Contributions à l'amélioration de la qualité de service dans les réseaux sans fil Wifi

Ahmed-Riadh Rebai

Université de Valenciennes et du Hainaut-Cambrésis

Numéro d'ordre : 09/37

Contributions à l'amélioration de la qualité de service dans les réseaux sans fil WiFi

THÈSE

Présentée et soutenue publiquement le 11 décembre 2009 pour l'obtention du

Doctorat de l'université de Valenciennes et du Hainaut-Cambrésis

Spécialité Automatique et Informatique des Systèmes Industriels et Humains

Discipline : Informatique

Par

Ahmed Riadh REBAI

Rapporteurs : M. Sami Tabbane, Professeur, Ecole Supérieure des Communication de Tunis
M. Toufik Ahmed, Professeur, Université Bordeaux I

Examinateurs : M. Dominique Dallet, Professeur, Université de Bordeaux I
M. Mazen Saghir, Professeur Associé, Université de Texas A&M au Qatar

Co-encadrant : M. Christophe Wilbaut, Maître de conférences, Université de Valenciennes

Directeur : M. Saïd Hanafi, Professeur, Université de Valenciennes

Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines – UMR 8530

A ma mère Afifa qui surclasse les mères de la terre entière,

A mon père Mohamed dont les encouragements m'ont permis d'avancer,

A ma femme Mariem dont la douceur et la patience m'ont permis d'évoluer,

A mon frère Chiheb qui m'a entouré avec son affection illimitée et à qui je dois tout,

A mon fils Youssef qui m'a apporté tant de bonheur dans ma vie et devenu ma source d'inspiration,

A tous ceux qui m'ont accompagné dans le travail que j'ai accompli,

A tous ceux que j'aime et qui m'aiment,

Qu'ils trouvent dans ce travail le résultat de leurs conseils et encouragements.

« Les hommes se plaisent à penser qu'ils peuvent se débrouiller seuls, mais l'homme, le vrai, sait que rien ne vaut le soutien et les encouragements d'une bonne équipe ».

Tim Allen, Acteur (1953-)

« Dans les sciences, le chemin est plus important que le but. Les sciences n'ont pas de fin ».

Erwin Chargaff, Biochimistre (1905-2002)

Avant-propos

Le travail que nous présentons dans cette thèse a été effectué dans le cadre de la préparation d'un diplôme de Doctorat en Informatique, Spécialité Automatique et Informatique des Systèmes Industriels et Humains de l'Université de Valenciennes et du Hainaut-Cambrésis (UVHC). Les travaux de recherche réalisés ont été menés au sein de l'équipe de recherche "Systèmes d'information, d'aide à la décision et embarqués" (SIADE) du Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaine (LAMIH). Je tiens à exprimer ma profonde reconnaissance et ma gratitude envers Monsieur Smail Niar, Professeur à l'Université de Valenciennes et Directeur du groupe de recherche SIADE de m'avoir accueilli dans l'équipe.

Au terme de ce travail de thèse, je tiens à remercier vivement :

Monsieur Sami Tabbane, Professeur à l'Ecole Supérieure des Communications de Tunis (SