 | 0,35 | 0,30 | 0,28 | 0,24 |
| Délai* moyen (s) | 0,46 | 1,23 | 0,76 | 5,32 | 1,57 | 7,52 | 1,93 | 8,14 | 2,85 | 7,6 | 2,78 | 7,20 | 3,02 | 6,63 |
| Délai* maximal (s) | 8,86 | 21,65 | 14,2 | 147,5 | 18,6 | 506,6 | 22,14 | 501,4 | 23,8 | 487,9 | 22,4 | 577,0 | 24,4 | 696,2 |
| Écart type (s) | 0,54 | 2,03 | 0,87 | 9,5 | 1,81 | 13,5 | 2,23 | 13,8 | 2,86 | 10,7 | 2,71 | 8,28 | 2,78 | 9,15 |
| Collisions (paquets) | 10^5 | 1.10^4 | 1.10^5 | 2.10^4 | 2.10^5 | 3.10^4 | 2.10^5 | 4.10^4 | 3.10^5 | 4.10^4 | 3.10^5 | 5.10^4 | 3.10^5 | 5.10^4 |

* Le délai calculé concerne le délai point à point entre émetteur et destinataire final du paquet

Les résultats montrent que **l'utilisation d'un accès au canal en contention ne permet pas au réseau de répondre correctement à un débit important** (le rapport débit soumis/débit reçu diminue quand le débit augmente). L'architecture mise en place ne semble donc pas performante dans le cas d'un réseau subissant une forte charge. La réponse du réseau est synthétisée sur la Figure 5.2.

Au-delà de cette simple constatation, il est intéressant d'analyser l'influence d'une zone de sensibilité à la porteuse (*IF_range*) étendue sur le comportement du réseau. Remarquons :

- Que la zone étendue est bénéfique pour des débits moyens (< 800 Kbits/s), voire très bénéfique pour de très faibles débits (99,5% de données reçus pour un débit de 100 kbits/s) notamment car elle permet de réduire le nombre de collisions;
- En revanche, elle allonge le délai de réception des données (facteur 3 à 4 sur le délai moyen) ce qui rend son utilisation impossible en pratique avec des applications nécessitant de faibles temps de latence, comme la voix sur IP par exemple.

En conclusion, la zone de sensibilité étendue en donnant une meilleure « vue » sur l'occupation du canal, entraîne une émission moins fréquente, mais plus sûre des données.

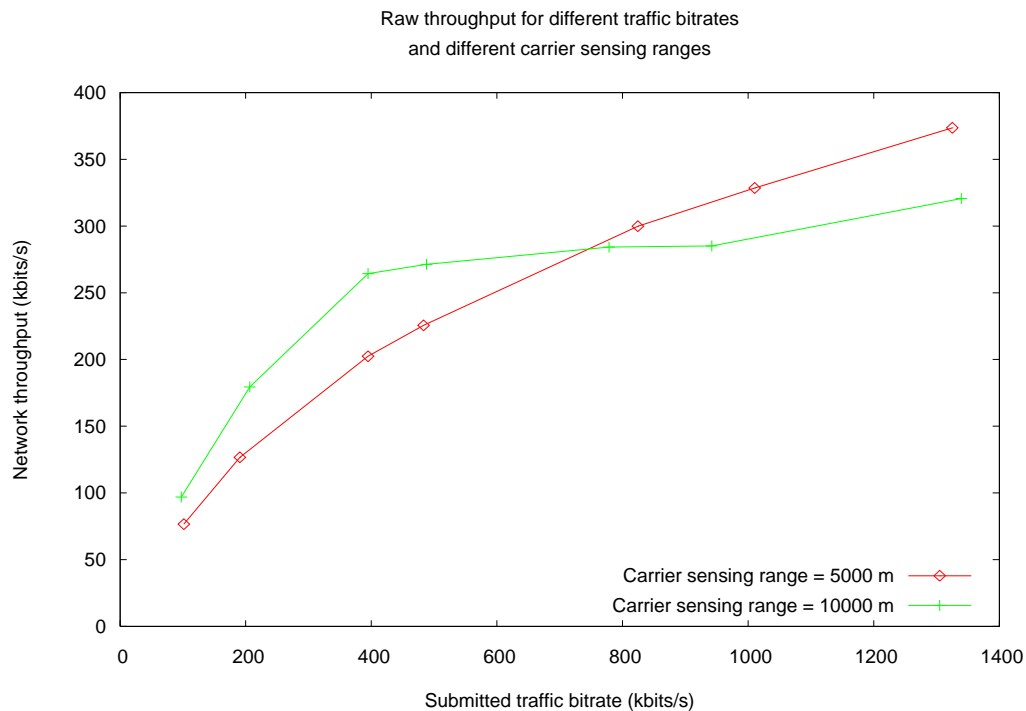


Figure 5.2 : Réponse en débit du réseau pour deux valeurs de la zone de sensibilité à la porteuse

5.1.2.2 ANALYSES ET JUSTIFICATIONS

L'observation du nombre de collisions et l'analyse des différentes causes de perte de paquets de données permet de mieux expliquer ces résultats. Leur évolution est donnée dans la Figure 5.3 et la Figure 5.4 ci-dessous.

Dans les simulations avec **zone de sensibilité à la porteuse réduite**, de nombreux paquets sont perdus car leur nombre maximal de tentatives de transmissions (7 dans la norme 802.11) a été atteint sans permettre de les envoyer avec succès. Ce qui signifie bien que **de nombreuses collisions surviennent dans le réseau et empêchent les communications de se dérouler correctement**.

D'autre part, la perte de paquets de données par absence de route vers une destination augmente avec le débit, ce qui signifie que le **protocole de routage ne parvient pas à fonctionner correctement au-delà d'une certaine charge**. Comme la taille de la file MAC est assez faible (moins de 15 paquets dans la file) quand surviennent ces erreurs, et que le nombre de collisions est assez important (10 fois plus qu'avec la sensibilité étendue), le **dysfonctionnement d'OLSR est attribué à de trop nombreuses collisions** survenant sur les trames envoyées par le protocole de routage (qui sont envoyées en broadcast, et donc sans mécanisme de signalisation RTS/CTS).

Dans le cas d'une **zone de sensibilité étendue**, la cause majeure de perte est liée à des files MAC pleines. L'hypothèse est émise que, comme **le canal est vu occupé beaucoup trop longtemps, les nœuds ne parviennent pas à émettre assez souvent pour vider leur file**. Au-delà de 500 kbits/s cet effet a aussi un impact important sur le fonctionnement d'OLSR pour les forts débits (nombre de paquets perdus par absence de route en forte hausse à partir de 500 kbits/s).

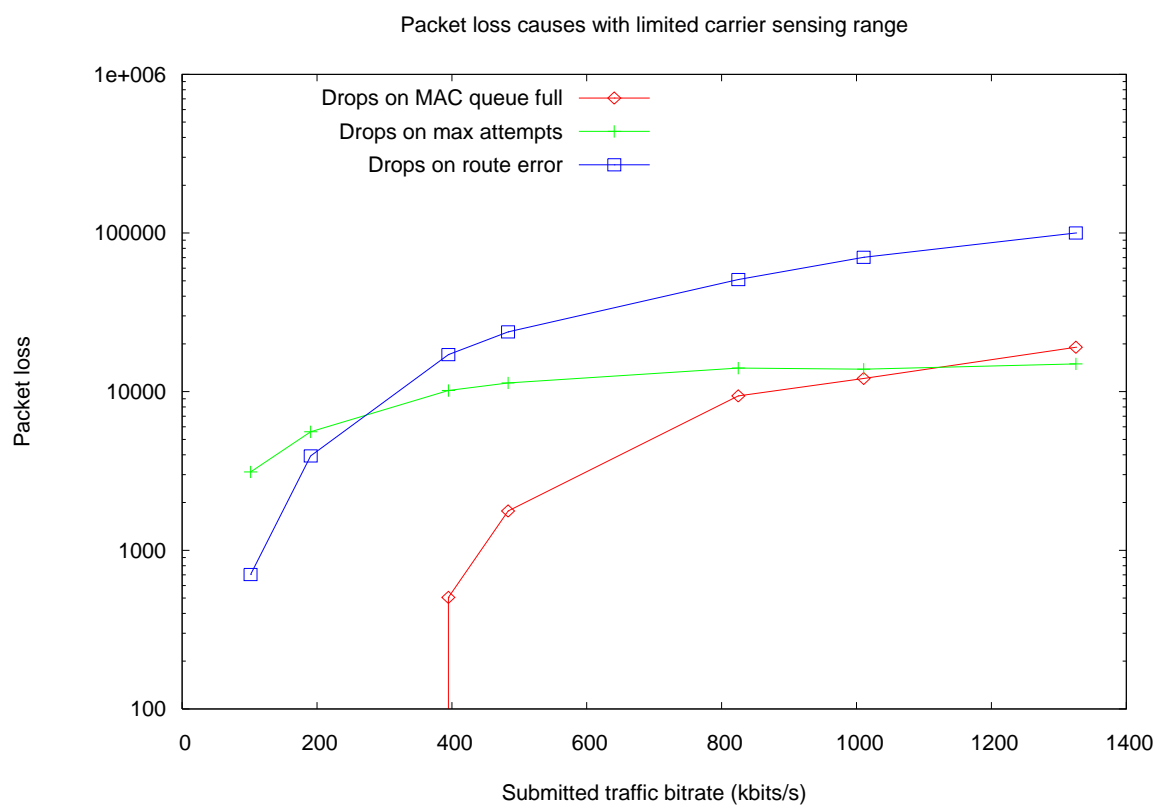


Figure 5.3 : Causes de perte de paquets (zone de sensibilité réduite)

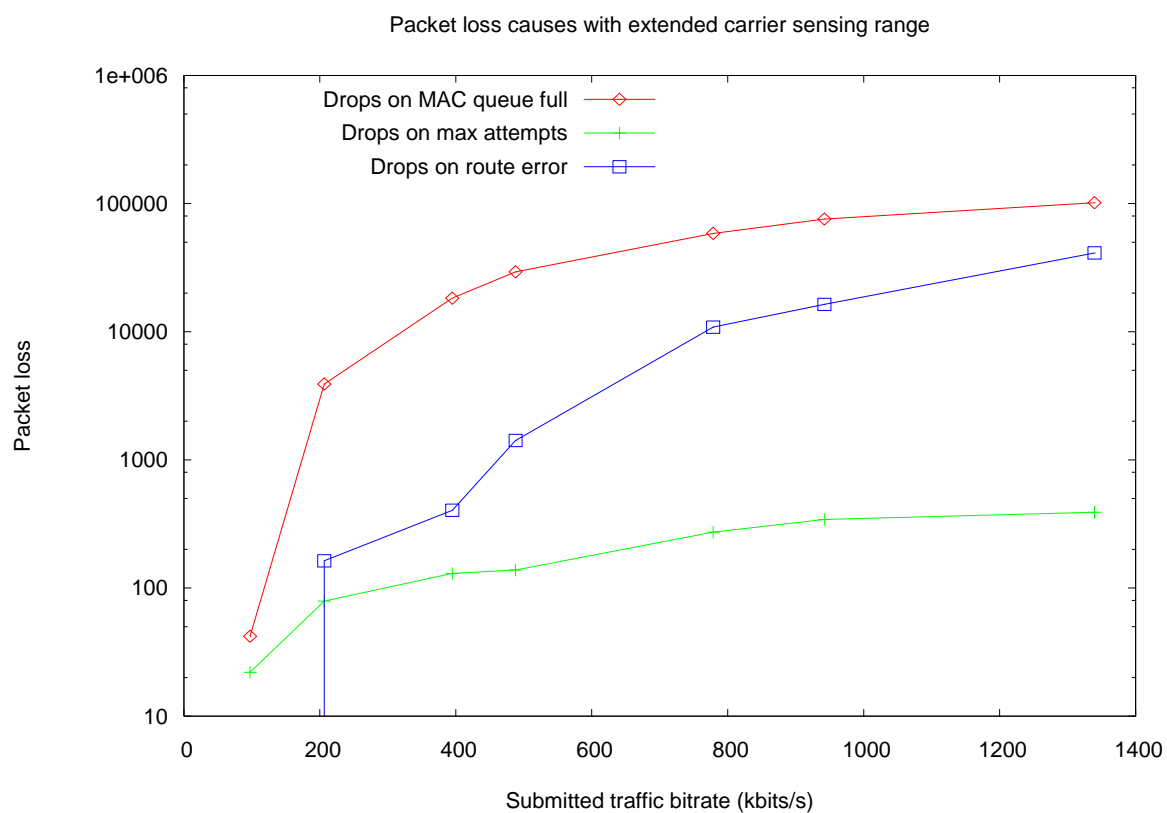


Figure 5.4 : Causes de perte de paquets (zone de sensibilité étendue)

5.1.2.2.1 Impact de la zone de sensibilité sur les collisions

L'analyse des types de collisions qui surviennent est intéressante et montre, outre que les collisions sont nettement plus nombreuses avec la zone de sensibilité réduite, que le mécanisme RTS/CTS reste fragile.

En effet, il subsiste de nombreuses collisions qui ne sont théoriquement pas sensées survenir (comme les collisions ACK sur données ou CTS sur données) en plus des collisions probables (comme les collisions frontales de RTS). En fait, l'analyse poussée du déroulement des simulations montre que des collisions sur les messages de signalisation causent un dysfonctionnement de la réservation virtuelle du canal effectué par MACA, ce qui cause en général des collisions sur les paquets de données et nécessite leur retransmission.

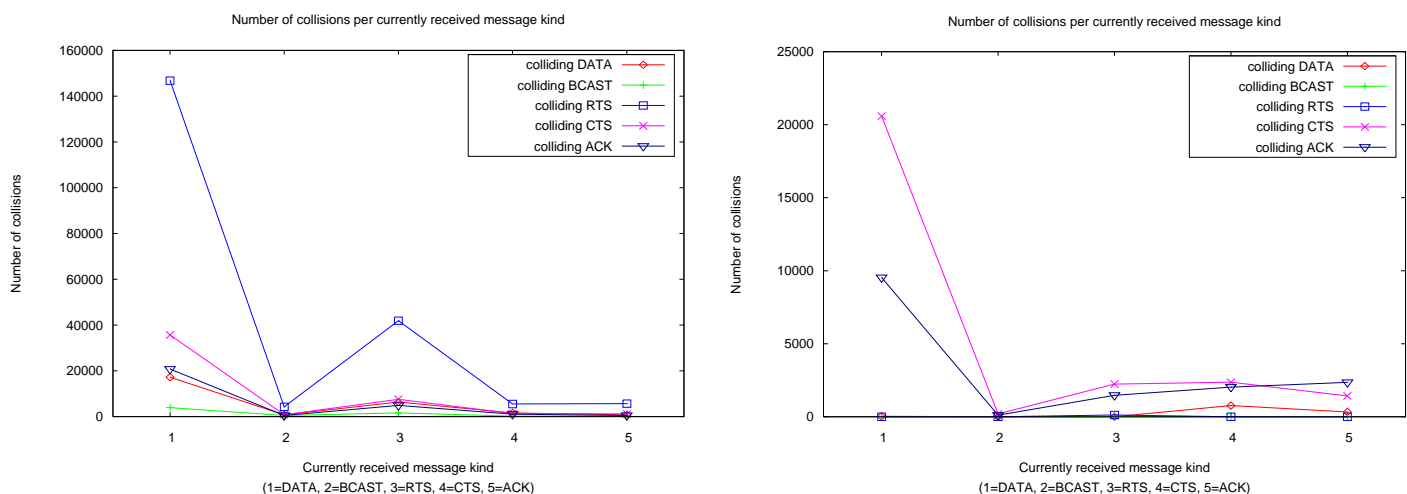


Figure 5.5 : Types collisions pour la zone de sensibilité réduite (à gauche) et étendue (à droite) pour un débit soumis de 800 kbits/s

5.1.2.2.2 Impact de la zone de sensibilité sur la latence

L'analyse du temps passé par les nœuds du réseau dans les différents états radios possibles (IDLE = aucune activité, RECV = réception/détection d'occupation du canal, SEND = envoi d'un message, SLEEP = sommeil –désactivé durant la simulation) montre que, comme supposé précédemment, les nœuds passent beaucoup plus de temps en mode RECV.

De fait, leurs fenêtres d'émission sont beaucoup plus réduites et ils gardent les paquets à émettre ou à relayer plus longtemps, ce qui a pour effet d'augmenter la taille moyenne des files MAC du réseau (jusqu'à les saturer quand le débit soumis devient trop important) et d'allonger le délai de réception des paquets.

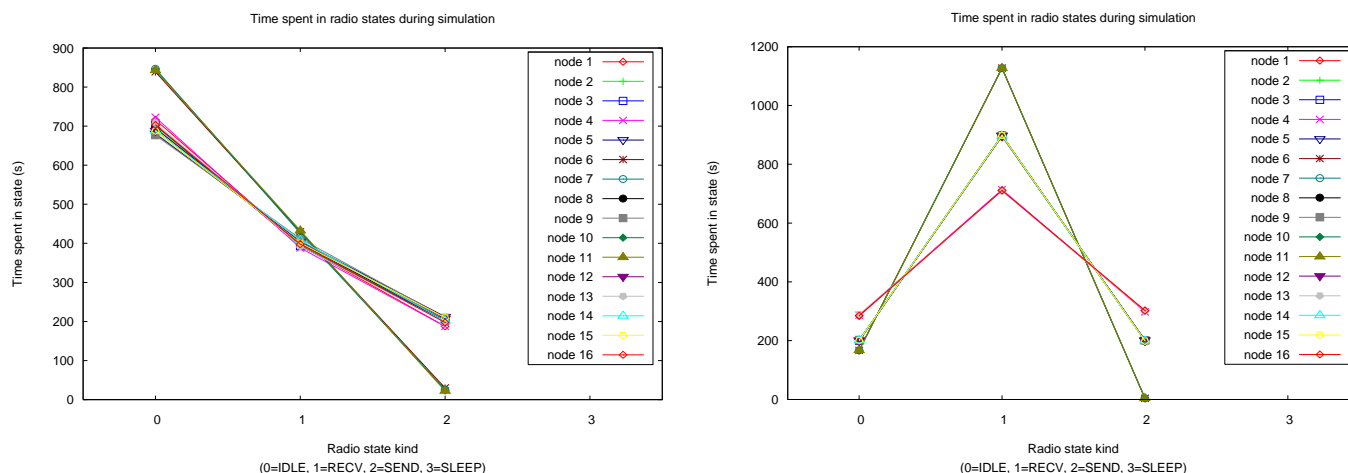


Figure 5.6 : Temps passé dans les différents états radio durant la simulation pour la zone de sensibilité réduite (à gauche) et étendue (à droite) pour un débit soumis de 800 kbits/s

5.1.2.2.3 Impact de la taille des files MAC sur le protocole de routage

Dans les simulations utilisant la zone de sensibilité étendue, l'interaction entre taille des files MAC et instabilité d'OLSR a été mise en évidence en observant l'évolution temporelle du nombre de paquet dans la file et le nombre de voisins connus d'un nœud du réseau (ici le nœud 7).

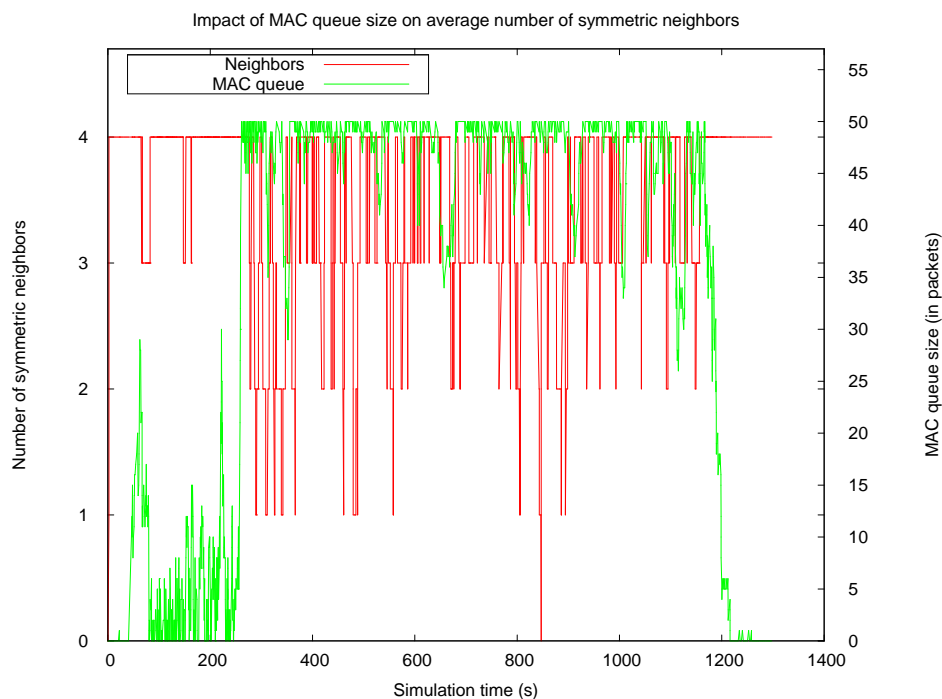


Figure 5.7 : Impact de la taille de la file MAC sur le fonctionnement d'OLSR

Trop longtemps stockés dans les files, les paquets d'information OLSR ne permettent pas au nœud de maintenir à jours leurs informations de topologie. Pour remédier à ce problème, le même scénario a été réalisé avec une file MAC dédiée et prioritaire pour les messages du protocole de routage.

5.1.3 OPTIMISATIONS DE LA SIMULATION

Par la suite, d'autres simulations ont été menées afin d'investiguer l'impact de différents facteurs susceptibles d'améliorer les résultats : l'utilisation d'une file MAC prioritaire pour les paquets OLSR, l'introduction d'une tolérance aux collisions sur les trames et l'ajustement de la zone de sensibilité à la porteuse.

5.1.3.1 TRAMES DE ROUTAGES PRIORITAIRES

Afin de limiter les dysfonctionnements du protocole de routage, notamment à forte charge lors de l'utilisation d'une zone de sensibilité à la porteuse étendue, une file MAC, vidée en priorité et contenant uniquement les paquets OLSR qu'un nœud souhaite émettre, a été implémentée.

Le comportement du réseau a été simulé à divers débits (de 100 kbits/s à 1300 kbits/s) avec une zone de sensibilité étendue. Les résultats obtenus sont synthétisés par la Figure 5.8.

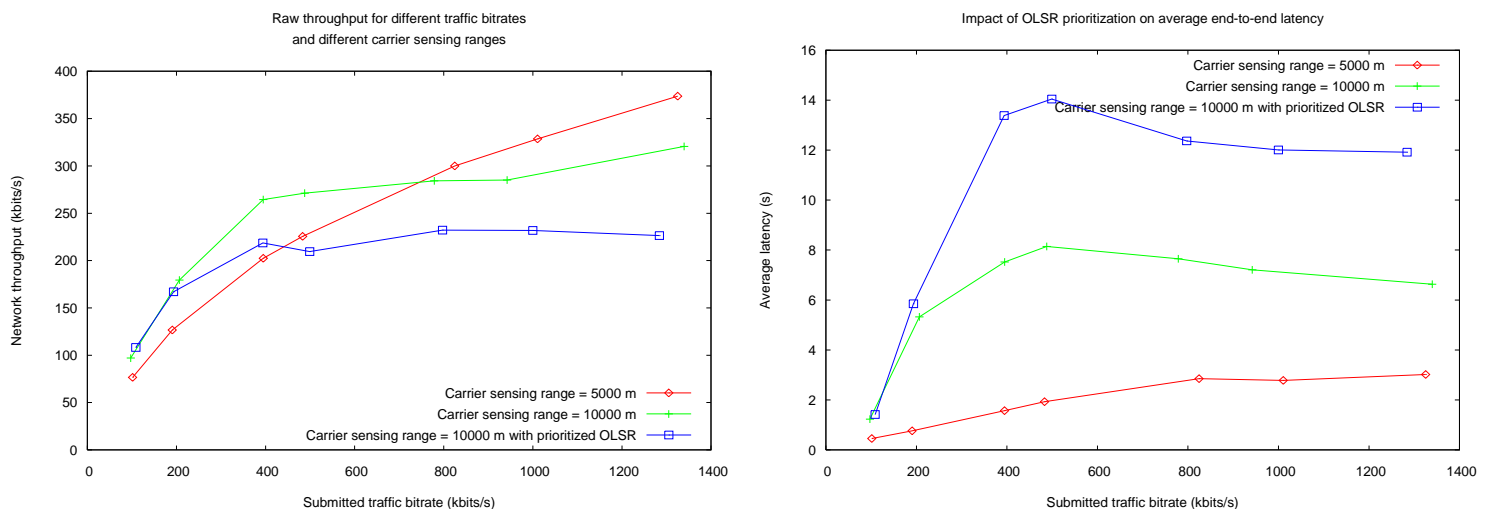


Figure 5.8 : Réponse en débit et temps de latence avec flux OLSR prioritaire

Contrairement aux attentes, le flux de routage prioritaire ne permet pas d'améliorer la réponse en débit du réseau. D'autre part, la latence point à point des paquets transmis s'en trouve allongé. Ceci s'explique par le fait que, notamment au-delà du fonctionnement nominal du réseau (>200 kbits/s), le trafic de contrôle écrase les transmissions des données (et ce malgré le fait que l'overhead mesuré ne dépasse pas 10%).

Avec une zone de sensibilité étendue, les nœuds n'accèdent que de manière trop disparate au canal. Lorsqu'ils y parviennent, un message OLSR est la plupart du temps en attente d'émission et celui-ci est donc envoyé sur le canal aux dépens du trafic de données.

A forte charge, il est même possible d'observer une congestion des messages de contrôle. Certains nœuds (principalement les nœuds centraux) accèdent trop peu au canal pour réguler la taille de leur file de paquets de routage, si bien qu'à partir de 400 kbits/s certains messages de contrôle sont supprimés pour cause de file pleine.

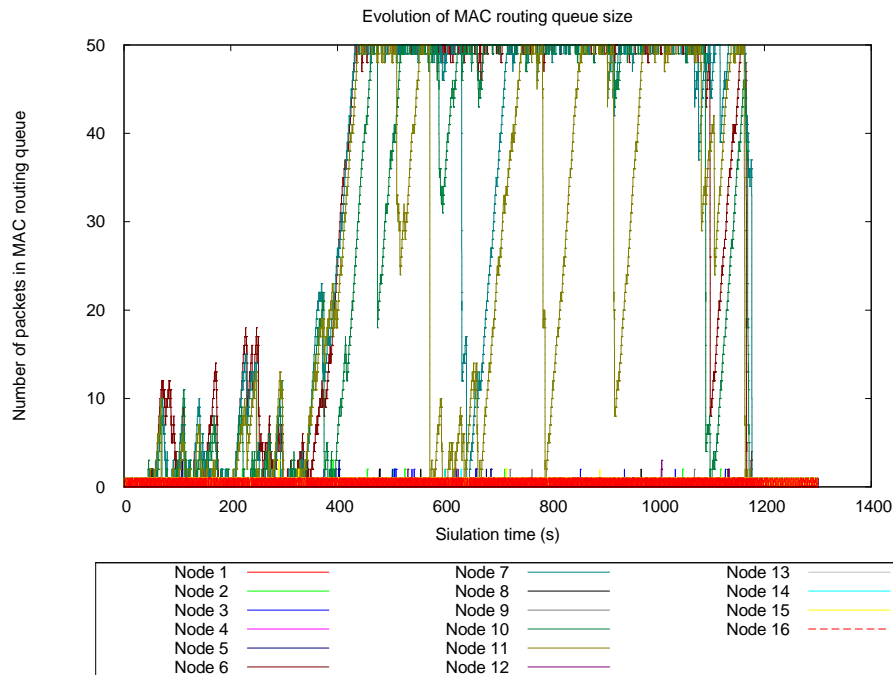


Figure 5.9 : Congestions du trafic de routage à 400 kbits/s

En contrepartie, la mise en priorité du flux OLSR permet au protocole de fonctionner plus correctement (moins de paquets perdus sur absence ou erreur de route) ce qui sera important dans les scénarios suivants, moins demandeurs en bande passante mais intégrant la mobilité des nœuds. Cette amélioration est en grande partie due à la décorrélation entre taille de la file MAC et fonctionnement du protocole de routage.

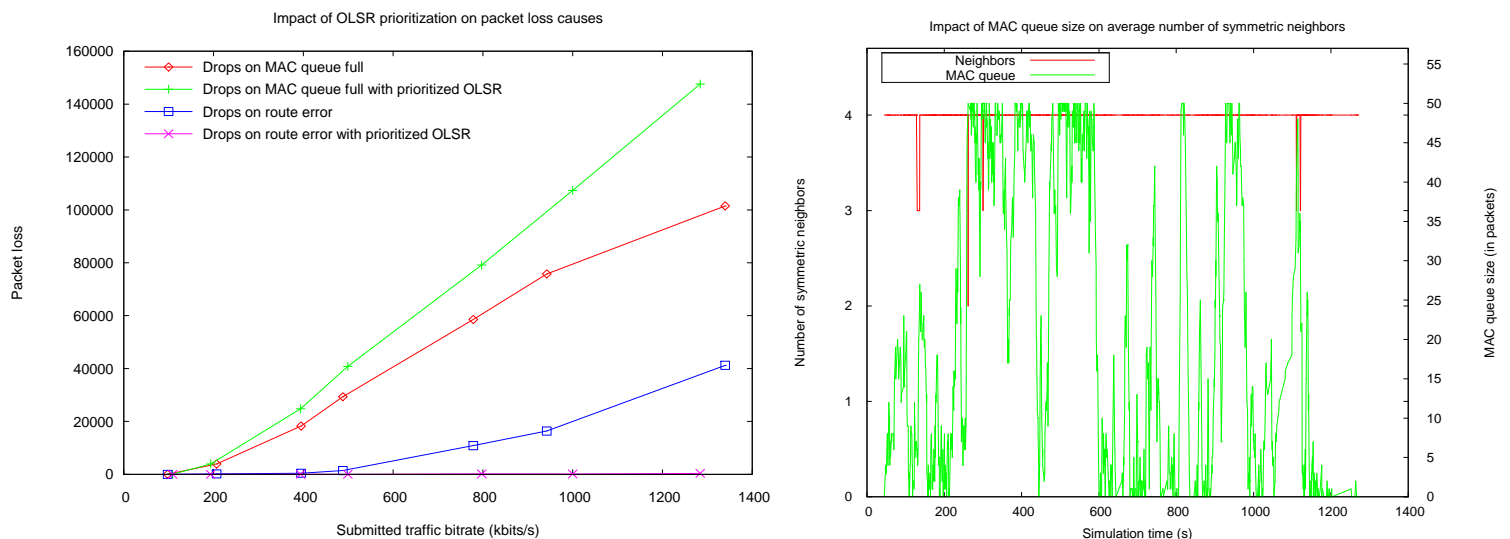


Figure 5.10 : Impact d'une file prioritaire OLSR sur le comportement du réseau (à g.) et du protocole de routage (à dr.)

5.1.3.2 TOLERANCE AUX COLLISIONS

Dans le cadre de communications tactiques, l'utilisation de codes correcteurs d'erreurs permet, à un débit de 547 kbits/s, de rendre les trames résistantes aux collisions jusqu'à 50% de leur durée.

Aussi, deux simulations soumettant le canal à un trafic usager de 500 kbits/s environ et implémentant une résistance aux collisions jusqu'à 50% des trames, ont été réalisées. La première avec une zone de sensibilité à la porteuse réduite, la seconde avec une zone étendue. Les valeurs de différentes métriques caractéristiques du réseau sont précisées dans le Tableau 5.2.

Tableau 5.2 : Impact de la résistance aux collisions

| Résistance | 0 % | | 0 % | 50 % | |
|------------------------------------|----------------|----------------|----------------|----------------|----------------|
| Priorité OLSR | NON | | OUI | OUI | |
| IF_range | réduit | étendu | étendu | réduit | étendu |
| Débit soumis (kbits/s) | 483,2 | 487,9 | 499,01 | 496,76 | 506,9 |
| Débit reçu (kbits/s) | 225,6 | 271,3 | 209,5 | 264,9 | 217,7 |
| Rapport | 0,47 | 0,56 | 0,42 | 0,53 | 0,43 |
| Délai* moyen (s) | 1,93 | 8,14 | 14,04 | 3,17 | 13,97 |
| Délai* maximal (s) | 22,14 | 501,4 | 1142,3 | 29,8 | 1151,7 |
| Écart type (s) | 2,23 | 13,8 | 76,83 | 3,17 | 87,07 |
| Collisions (paquets) | $2 \cdot 10^5$ | $4 \cdot 10^4$ | $4 \cdot 10^4$ | $4 \cdot 10^5$ | $4 \cdot 10^4$ |
| Pertes sur file pleine | 1770 | 29343 | 40841 | 5559 | 40805 |
| Pertes sur max. de retransmissions | 11348 | 138 | 201 | 16168 | 176 |
| Pertes sur erreur de routage | 23816 | 1416 | 0 | 10812 | 24 |

** Le délai calculé concerne le délai point à point entre émetteur et destinataire final du paquet*

L'introduction d'une résistance des paquets aux collisions ne semble avoir qu'un impact marginal sur les performances du réseau.

En effet, la latence moyenne est légèrement améliorée (du fait que certains paquets, jadis considérés comme corrompus sont désormais acceptés) même si cela reste faible.

D'autre part, les liaisons sont légèrement plus fiables, notamment grâce à la légère diminution des pertes sur file pleine et sur maximum de tentatives de transmission (dues au fait qu'un nombre un peu plus important de paquets considérés comme valides). Cette amélioration se paie par une légère déstabilisation d'OLSR qui engendre un nombre un peu plus important de perte par erreur de routage.

Par conséquent, dans les simulations suivantes, sauf indication contraire, une résistance aux collisions à hauteur de 50% de la durée du paquet sera implémentée.

5.1.3.3 ZONE DE SENSIBILITE INTERMEDIAIRE

La dernière série de simulation vise à utiliser une zone de sensibilité intermédiaire ($IF_range = \sqrt{2} \cdot TX_range + \Delta$) pour différents débits (200 kbits/s, 500 kbits/s, 800 kbits/s) avec une file OLSR prioritaire et une tolérance aux collisions de 50%. L'ensemble des mesures effectuées est repris dans le ci-dessous.

Tableau 5.3 : Impact de la zone de sensibilité intermédiaire

| Débit | 200 kbits/s | 500 kbits/s | 800 kbits/s |
|------------------------------------|------------------|----------------|----------------|
| IF_range | intermédiaire | intermédiaire | intermédiaire |
| Débit soumis (kbits/s) | 198,53 | 514,34 | 760,49 |
| Débit reçu (kbits/s) | 159,3 | 209,73 | 182,29 |
| Rapport | 0,80 | 0,41 | 0,24 |
| Délai* moyen (s) | 2,40 | 6,62 | 7,05 |
| Délai* maximal (s) | 28,13 | 90,63 | 95,79 |
| Écart type (s) | 3,20 | 8,44 | 10,24 |
| Collisions (paquets) | $1,5 \cdot 10^5$ | $3 \cdot 10^5$ | $3 \cdot 10^5$ |
| Pertes sur file pleine | 925 | 33109 | 69250 |
| Pertes sur max. de retransmissions | 4730 | 9760 | 11716 |
| Pertes sur erreur de routage | 1 | 55 | 209 |

* Le délai calculé concerne le délai point à point entre émetteur et destinataire final du paquet

L'introduction d'une zone de sensibilité intermédiaire (n'intégrant plus l'activité des voisins à deux bonds dans l'occupation du canal) permet d'améliorer les performances en terme de latence.

Quel que soit le débit, la latence de transmission des données est réduite et le fonctionnement du protocole OLSR est plus stable (moins d'erreur de routage). Ces deux effets sont principalement dus à un accès moins restrictif au canal, au prix d'un plus grand nombre de collisions.

Ce nombre accru d'interférences entre messages cause un plus grand nombre de pertes de paquets sur atteinte du maximum de tentatives de retransmission.

En conclusion, on retiendra que **l'ajustement de la zone de sensibilité à la porteuse est un bon moyen de trouver un compromis entre temps de latence et nombre de collisions sur le canal** et que, pour la topologie présentée en Figure 5.1, la distance de sensibilité intermédiaire apparaît comme un choix intéressant.

5.1.4 AMELIORATIONS DE LA FORME D'ONDE

Les améliorations apportées sur la forme d'onde (comme la priorité donnée aux données de contrôle ou la résistance aux collisions) ne suffisent pas à permettre au réseau de répondre correctement au débit soumis.

Parmi les solutions possibles permettant une amélioration du fonctionnement, tout en conservant la philosophie d'accès en contention, figurent :

- **L'extension de la couche MAC à une gestion du multicanal** (par exemple 3 ou 5 canaux Wi-Fi différents) qui permettrait d'autoriser des communications d'avoir lieu en parallèle ;
- **L'introduction de mécanismes de priorité des flux, et notamment de *fast forwarding***, permettant à un nœud relais de gagner avec une plus forte probabilité la prochaine contention et de relayer plus rapidement un paquet prioritaire (ce qui permettrait de réduire la latence point à point dans le cas d'applications ayant de fortes exigences dans ce domaine, comme la voix sur IP ou la vidéoconférence).

5.2 MOBILITE ALEATOIRE

Le second scénario propose de tester l'influence de la mobilité sur le fonctionnement du protocole de routage en présence de flux TCP et UDP dont le trafic simule une situation opérationnelle.

5.2.1 DESCRIPTION DU SCENARIO

Les nœuds se déplacent selon le modèle dit de *random waypoint mobility*, où chaque nœud choisit une destination aléatoire dans un domaine (ici un carré de 16 km de côté) et s'y rend avec une vitesse constante (60 km/h) et choisit une nouvelle destination après un temps de pause (30 s). La topologie de départ est la grille de 16 stations présentée dans la Figure 5.1.

Le trafic soumis simule l'envoi de message de *situation awareness* (informations sur la position courante) entre les nœuds d'un groupe et le chef de celui-ci. Les deux groupes définis sont $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8\}$ avec S_6 comme chef de groupe et $\{S_9, S_{10}, S_{11}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}\}$ avec S_{11} comme chef de groupe. Des messages UDP de 20 octets sont envoyés des nœuds vers le chef de groupe toutes les 3 secondes. Le chef de groupe émet quant à lui un trafic multicast (simulé par n envois unicast) de 120 octets toutes les 3 secondes vers les nœuds de son groupe.

D'autre part, 43 secondes après le début de la simulation, le transfert d'un fichier d'1 Mo utilisant TCP entre les deux chefs de groupe (S_6 vers S_{11}) débute.

La distance étendue de sensibilité est utilisée (TX_range = 5050m, IF_range = 10000m) et un facteur de résistance aux collisions de 50% est retenu. Les données du protocole de routage OLSR sont rendues prioritaires par l'utilisation d'une file MAC séparée.

5.2.2 PRESENTATION ET ANALYSE DES RESULTATS

La première analyse porte sur la qualité du transfert des données UDP (trafic de *situation awareness*) dont différentes métriques sont présentées dans le Tableau 5.4

Tableau 5.4 : Résultats de la simulation à mobilité aléatoire (trafic UDP)

| | |
|---|-------|
| Débit soumis (kbits/s) | 5,24 |
| Débit reçu (kbits/s) | 4,77 |
| Rapport | 0,909 |
| Délai* moyen (s) | 0,124 |
| Délai* maximal (s) | 6,44 |
| Écart type (s) | 0,163 |
| Collisions (paquets) | 5706 |
| Pertes sur file pleine | 3 |
| Pertes sur max. de retransmissions | 466 |
| Pertes sur erreur de routage | 0 |
| Overhead OLSR (kbits/s) | 11,56 |

* Le délai calculé concerne le délai point à point entre émetteur et destinataire final du paquet

Avec une quantité de données reçues supérieure à 90% et un délai moyen de 120 ms environ, le comportement du réseau est très satisfaisant. Même si la gigue reste importante, les résultats sont compatibles avec l'application de *situation awareness*.

L'étude du déplacement des stations montre que la grille initiale (déjà peu favorable à un accès en contention avec zone de sensibilité à la porteuse étendue) se déforme et crée une forte concentration des stations dans une même zone, ce qui engendre un canal peu souvent libre, entraînant une augmentation de la latence.

Le caractère prioritaire des données de routage permet d'assurer le bon fonctionnement d'OLSR (pas de pertes sur absence de route, taille de la file MAC de données de routage toujours inférieure à deux paquets).

La présence d'un flux TCP, même s'il génère des files assez importante pour les nœuds, source, relais et destination de ce trafic, n'empêche pas une livraison correcte des messages de positions (émis par les nœuds vers leurs chef de groupe toutes les 3 secondes).

Le débit obtenu pour le flux TCP est de 166,30 Kbits/s. Selon la configuration initiale du réseau, un relais radio est nécessaire pour acheminer les données. Pendant le scénario, puisque les nœuds se meuvent, cette situation peut changer. Aussi, il peut être utile de mesurer l'influence du nombre de relais (ou nombre de bonds) sur le débit TCP avec la forme d'onde étudiée.

En conclusion, la simulation montre que **la mobilité est globalement supportée et que l'architecture protocolaire semble répondre à des besoins opérationnels même si la gigue reste importante et que la fiabilité des liaisons doit être accrue, notamment en limitant les pertes sur files pleines (c'est à dire en désengorgeant certains nœuds).**

5.2.3 INFLUENCE DU NOMBRE DE BONDS SUR LE DEBIT TCP

Le fait que la mobilité de ce scénario soit aléatoire rend difficile l'établissement de conclusion fiables (notamment car il est difficile de faire corrélés résultats et évolution de la topologie du réseau).

Par contre, il est intéressant de mesurer l'évolution du débit du transfert de fichier (utilisant le protocole TCP) en fonction du nombre de bonds radio. Ces mesures ont été effectuées avec une zone de sensibilité à la porteuse étendue (10000m), en faisant varier la source et la destination du trafic TCP (soit en présence du seul trafic de routage, soit en l'absence de toute autre communication sur le réseau) sur la grille de 16 stations.

La forte différence de débit (favorable à l'utilisation d'une zone étendue de sensibilité à la porteuse) s'explique en grande partie par le nombre de collisions et de pertes sur maximum de tentatives de transmissions nettement plus importantes quand la zone de sensibilité est réduite.

Aussi, ces erreurs et retransmissions ont un double impact négatif : non seulement ils occasionnent une baisse directe du débit (moins de paquets différents transmis pendant la même durée) mais ils occasionnent aussi une limitations des performances de TCP (diminution de la fenêtre de congestion sur réception de trois *DUPACKs* ou sur expiration d'un *timer*, voir paragraphe 3.2.2.2.2).

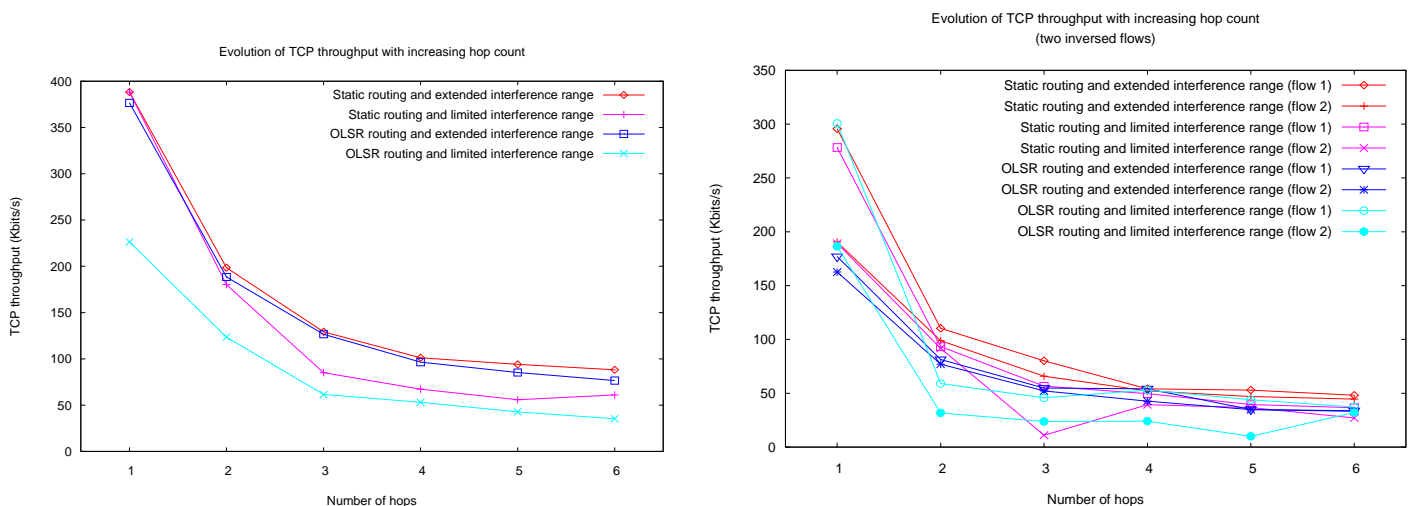


Figure 5.11 : Influence du nombre de bonds et de la zone de sensibilité à la porteuse sur le débit TCP (pour 1 flux à gauche, pour deux flux tête-bêche à droite)

Par contre, l'influence de l'introduction du protocole de routage perturbe assez peu le débit (les données étant envoyées en broadcast, il y a assez peu d'overhead associé) même si l'impact est plus marqué quand la sensibilité à la porteuse est réduite.

Enfin, l'introduction d'un flux TCP inverse diminue le débit de chacun des flux d'un facteur deux environ à partir de 2 nœuds. Il est de plus intéressant de noter qu'il existe une disparité entre les 2 flux apparaît. C'est toujours le flux initié en premier qui obtient un débit plus important ainsi que présenté dans le paragraphe 3.2.2.2.3. Cet effet tend à disparaître quand la sensibilité à la porteuse est étendue.

5.2.4 AMELIORATIONS DE LA FORME D'ONDE

En terme de débit pour le transfert de fichier utilisant le protocole TCP, **l'accès en contention montre son efficacité** avec un débit de plus de 100 kbits/s.

Concernant le trafic UDP (communications de *situation awareness*), la fiabilité reste correcte. Les contraintes de latence sont quant à elles assez bien respectées par la forme d'onde avec accès en contention (125 ms en moyenne). Toutefois, le délai maximal est de plus de 6 secondes dans la simulation Wi-Fi/OLSR, ce qui montre que **l'accès en contention ne garantit pas de délai et ne permet pas de limiter la gigue** (notamment lorsqu'il existe une transaction TCP concurrente).

Cette information est importante et implique la **nécessité d'un routage avec qualité de service, pour permettre par exemple de faire cohabiter un transfert de fichier** par FTP (utilisant le protocole TCP, demandant une bande passante importante et pour lequel la latence dans la transmission n'a qu'une importance moindre) **et des communications VoIP** ou de visioconférence par IP (utilisant UDP et amenant de fortes contraintes sur la latence et la gigue des paquets).

5.3 100 STATIONS DONT 4 MOBILES

Sur une vaste grille de 100 stations, ce scénario propose de tester la capacité à maintenir des flux (TCP et UDP) entre diverses stations lorsque les émetteurs sont en mouvement, dans un réseau où un trafic d'arrière plan (*situation awareness*) est présent.

5.3.1 DESCRIPTION DU SCENARIO

Les nœuds sont disposés sur une grille carrée de 100 stations. La distance horizontale/verticale entre chaque nœud est de 5 km. La zone de transmission (*TX_range*) est adaptée afin de permettre une connectivité telle que décrite sur la Figure 5.12. Une zone de sensibilité à la porteuse (*IF_range*) étendue a été utilisée (égale à deux fois la zone de transmissions). Les données de routage sont rendues prioritaires par l'utilisation d'une file MAC séparée et une résistance des paquets aux collisions de 50% a été retenue.

Durant le scénario, les nœuds sont répartis en groupes opérationnels (repérés par des couleurs différentes sur la Figure 5.12). Du trafic d'information de position est échangé entre les nœuds d'un groupe et leur chef (matérialisé par un trait de bordure épais).

De plus, les nœuds N_1 , N_2 , N_{11} et N_{45} se déplacent le long de la deuxième diagonale de la grille à une vitesse de 60 km/h environ. Durant leur déplacement, N_1 et N_{45} effectuent une communication utilisant la voix sur IP et N_2 et N_{11} échangent un fichier par FTP respectivement avec les nœuds N_{10} et N_{91} .

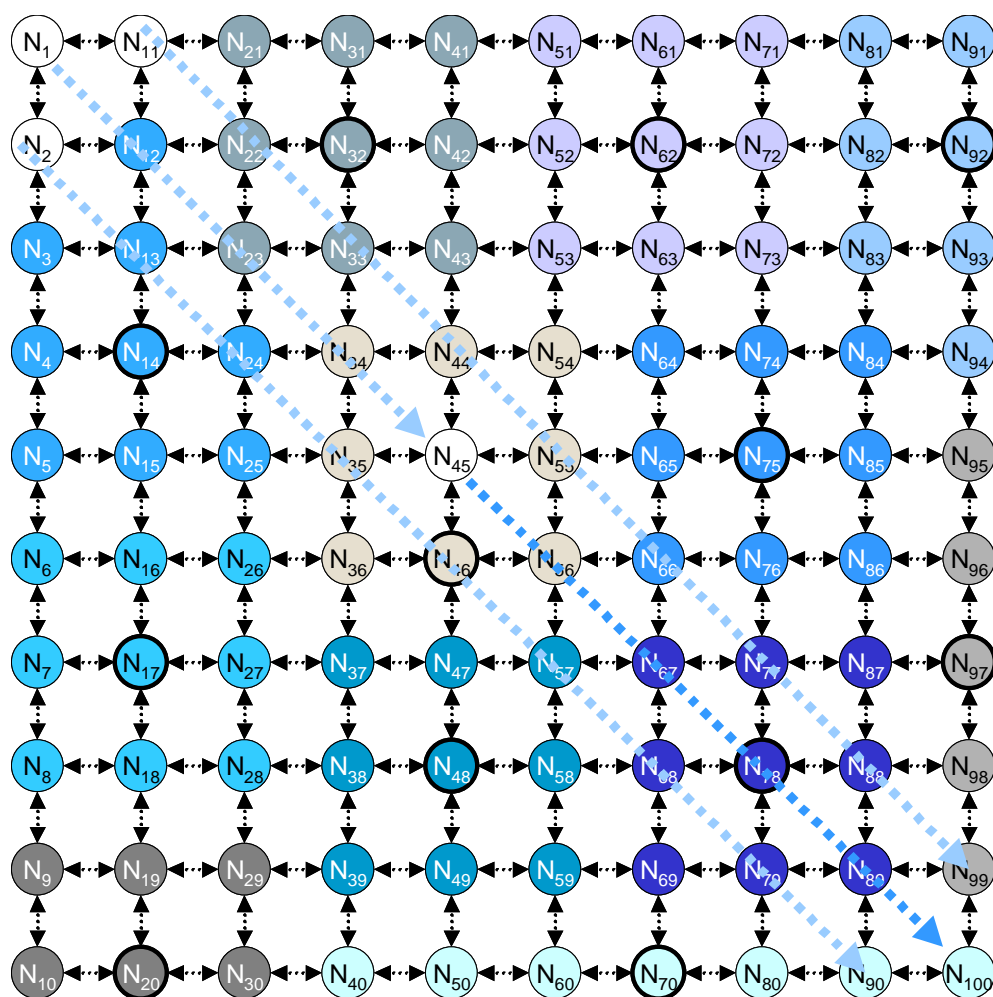


Figure 5.12 : Grille de 100 stations

5.3.2 PRESENTATION ET ANALYSE DES RESULTATS

Les résultats obtenus avec le scénario décrit précédemment sont consignés dans le Tableau 5.5 qui distingue le trafic UDP de *situation awareness* et de voix sur IP.

Tableau 5.5 : Résultats de simulation avec la grille de 100 stations

| Type de trafic | S.A | VoIP | Total |
|------------------------------------|-------|-------|--------|
| Débit soumis (kbits/s) | 32,52 | 8,535 | 41,095 |
| Débit reçu (kbits/s) | 32,23 | 7,29 | 39,52 |
| Rapport | 0,991 | 0,929 | 0,961 |
| Délai* moyen (s) | 0,168 | 0,718 | 0,330 |
| Délai* maximal (s) | 22,02 | 17,78 | 22,02 |
| Écart type (s) | 0,472 | 0,879 | 0,501 |
| Collisions (paquets) | - | - | 652694 |
| Pertes sur file pleine | - | - | 6100 |
| Pertes sur max. de retransmissions | - | - | 817 |
| Pertes sur erreur de routage | - | - | 1 |
| Overhead OLSR (kbits/s) | - | - | 643,13 |

Le comportement du réseau dans le cadre de ce scénario est mitigé. D'une part, la fiabilité des liaisons (qui est toujours supérieure à 90 %) est correcte. D'autre part, la latence est beaucoup trop élevée, notamment pour les flux de VoIP.

Cette mesure, corrélée avec le nombre élevé de pertes sur files pleines et le nombre impressionnant de collisions, montre que le réseau est saturé, même si le débit utilisateur soumis est modéré (41 Kbits/s au total).

Cet engorgement du réseau est du à l'overhead engendré par OLSR qui atteint presque 650 kbits/s. Les analyses menées sur les fichiers de traces du réseau montrent que chaque nœud a en moyenne 3.6 MPRs durant la simulation. Aussi, la plupart du temps, chaque nœud choisit tous ses voisins comme MPRs.

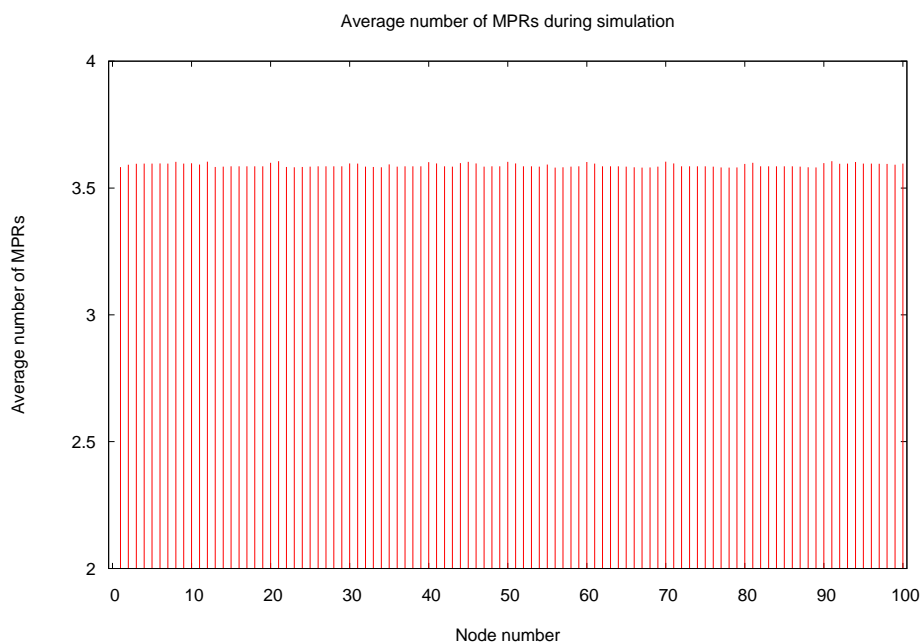


Figure 5.13 : Nombre moyen de MPRs pendant la simulation

Cette caractéristique est une des limitations d'OLSR qui tend à supprimer toutes ces optimisations du mécanisme d'inondation du réseau et qui engendre les faits suivants :

- Chaque nœud du réseau (à l'exception des nœuds des coins) est élu MPR d'au moins un autre nœud du réseau, ce qui tend à augmenter la taille des différents ensembles de nœuds relatifs au fonctionnement d'OLSR et donc la taille des messages de contrôle du protocole (qui atteignent rapidement la taille maximale des trames MAC 802.11) ;
- Les messages TC, qui sont relayés de MPRs en MPRs, parviennent de manière multiple aux divers nœuds du réseau, ce qui génère un overhead important.

Ces deux constatations montrent que dans le cas d'une topologie en grille (ce qui s'avère être le cas le plus défavorable pour OLSR, l'overhead augmente en N^2 (N étant la taille du réseau)). Cette assertion est vérifiée en faisant le rapport avec l'overhead mesurée sur une grille de 16 nœuds et qui avoisinait les 15 kbits/s).

Le débit TCP fourni par le réseau s'élève respectivement à 47,09 Kbits/s (N_2 vers N_{10}) et 21,41 Kbits/s (N_{11} vers N_{91}), ce qui semble satisfaisant même au vu du nombre de bonds radio utilisés.

5.4 CLUSTER DE 7 STATIONS ET MOBILITE

Le scénario suivant propose d'étudier le comportement de la forme d'onde dans une topologie proche d'une situation opérationnelle avec un trafic correspondant à divers besoins tactiques (*situation awareness*, VoIP, transfert de données).

5.4.1 DESCRIPTION DU SCENARIO

Les stations 1,2,3,6 et 7 sont fixes et forment un *cluster* dont le chef de groupe est 1. Les stations 4 et 5 sont mobiles et vont traverser et se joindre au cluster. Elles se déplacent en ligne droite à environ 20 km/h.

La portée de transmission a été ajustée pour obtenir le maillage suivant. La zone de sensibilité à la porteuse a été fixée à deux fois cette valeur.

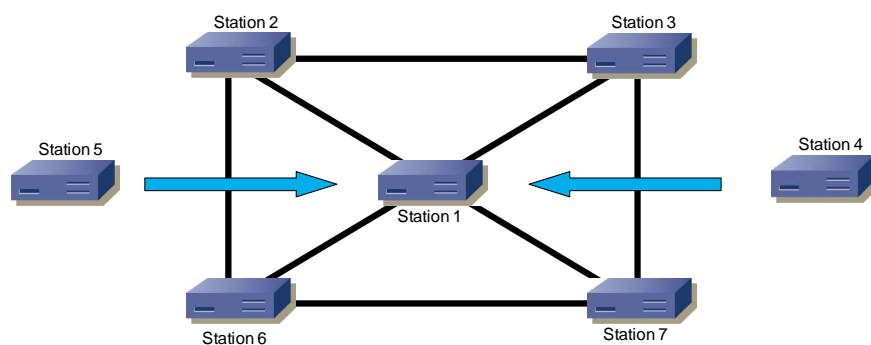


Figure 5.14 : Scénario de simulation avec cluster de 7 stations

Le trafic échangé mêle des messages de *situation awareness* (entre les nœuds et le chef de cluster et réciproquement), des communications VoIP entre le chef de cluster et les autres nœuds et un échange de fichier par FTP de la station 4 à la station 2.

La distance étendue de sensibilité est utilisée (TX_range = 5050m, IF_range = 10000m) et un facteur de résistance aux collisions de 50% est retenu. Les données du protocole de routage OLSR sont rendues prioritaires par l'utilisation d'une file MAC séparée.

5.4.2 PRESENTATION ET ANALYSE DES RESULTATS

Les résultats obtenus révèlent que la forme d'onde Wi-Fi/OLSR se montre très performante sur ce scénario.

Le trafic UDP (*situation awareness* et voix sur IP) est acheminé sans trop de perte (94,6% de réception au total et 94,8% pour la voix sur IP) avec des délais acceptables même si un peu trop élevés.

Les principales causes de pertes sont liées à des débordements de file (notamment du fait que les durées inter-trames soit élevées) et à l'atteinte du maximum de retransmission (soit à cause de collisions, soit parce que la route utilisée n'est plus valable). Ces problèmes surviennent quasiment exclusivement sur la station 1, qui est au centre du cluster. Elle est donc le plus exposée à l'occupation du canal et aux collisions. De plus, c'est elle qui génère et qui reçoit le plus de trafic.

Notons que, **le contrôle d'accès au canal fonctionne correctement avec la zone de sensibilité étendue**, puisque les seules collisions notables concernent des RTS entrant en conflit avec d'autres RTS (ce qui est le principal inconvénient du mécanisme de *binary exponential backoff*).

Tableau 5.6 : Résultats de la simulation avec cluster de stations (trafic UDP)

| Type de trafic | Total | dont VoIP |
|------------------------------------|-------|-----------|
| Débit soumis (kbits/s) | 53,21 | 51,20 |
| Débit reçu (kbits/s) | 50,33 | 48,34 |
| Rapport | 0,946 | 0,944 |
| Délai* moyen (s) | 0,059 | 0,068 |
| Délai* maximal (s) | 7,29 | 1,97 |
| Écart type (s) | 0,167 | 0,165 |
| Collisions (paquets) | 1658 | - |
| Pertes sur file pleine | 2083 | - |
| Pertes sur max. de retransmissions | 560 | - |
| Pertes sur erreur de routage | 69 | - |
| Overhead OLSR (kbits/s) | 2,71 | - |

* Le délai calculé concerne le délai point à point entre émetteur et destinataire final du paquet

Le débit TCP entre la station 4 et la station 2 atteint quant à lui 109,26 Kbits/s, ce qui correspond, d'après la Figure 5.11 aux débits obtenus lors d'une connexion à deux bords. Cette hypothèse semble vraisemblable au vu de la topologie et de la mobilité adoptées.

Enfin, le fonctionnement d'OLSR semble satisfaisant. Outre les paquets perdus sur maximum de retransmission, les erreurs par absence de route sont quasi-nulles, et ce avec un *overhead* limité à moins de 3 kbits/s.

5.4.3 IMPACT DE LA TAILLE DE LA ZONE DE SENSIBILITE A LA PORTEUSE

Une simulation avec le même scénario a été réalisée en diminuant la zone de sensibilité à la porteuse (et en la rendant égale à la zone de transmission des données). Les résultats obtenus avec cette configuration sont consignés dans le Tableau 5.7 ci-dessous.

Tableau 5.7 : Résultats de la simulation avec cluster de stations et zone de sensibilité à la porteuse réduite (trafic UDP)

| Type de trafic | Total | dont VoIP |
|------------------------------------|-------|-----------|
| Débit soumis (kbits/s) | 53,21 | 51,20 |
| Débit reçu (kbits/s) | 50,31 | 48,35 |
| Rapport | 0,945 | 0,944 |
| Délai moyen (s) | 0,059 | 0,029 |
| Délai maximal (s) | 6,55 | 1,75 |
| Écart type (s) | 0,236 | 0,118 |
| Collisions (paquets) | 10702 | - |
| Pertes sur file pleine | 1621 | - |
| Pertes sur max. de retransmissions | 1536 | - |
| Pertes sur erreur de routage | 634 | - |
| Overhead OLSR (kbits/s) | 2,73 | - |

La réduction de la zone de sensibilité à la porteuse, si elle fait logiquement croître le nombre de collisions d'un facteur 10, n'a qu'un effet limité sur la fiabilité des communications dans le réseau (toujours un peu plus de 94% des émissions sont correctement reçues).

Cette dégradation des performances est liée à l'augmentation des pertes sur maximum de retransmission du fait d'une connaissance plus limitée de l'occupation du canal qui rend possible des collisions entraînées par la fragilité du mécanisme de signalisation.

D'autre part, le fonctionnement du protocole de routage est également affecté puisque les erreurs sur absence de route deviennent non négligeables. Les paquets OLSR étant envoyés en broadcast (donc sans mécanisme RTS/CTS), seule la sensibilité à la porteuse permet d'éviter les collisions lors de leur envoi, la réduction de cette dernière leur est donc fortement préjudiciable.

Comme dans les comparaisons précédentes, la diminution de la zone de sensibilité à la porteuse permet de diminuer la latence moyenne d'acheminement des paquets, notamment dans le cas de la VoIP. Cette amélioration s'obtient en contrepartie d'une perte de la fiabilité des liaisons, qui nécessite un plus grand nombre moyen d'émission des paquets pour les transmettre avec succès, ce qui explique la plus grande gigue au sein du réseau.

En conclusion, si elle permet d'améliorer légèrement la latence, la réduction de la zone de sensibilité à la porteuse rend le réseau moins fiable, et difficilement utilisable pour des applications de voix sur IP.

Le débit TCP chute à 59,52 kbits/s avec une zone de sensibilité réduite. Le même constat avait été effectué lors de l'analyse de l'influence du nombre de bonds sur le débit TCP (paragraphe 5.2.3). On peut donc supposer que les mêmes causes ont eu le même effet et ainsi causé la chute du débit de la transaction TCP.

5.5 FUSION DE RESEAUX

Cette simulation envisage un cas fréquent de situations opérationnelles : la fusion de deux sous-réseaux passant à proximité l'un de l'autre et qui vont pouvoir échanger des informations.

5.5.1 DESCRIPTION DU SCENARIO

Cette simulation met en œuvre deux sous-réseaux composés de neuf nœuds disposés en grille. Chaque nœud est distant de son voisin de 1000 mètres. La distance de transmission (TX_range) est ajustée à 1600 mètres (elle couvre donc également les diagonales de la grille). Une distance de sensibilité à la porteuse étendue ($IF_range = 3200$ mètres) est utilisée. Le flux de routage OLSR est rendu prioritaire par l'utilisation d'une file MAC séparée dans chacun des nœuds et les trames sont supposées résistantes aux collisions et interférences jusqu'à 50% de leur durée totale.

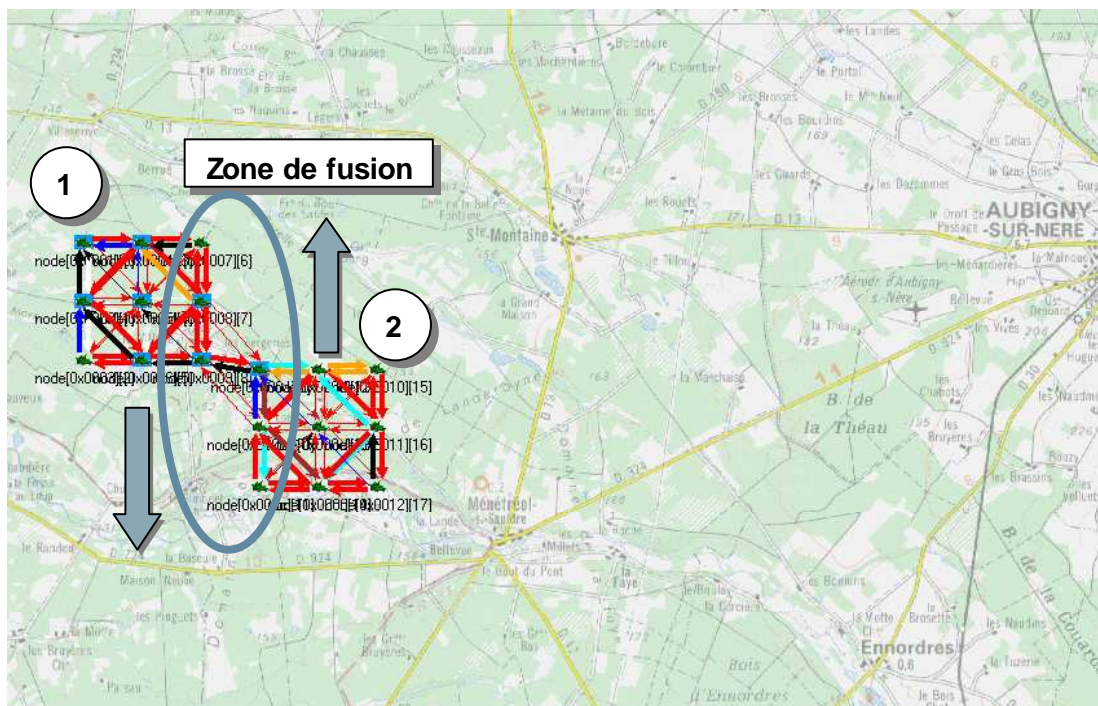


Figure 5.15 : Scénario de fusion de réseau

Les nœuds de chacune des grilles se déplacent sans déformer celle-ci à environ 70 km/h. Durant la simulation les nœuds les plus à droite de la grille 1 sur la Figure 5.15 (soit les nœuds N_7 , N_8 , N_9) vont entrer en portée avec les nœuds les plus à gauche de la grille 2 (soit les nœuds N_{10} , N_{11} , N_{12}).

Pendant le scénario, un trafic de *situation awareness* est échangé entre le chef de groupe (le nœud haut-gauche de chacune des grilles, soit respectivement N_1 et N_{10}) et les autres nœuds de la grille. De plus, des informations de positionnement sont également échangées entre les deux chefs de groupe. Enfin, une conversion de voix sur IP de N_1 vers N_2 , N_3 , N_4 , N_5 et N_6 vient charger la première grille.

Une transmission utilisant TCP (simulant un transfert de fichier via FTP) est également opérée de manière bidirectionnelle entre les nœuds les plus distant du réseau fusionné (N_1 et N_{18}) pendant que les deux réseaux sont interconnectés.

5.5.2 PRESENTATION ET ANALYSE DES RESULTATS

Les résultats relatifs au trafic UDP soumis au réseau pendant la simulation sont résumés dans le Tableau 5.8 ci-dessous.

Tableau 5.8 : Résultats pour la simulation de fusion de réseaux (trafic UDP)

| Type de trafic | S.A | VoIP | Total |
|------------------------------------|-------|-------|-------|
| Débit soumis (kbits/s) | 5,89 | 42,67 | 48,56 |
| Débit reçu (kbits/s) | 4,32 | 42,60 | 44,92 |
| Rapport | 0,733 | 0,911 | 0,905 |
| Délai* moyen (s) | 0,153 | 0,315 | 0,222 |
| Délai* maximal (s) | 5,34 | 0,740 | 5,34 |
| Écart type (s) | 0,125 | 0,044 | 0,091 |
| Collisions (paquets) | - | | 2676 |
| Pertes sur file pleine | - | | 12545 |
| Pertes sur max. de retransmissions | - | | 22 |
| Pertes sur erreur de routage | - | | 396 |
| Overhead OLSR (kbits/s) | - | | 11,57 |

* Le délai calculé concerne le délai point à point entre émetteur et destinataire final du paquet

Ce scénario, au trafic certes limité, se montre assez favorable à la forme d'onde Wi-Fi/OLSR. En particulier, **la fiabilité des communications est correcte avec plus de 91% de paquets correctement reçus.**

Les délais d'acheminement sont par contre incompatibles avec du trafic de voix sur IP (délai moyen de 315 ms qui ne permet pas une utilisation correcte de cette application),

Les flux présentant le délai le plus important sont les flux de *situation awareness* de la première grille et ceux échangés entre les deux chefs de groupe (notamment car ces messages sont ceux qui sont relayés par le plus de bonds radio).

Les flux de VoIP sont acheminés avec une latence bien plus grande que les flux de *situation awareness*. Ceci est dû au fait que les communications sont échangées sur la première grille qui est déjà chargée par différents flux de *situation awareness* et TCP (échange de fichiers par FTP).

Par conséquent, il semble important de mettre en place une technique de **qualité de service** afin de garantir le respect des exigences de délai pour la voix sur IP.

Le protocole de routage, malgré la topologie de grille retenue pour les sous-réseaux, s'avère efficace, puisque peu de paquets sont perdus par absence de route. De plus, **les changements de topologie sont détectés relativement rapidement (2,5 à l'apparition, et 6,5 secondes à la disparition, soit 0,5 après l'expiration du timer NEIGHB_HOLD_TIME)** ce qui permet à chacun des nœuds d'avoir une image assez correcte de son voisinage réel ce qui entraîne une faible perte sur erreur de route (partie des pertes par atteinte du maximum de retransmission).

Enfin, comme la taille de chaque sous-réseau est faible, l'overhead du protocole reste limitée.

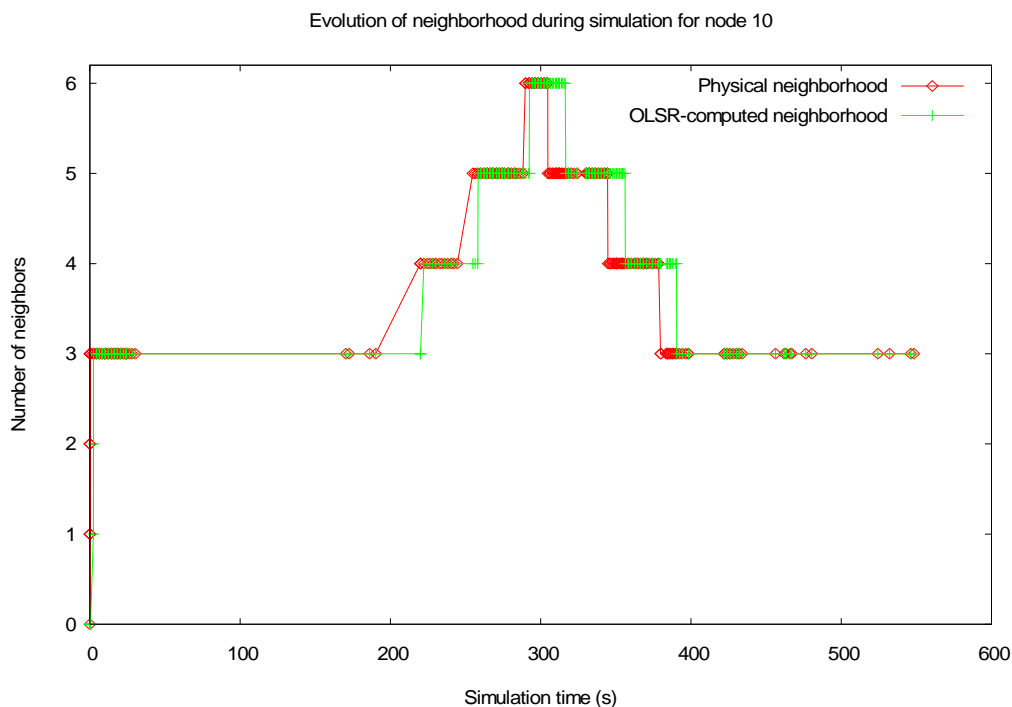


Figure 5.16 : Évolution du voisinage (physique et vu d'OLSR) pour le nœud 10 durant la simulation

Concernant le flux TCP échangé entre les deux chef de groupe, on observe une forte limitation du débit et une disparité entre les deux flux. Ainsi, le flux établi entre N_1 et N_{18} atteint un débit de 9,68 Kbits/s alors que le flux inverse (N_{18} vers N_1) ne parvient même pas à s'établir, du fait de l'engorgement de la première grille en général et du nœud 1 en particulier. Les causes de cet engorgement sont décrites et analysées dans le paragraphe 5.6.2 qui présente une situation similaire.

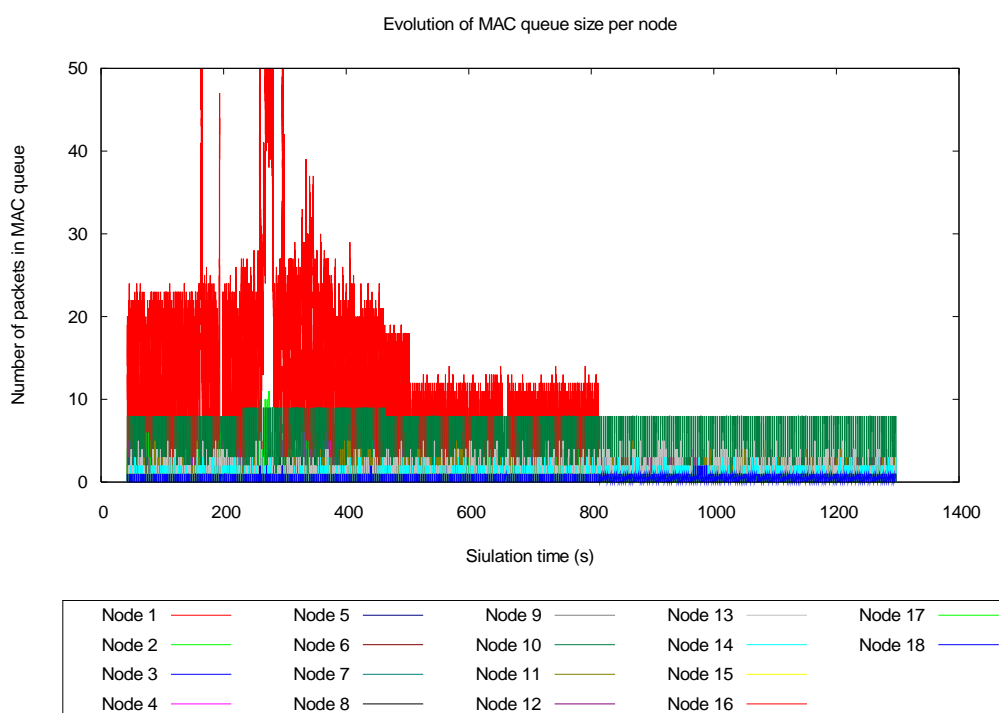


Figure 5.17 : Engorgement du nœud 1

La limitation du débit s'explique en partie par la charge du réseau pendant la transaction, conjuguée au nombre élevé de bonds. Les valeurs théoriques présentées en Figure 5.11 sont obtenues en l'absence de tout autre trafic sur le réseau. **La présence d'un autre trafic cause donc une baisse importante du débit TCP, surtout quand un nombre important de bonds est nécessaire.**

Par conséquent, **l'introduction d'un routage optimisé intégrant la charge des nœuds (*load balancing*) semble indispensable pour maintenir les métriques de latence et de débit à des niveaux corrects pour les flux le nécessitant. Ce protocole de routage QoS permettant d'assurer le passage des flux TCP et un meilleur respect des délais pour les flux de VoIP.**

5.6 SCENARIO OPERATIONNEL : SECURISER BRINON

Le dernier scénario d'évaluation des performances se base sur une situation opérationnelle dans laquelle diverses unités militaires s'attachent à sécuriser un objectif, la ville de Brinon.

Mettant en jeux plusieurs unités aux missions différentes, parmi lesquelles le Génie (reconnaissance et franchissement), l'artillerie (appui feu), des unités de blindés et d'infanterie mécanisée (sécurisation), ce scénario permet de démontrer et de tester les différents services « ad-hoc » offerts par la forme d'onde (initialisation et fusion de réseaux, adaptation automatique au changement de topologie, routage et acheminement automatique du trafic.

5.6.1 DESCRIPTION DU SCENARIO

Pendant le déroulement du scénario, les unités de reconnaissance, assistés par des membres du génie vont progresser vers l'objectif. Ils sont appuyés par une unité d'infanterie mécanisée au nord, d'unités de blindés qui vont créer une zone d'interdiction au sud de la ville, et d'unités d'artillerie assurant un appui feu depuis l'arrière. Le nœud 1, représentant le QG opérationnel de la mission reste fixe.

Les nœuds mobiles échangent des informations de *situation awareness* pendant tout le déroulement de la mission. De plus, le nœud 2, chargé de la mission de reconnaissance émet un flux de voix sur IP point-multipoint (simulé ici par N flux point à point) vers les autres nœuds du réseau (y compris le nœud fixe). Enfin, un transfert de fichier par FTP est également simulé entre les nœuds 2 et 7 et entre les nœuds 3 et 8.

La zone de transmission adoptée à un rayon de 7km autour du nœud émetteur et une zone de sensibilité à la porteuse étendue (14km) est utilisée. De plus, les trames émises sont supposées résistantes aux interférences jusqu'à concurrence de 50% de leur durée.

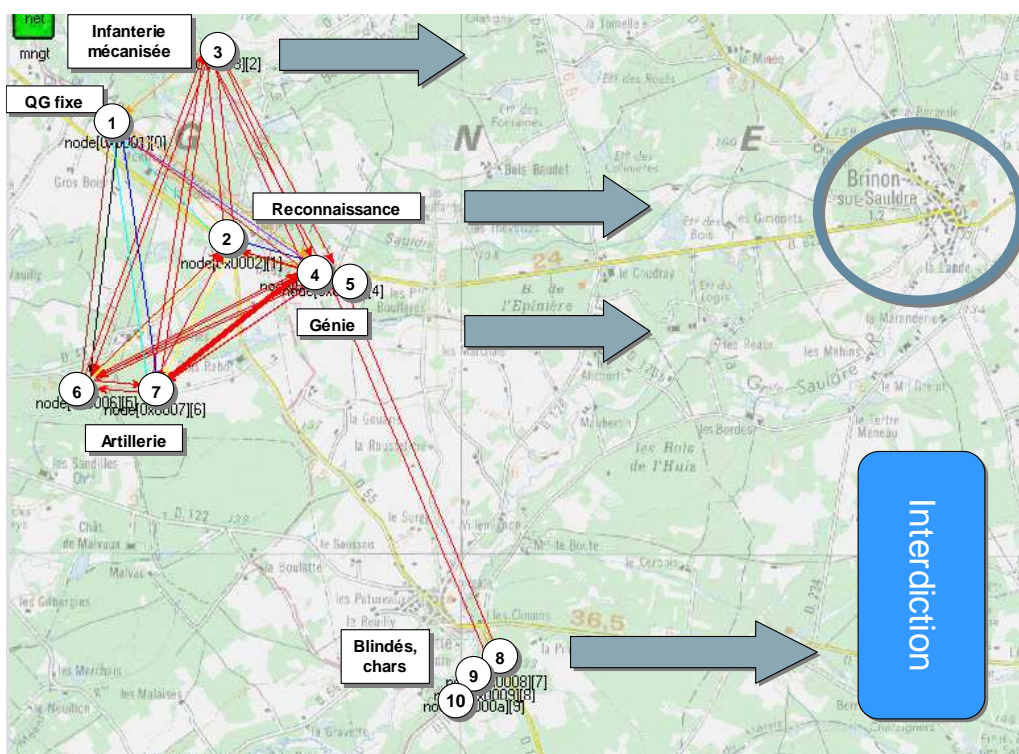


Figure 5.18 : Scénario opérationnel (sécuriser Brinon)

5.6.2 PRESENTATION ET ANALYSE DES RESULTATS

Les résultats obtenus avec le scénario décrit précédemment sont consignés dans le Tableau 5.9 qui distingue le trafic UDP de *situation awareness* et de voix sur IP.

Tableau 5.9 : Résultats de simulation avec la grille de 100 stations

| Type de trafic | S.A | VoIP | Total |
|------------------------------------|-------|--------|--------|
| Débit soumis (kbits/s) | 3,637 | 76,816 | 80,453 |
| Débit reçu (kbits/s) | 3,542 | 43,171 | 55,722 |
| Rapport | 0,974 | 0,562 | 0,581 |
| Délai* moyen (s) | 0,129 | 0,204 | 0,186 |
| Délai* maximal (s) | 24,80 | 3,384 | 24,80 |
| Écart type (s) | 0,240 | 0,072 | 0,112 |
| Collisions (paquets) | - | - | 5531 |
| Pertes sur file pleine | - | - | 121137 |
| Pertes sur max. de retransmissions | - | - | 414 |
| Pertes sur erreur de routage | - | - | 23238 |
| Overhead OLSR (kbits/s) | - | - | 4,83 |

* Le délai calculé concerne le délai point à point entre émetteur et destinataire final du paquet

Le trafic de *situation awareness* est écoulé de manière satisfaisante (98% de paquets correctement reçus) et avec des délais corrects pour cette application (moins de 130ms en moyenne).

Par contre, la simulation permet de montrer des insuffisances dans le comportement du réseau concernant les flux de voix sur IP. En effet, ces derniers ne sont reçus correctement que dans une trop faible proportion (moins de 60%) et les contraintes de latence ne sont pas tenues (200 ms de latence moyenne pour la réception).

Dans les faits, si les données moyennes, bien qu'insuffisantes, ne sont pas catastrophiques, une analyse détaillée montre une grande disparité entre différents flux ainsi que le montre la Figure 5.19.

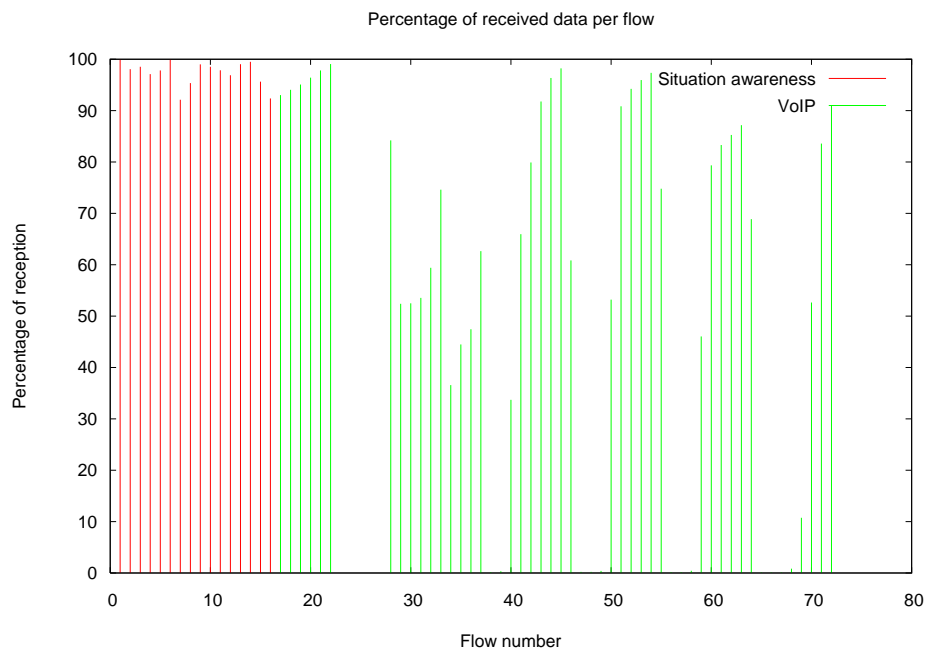


Figure 5.19 : Pourcentage de paquets reçus pour les différents flux

L'examen de l'évolution de la file MAC de données pour les différents nœuds, illustrée par la Figure 5.20, montre que, dès le début des communications de voix sur IP, le nœud 2 est saturé car il génère ou relaie une quantité trop importante de trafic.

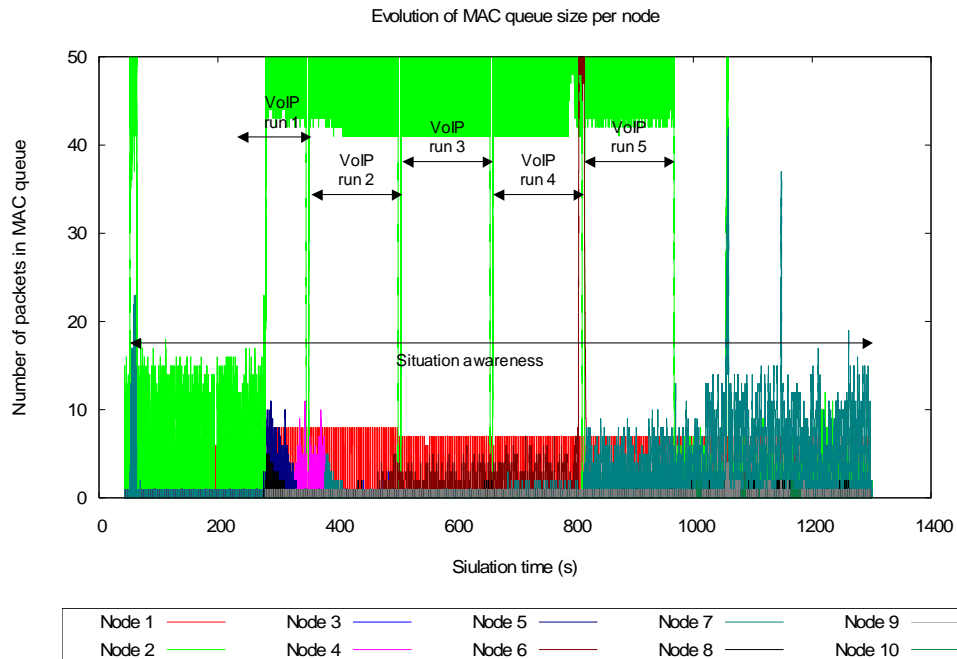


Figure 5.20 : Évolution de la taille des files MAC dans le scénario opérationnel

L'une des principales interrogations vient du fait que des situations similaires ont déjà été rencontrées dans les scénarios précédents. Dans le scénario 4 (cluster de 7 stations), le nœud 1, pourtant au centre du cluster, parvenait à échanger correctement du trafic VoIP avec les 6 autres nœuds du réseau. De même, dans le scénario 5 (fusion de réseaux), le nœud 1 (cette fois en coin de grille) communiquait sans trop d'encombre avec 5 autres nœuds de son groupe. Dans les deux cas, le débit total demandé pour la voix sur IP ne dépassait pas 50 kbits/s.

La réalisation de ce scénario permet de mettre à jour une des limitations majeures de la méthode de contrôle d'accès au canal MACA avec mécanisme RTS/CTS. En effet, un calcul d'overhead utilisant l'équation 3.3 page 45 avec les paramètres d'une transmission voix sur IP (datagrammes de 32 octets émis toutes les 30 ms) et les durées inter-trames calculées en 4.2.4.2.2, donne le résultat suivant : **le rendement avec mécanisme RTS/CTS pour la voix sur IP est inférieur à 20%.**

Le débit théorique du canal étant de 547 kbits/s, un nœud ne peut donc écouler au maximum que 100 à 110 kbits/s de débit usager pour les conversations de voix sur IP. Dans les scénarios précédents, la charge était telle que cette limite n'était pas atteinte. Dans le cas présent, le nombre de conversations simultanées (et donc le débit total) étant plus important, le débit utile n'est plus suffisant pour permettre au nœud 2 d'écouler son trafic.

Aussi, les idées suivantes pourraient permettre de corriger dans une certaine mesure les insuffisances du réseau :

- **Introduction du multicast** : les flux de VoIP utilisent N connexions point à point pour simuler une connexion point-multipoint. L'introduction du multicast pourrait donc diminuer d'un facteur N le débit sortant du nœud 2 ;
- **Agrégation de paquets** : permettrait d'éviter l'envoi répété de petits paquets et d'ainsi limiter l'overhead, au prix d'une latence accrue (temps d'attente pour agréger plusieurs paquets) ;

- **Priorité des flux** : l'ajout d'une notion de priorité des flux et l'introduction de fenêtres sans contention (*contention free bursting* de 802.11e) pour permettre de mieux écouler les flux prioritaires (au prix cependant d'un éventuel écrasement des flux d'arrière-plan) ;
- **Routage en fonction de la charge** : introduire la notion de capacité d'un nœud/d'un lien permettrait de contourner un nœud chargé afin de le soulager et de lui permettre d'écouler mieux son trafic.

Le débit TCP observé atteint 53,35 Kbits/s, ce qui est bien en-dessous des valeurs observées pour un trafic à 2 ou 3 bonds dans la Figure 5.11. Rappelons que ces mesures ont été obtenues en l'absence de tout autre trafic utilisateur.

Dans les faits, TCP cède naturellement de la bande passante aux différents trafics UDP. Ici, ce phénomène est accentué par la charge du nœud 2. **Il est donc nécessaire d'introduire un routage fonction de la capacité afin de répartir, autant que faire se peut, le trafic entre les différents nœuds du réseau** (et trouver par exemple une route plus longue, mais moins chargée et donc assurant un meilleur débit, pour le trafic TCP).

Enfin, le fonctionnement d'OLSR est satisfaisant. Le réseau ne comptant qu'un nombre faible de nœuds et le maillage étant limité, l'overhead du protocole de routage est limité à 4,83 Kbits/s. D'autre part, l'image du voisinage donnée par OLSR est assez proche de la réalité ainsi que le montre la Figure 5.21.

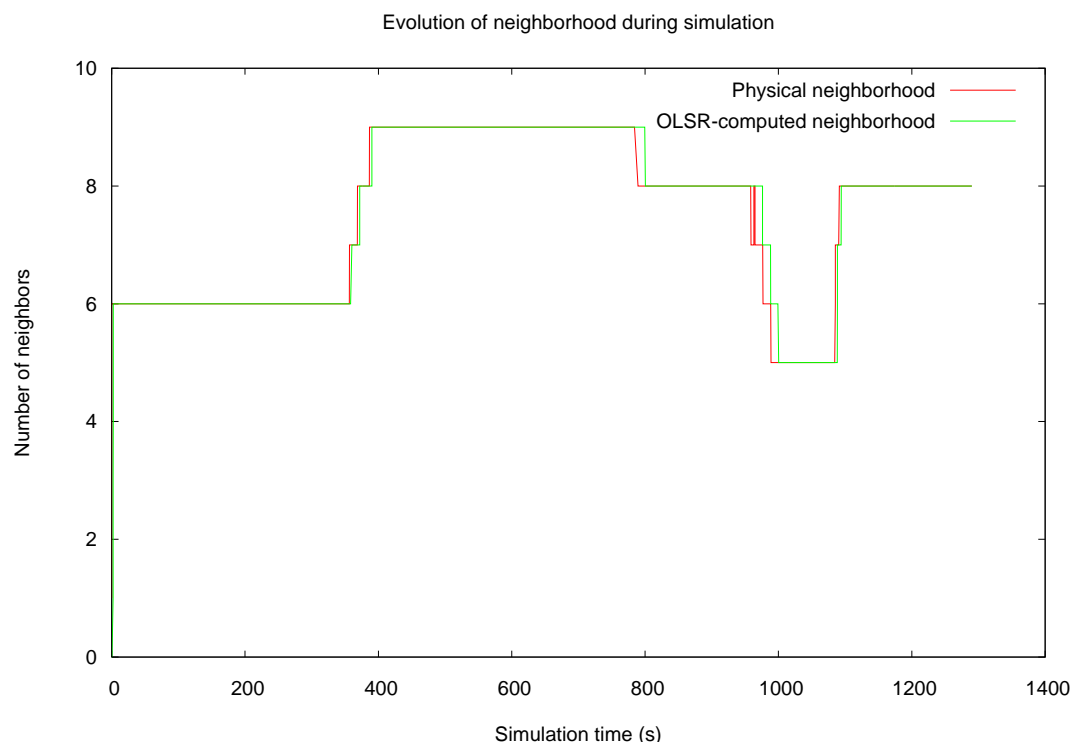


Figure 5.21 : Évolution du voisinage du nœud 2 pendant le scénario opérationnel

Le même scénario a été simulé en retirant la résistance aux collisions. Les résultats obtenus pour les différentes métriques observées sont sensiblement égales à celles présentées ici.

5.7 SYNTHÈSE DES RESULTATS

L'ensemble de l'étude menée précédemment permet de dresser un tableau des forces et des faiblesses de la solution basée sur CSMA/CA et OLSR, et de proposer des améliorations permettant de faire progresser ses performances.

Tableau 5.10 : Conclusions sur les performances de la forme d'onde

| Donnée | Forme d'onde Wi-Fi/OLSR | Améliorations pour la f.o. Wi-Fi/OLSR |
|-----------------------------|--|--|
| Accès au canal (couche MAC) | | |
| Type | CSMA/CA accès en contention | Routage en fonction de la charge <i>Contention Free Bursting</i> (cf. 802.11e paragraphe 3.4.1) Agrégation de paquets |
| Avantages | Répartition statistiquement équitable Pas d'allocation pour les liens inusités | |
| Inconvénients | « Famine » de canal pour les nœuds exposés Overhead important pour les petits paquets | |
| Routage | | |
| Type | Proactif (OLSR) | Priorité des flux de routage Améliorations d'OLSR (<i>fish-eye, cross-layering</i>) Gestion du multicast |
| Avantages | Bonne gestion de la mobilité Faible délai à l'établissement des routes | |
| Inconvénients | Fonctionnement perturbé en réseau chargé Fort overhead (dépend de la taille du réseau et de la topologie) | |
| Exigences de débit | | |
| Avantages | Bonnes performances TCP Fiabilité à faible charge | Extension CSMA/CA multicanal Routage QoS |
| Inconvénients | Écroulement des performances en réseau chargé Interactions entre flux | |
| Exigences de latence | | |
| Avantages | Très faible latence à faible charge | Création de circuits avec <i>fast forwarding</i> pour les flux contraints en latence |
| Inconvénients | Forte gigue Pas de garantie de délai | |

Les résultats présentés dans le Tableau 5.10 ci-dessus montre que la forme d'onde étudiée présente un profil de performance assez inégal. Pour résumer son comportement, il est possible de dire que :

- **La forme d'onde Wi-Fi/OLSR se montre relativement performante quand la charge du réseau est faible** et que la topologie du réseau (maillage notamment) n'est pas contraignante au point soit d'annuler les optimisations d'OLSR (ce qui charge le réseau par les seules données de routage) soit d'exposer certains nœuds aux communications de nombreux voisins (ce qui les met alors dans une situation de « famine » du canal) ;

- **Cette même forme d'onde n'est pas en mesure d'assurer une livraison correcte des paquets avec contraintes de latence.** Ce phénomène s'explique notamment par le fait que les durées inter-trames sont très longues (10 fois plus que dans un Wi-Fi « domestique » du fait de la portée accrue des nœuds) ce qui allonge la latence à l'envoi lors de l'utilisation du mécanisme de *binary exponential backoff*. De plus, l'accès en contention CSMA/CA ne garantit que statistiquement l'accès équitable au canal, ce qui entraîne dans les faits une gigue non-négligeable.

Les communications tactiques demandant une fiabilité quelles que soient les contraintes d'utilisation du réseau et la topologie adoptée, l'utilisation d'une forme d'onde avec gestion d'accès en contention (type Wi-Fi/OLSR) dans un contexte opérationnel semble difficile sans l'introduction d'une ou plusieurs des améliorations proposées dans le Tableau 5.10, et ce même si ce type d'accès permet de bonnes performances « de pointe » dans certaines situations favorables.

Toutefois, l'ensemble des mesures et analyses menées restent très utiles à la fois pour une meilleure compréhension des mécanismes inhérents au fonctionnement des réseaux ad-hoc avec accès en contention, mais aussi pour avoir une série de points de comparaison fiables et chiffrés sur différents scénarios types.

De plus, les résultats obtenus sont destinés à offrir un référentiel permettant de jauger de la pertinence des évolutions proposées sur cette même pile protocolaire ou sur tout autre pile protocolaire accessible.

6. BILAN

L'ensemble de ce stage a été pour moi l'occasion à la fois de développer mes connaissances dans le domaine des réseaux sans-fil, et plus particulièrement l'utilisation de ces derniers dans un contexte ad-hoc (nœuds mobiles, utilisations de multiples bonds radio).

La première phase de mon étude m'a permis de parfaire ma connaissance théorique du domaine (fonctionnement détaillé de CSMA/CA, du protocole de routage OLSR, introduction de la qualité de service sur Wi-Fi avec 802.11e, interactions fortes du mode ad-hoc avec TCP). Par la suite, en réalisant diverses simulations, j'ai pu confronter ces connaissances avec des résultats pratiques.

Au vu des résultats obtenus et des constatations effectuées, force est de constater que, dans la littérature (qu'elle concerne la description de standards ou de protocoles ou bien des papiers universitaires traitant de telle ou telle amélioration) de nombreux problèmes, notamment liés à l'adaptations des protocoles à des contraintes particulières, ne sont pas abordés. Ces cas de figure (comme l'overhead engendré par OLSR sur une topologie de grille ou l'inefficacité de la signalisation de CSMA/CA avec mécanisme RTS/CTS dans certaines conditions de trafic et de portée radio) ne sont révélés que par l'expérience.

Il a aussi été intéressant pour moi de construire un environnement de simulation, avec l'ensemble des simplifications nécessaires permettant de modéliser le comportement du réseau sur ordinateur. L'exercice est souvent complexe, surtout qu'il est nécessaire de judicieusement choisir les caractéristiques de la modélisation (afin d'avoir des résultats ni trop optimistes, ni trop pessimistes) et de savoir faire le lien entre les résultats obtenus et les hypothèses faites *a priori*.

D'un point de vue de l'intégration de l'entreprise, si je regrette, à cause de restrictions liées à la sécurité et la confidentialité des travaux, de n'avoir pas été plus directement intégré aux activités du service, l'étude que j'ai pu mener (qui s'inscrit donc en parallèle des développements de la forme d'onde Thales) ainsi que les sujets périphériques sur lesquels j'ai travaillé m'ont tout de même permis de bien appréhender l'ensemble des projets en cours au sein du service, leur portée technique et leurs enjeux.

De plus, de par les relations entretenues avec divers membres du service (chef de laboratoire, responsable chargé d'affaire, responsable de l'intégration système, ...), j'ai pu découvrir le mode de fonctionnement interne du service et les méthodologies de gestion de projet au sein de Thales Communications qui sont très développées et formalisées au sein d'un vaste référentiel.

En conclusion, l'expérience acquise pendant ce stage, à la fois dans le domaine des réseaux et dans celui de l'organisation et de la conduite de vastes projets au sein d'un grand groupe me permettra d'aborder mieux encore la suite de mon cursus scolaire et, à terme, mon entrée dans la vie professionnelle.

ANNEXE A : ADRESSAGE DANS L'ENVIRONNEMENT DE SIMULATION

L'incorporation de différents modules de différentes provenances (*mobility framework*, *INET framework*, simulations précédentes) a conduit à la coexistence de différents modes d'adressage au sein de la simulation. Les différents types d'adressage et leur zone de validité sont décrits dans la Figure A.1. Les différents modes qui coexistent sont les suivants :

1) L'adressage du générateur de trafic

Le générateur de trafic est issu d'une simulation précédente où les nœuds étaient uniquement identifiés par leur adresse MAC (en fait un simple entier), qui leur était affectée pendant la lecture du fichier de topologie.

Cette méthode a été conservée, et lors de création de chacun des nœuds (lors de la lecture de la partie initialisation du fichier de topologie), la **variable *address* est initialisée à la valeur du paramètre @MAC du fichier lu.**

Pour plus de simplicité, **l'adresse réseau de chacun des nœuds a été choisie comme égale à la valeur de l'adresse MAC.** Il ne s'agit donc pas d'une réelle adresse IP, mais d'un simple entier.

La valeur de l'adresse de broadcast est « -1 ».

2) L'adressage de la table de routage

La table de routage est issu du *INET framework*, qui définit des adresses IP complètes (A.B.C.D.) au moyen de la classe *IPAddress*. Aussi, pour être stockée dans la table de routage, l'adresse réseau de chacun des nœuds (au sens du générateur de trafic) doit être convertie (par une transformation bijective) en adresse IP (au sens du *INET framework*).

La conversion est effectuée en utilisant le paramètre *addressPrefix* de la simulation : il s'agit des 24 premiers bits de l'adresse IP (sous la forme « A.B.C ») qui sont ajoutés ou retranchés à l'adresse réseau du nœud suivant le sens de la conversion.

Exemple : *addressPrefix* = « 193.48.225 », *host*.address* = 174
 adresse IP de la classe *IPAddress* résultante = 193.48.225.174

Le seul module accédant à la table de routage (et implémentant ces opérations de conversion) est le module *RoutingNetwLayer*.

3) L'adressage du *mobility framework*

Avant toute chose, il est important de noter que dans la simulation **l'adressage du *mobility framework* n'est jamais utilisé pour le routage et la transmission des données.** Il est uniquement utilisé pour la mobilité des nœuds et la mise à jour des informations de connexion et agit de manière transparente pour l'utilisateur.

Dans ce mode, l'adresse MAC est l'*id* du module *Nic*, et l'adresse réseau l'*id* du module *NetwLayer*. Aussi, **la conversion (bijective) avec l'adressage du générateur de trafic peut être effectuée en remontant au module parent du module *Nic* ou *NetwLayer* et à sa variable *address* qui désigne l'adresse MAC et réseau au sens du générateur de trafic.**

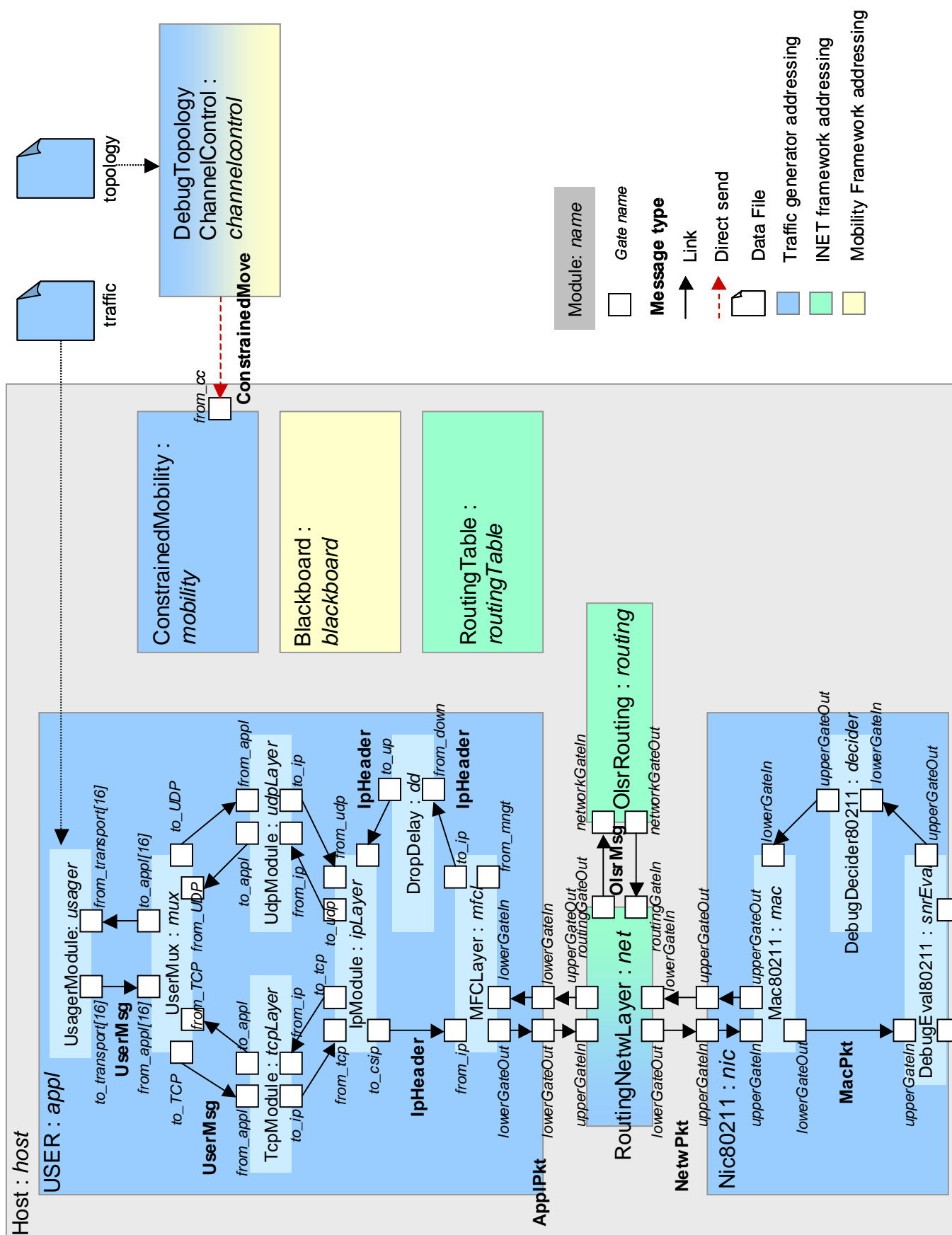


Figure A.1 : Zones d'adressage dans l'environnement de simulation

ANNEXE B : PARAMETRES DE SIMULATION

Le Tableau B.1 ci-dessous reprend l'ensemble des paramètres des différentes couches protocolaires utilisées dans la simulations et détaille leurs signification ainsi que les valeurs retenues par défaut.

Tableau B.1 : Paramètres de simulation

| Variable | Type | Valeur par défaut | Signification |
|--|----------------|-----------------------------|---|
| <i>Paramètres globaux</i> | | | |
| <i>addressPrefix</i> | Chaîne de car. | "192.168.0" | Préfixe pour la translation des adresses du <i>INET Framework</i> |
| <i>carrierFrequency</i> | Réel | 2.4E+9 | Fréquence de la porteuse |
| <i>bitrate</i> | Réel | 547E+3 | Débit de transmission des données |
| <i>ccModule</i> | Chaîne de car. | DebugTopologyChannelControl | Module utilisé pour le channel control |
| <i>snrModule</i> | Chaîne de car. | DebugEval80211 | Module utilisé pour l'évaluation de la recevabilité des paquets |
| <i>dcrModule</i> | Chaîne de car. | DebugDecider80211 | Module utilise pour l'ajout de bits d'erreur |
| <i>ChannelControl (sim.channelcontrol)</i> | | | |
| <i>IF_range</i> | Réel | 10000.0 | Rayon (en m) de la zone d'interférence |
| <i>TX_range</i> | Réel | 5000.0 | Rayon (en m) de la zone de transmission |
| <i>*pMax</i> | Réel | 2.0 | Puissance maximale d'émission des nœuds |
| <i>*sat</i> | Réel | -110 | Seuil d'atténuation des signaux |
| <i>*alpha</i> | Réel | 2.0 | Coefficient d'atténuation de la puissance |
| <i>Module de mobilité (sim.host*.mobility)</i> | | | |
| <i>updateInterval</i> | Réel | 0.1 | Intervalle (en sec.) de mise à jour de la position des nœuds |
| <i>Couche Applicative (sim.host*.appl)</i> | | | |
| <i>headerLength</i> | Entier décimal | 0 | Taille (en bits) de l'en-tête des paquets |

| <i>Couche TCP (sim.host*.appl.tcp)</i> | | | |
|--|----------------|-------|---|
| <i>TCPDelayedAck</i> | Booléen | False | Utilisation de l'acquittement retardé dans TCP |
| <i>TCPFastRetrans</i> | Booléen | false | Utilisation de la retransmission rapide après un paquet erroné ou dupliqué |
| <i>TCPFastRecovery</i> | Booléen | true | Utilisation de la retransmission rapide et du recouvrement rapide de la fenêtre de congestion après une congestion réseau |
| <i>TCPNewReno</i> | Booléen | false | Utilisation du modèle TCP dit « New Reno » |
| <i>mss</i> | Entier décimal | 536 | <i>Maximum segment size</i> (taille initiale de la fenêtre de congestion, en bits) |
| <i>rx_wnd</i> | Entier décimal | 65535 | Taille maximal de la fenêtre de congestion, en bits |
| <i>Couche UDP (sim.host*.appl.udp)</i> | | | |
| <i>connectionLessUdp</i> | Booléen | false | Mode special de UDP à ne pas utiliser dans cette simulation, valeur à ne pas modifier. |
| <i>Couche Réseau (sim.host*.netl)</i> | | | |
| <i>headerLength</i> | Entier décimal | 0 | Taille (en bits) de l'en-tête des paquets |
| <i>Couche Routage OLSR (sim.host*.routing)</i> | | | |
| <i>protocol</i> | Entier décimal | 100 | Identifiant du protocole OLSR |
| <i>startTime</i> | Réel | 0 | Date (en secondes) du lancement du protocole |
| <i>statStartTime</i> | Réel | 20 | Date (en secondes) du lancement des statistiques sur le protocole |
| <i>minEmissionTime</i> | Entier décimal | 2 | Intervalle (en sec.) minimal d'envoi de données de contrôle OLSR |
| <i>hnaAddr</i> | Chaîne de car. | "" | Adresse de liaison avec un réseau non-OLSR (non-usité dans la simulation) |
| <i>hnaMask</i> | Chaîne de car. | "" | Masque des adresses du réseau non-OLSR lié (non-usité dans la simulation) |
| <i>mcAddrSnd</i> | Chaîne de car. | "" | (non-usité dans la simulation) |

| <i>Table de routage (sim.host*.routingTable)</i> | | | |
|--|----------------|----------|--|
| <i>IPForward</i> | Booléen | false | Activer le routage de paquets IP |
| <i>routingFile</i> | Chaîne de car. | "" | Fichier contenant la table de routage initiale |
| <i>Couche MAC (sim.host*.nic.mac)</i> | | | |
| <i>headerLength</i> | Entier décimal | 272 | Taille (en bits) de l'en-tête des paquets |
| <i>maxQueueSize</i> | Entier décimal | 14 | Taille maximale de la file d'attente de la couche MAC |
| <i>prioritizeRouting</i> | Booléen | True | Ajoute une file MAC prioritaire pour les trames de routage |
| <i>ST</i> | Réel | 1.552E-4 | Durée d'un slot de la couche MAC |
| <i>SIFS</i> | Réel | 4.7E-5 | Durée de l'espace court entre deux trames |
| <i>rtsCts</i> | Booléen | True | Utilisation du mode RTS/CTS (MACA) |
| <i>broadcastBackoff</i> | Entier décimal | 31 | Fenêtre de backoff pour les paquets de broadcast |
| <i>Couche Evaluation (sim.host*.nic.snrEval)</i> | | | |
| <i>*headerLength</i> | Entier décimal | 192 | Taille (en bits) de l'en-tête des paquets |
| <i>bitrate</i> | Réel | 547E+3 | Débit de la couche évaluation |
| <i>*transmitterPower</i> | Réel | 2.0 | Puissance d'émission du nœud |
| <i>*snrThresholdLevel</i> | Réel | 3 | Seuil du rapport signal à bruit pour considérer le paquet comme non-bruité |
| <i>*carrierFrequency</i> | Réel | 2.4E+9 | Fréquence de la porteuse |
| <i>*thermalNoise</i> | Réel | -110 | Niveau de bruit thermique sur le canal |
| <i>*sensitivity</i> | Réel | -85 | Sensibilité au bruit du nœud |
| <i>*pathLossAlpha</i> | Réel | 2.0 | Coefficient d'atténuation de puissance du nœud |
| <i>Couche Décideur (sim.host*.nic.decider)</i> | | | |
| <i>*snrThreshold</i> | Réel | 4 | Seuil du rapport signal à bruit pour considérer le paquet comme étant sans collision |
| <i>syncBits</i> | Entier décimal | 80 | Nombre de bits de synchronisation |
| <i>maxInterfRatio</i> | Réel | 0.5 | Proportion maximale de collision par rapport à la taille du message |

* Paramètres inusités lors de l'utilisation du modèle radio « debug » (décrit en 4.2.5)

ANNEXE C : TAILLE ET DUREE D'ÉMISSION DES MESSAGES

Le Tableau C.1 ci-dessous reprend la taille et la durée d'émission à 547 kbits/s des différents messages de données et de contrôle utilisés dans la simulation :

Tableau C.1 : Taille et durée d'émission des messages

| Message | Taille | Durée | Proportion (PDU standard ^{**}) | Proportion (PDU max. ^{***}) |
|------------|----------------|----------------|---|--|
| Données | <= 12160 bits* | <= 33,8 ms | - | - |
| RTS | 160 bits | 0,3 ms | 16 % | 1,31% |
| CTS | 112 bits | 0,2 ms | 11,2 % | 0,92% |
| ACK | 112 bits | 0,2 ms | 11,2 % | 0,92% |
| OLSR Hello | >= 192 bits | 0,35 à 33,8 ms | >= 19,2% | >= 1,58% |
| OLSR TC | >= 160 bits | 0,3 à 33,8 ms | >= 16% | >= 1,31% |

* Taille maximale d'un paquet IP (1500 octets de données et 20 octets d'en-tête IP)

** Calculée par rapport à une trame de données de 1000 bits (taille caractéristique des paquets utilisés dans les scénarios de simulation).

*** Calculé par rapport à la taille maximale d'un paquet IP (soit 1520 octets, ou 12160 bits).

BIBLIOGRAPHIE

- [1] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot. *Optimized Link State Routing Protocol for Ad-hoc Networks*.
- [2] T. Clausen, P. Jacquet. *RFC 3636, Optimized link-state routing protocol*.
- [3] A. Laouti, P. Mühlethaler A. Najid, E. Plakoo. *Simulation Results of the OLSR Routing Protocol for Wireless Network*.
- [4] T. Clausen, G. Hansen, L. Christensen, G. Behrmann. *The Optimized Links State Routing Protocol Evaluation through Experiments and Simulation*.
- [5] P. Kuosmanen. *Classification of Ad-hoc Routing Protocols*.
- [6] C. Siva Ram Murthy, B. S. Manoj. *Ad-hoc Wireless Networks, Architectures and Protocols*. Prentice Hall.
- [7] M. Benzaid, P. Minet, K. Al Agha. *Integrating fast mobility in the OLSR routing protocol*.
- [8] M. Benzaid, P. Minet, K. Al Agha. *Analysis and simulation of Fast-OLSR*.
- [9] J. Schiller. *Mobile Communications, Second Edition*.
- [10] M. Gast. *802.11 Réseaux sans fil, La référence*.
- [11] S. Basagni, M. Conti, S. Giordano, I. Stojmenovic. *Mobile Ad-hoc Networking*.
- [12] S. Xu, T. Saadawi. *Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad-hoc Networks?*
- [13] C. T. Calafate, P. Manzoni, M. P. Malumbres. *Assessing the effectiveness of IEEE 802.11^e in multi-hop mobile network environments*.
- [14] J. del Prado Pavón, S. Shankar N. *Impact of Frame Size, Number of Stations and Mobility on the Throughput Performance of IEEE 802.11e*.
- [15] A. Leon-Garcia, I. Widjaja. *Communication Networks : Fundamental Concepts and Key Architectures, Second Edition*.
- [16] A. Kanzaki, T. Uemukai, T. Hara, S. Nishio. *Dynamic TDMA Slot Assignment in Ad-hoc Networks*.
- [17] C.D. Young. *USAP : a unifying dynamic distributed multichannel TDMA slot assignment protocol*.
- [18] C.D. Young. *USAP multiple access : dynamic resource allocation for mobile multihop multichannel wireless networking*
- [19] A. Varga. *OMNeT++ Manual*. March 2005.
- [20] M. Löbbers, D. Willkomm. *A Mobility Framework for OMNeT++, User Manual*.
- [21] Xiaoyan Hong, Kaixin Xu and Mario Gerla. *Scalable Routing Protocols for Mobile Ad Hoc Networks*. July 2002.
- [22] ANSI/IEEE Standard 802.11, 1999 Edition (R2003). June 2003.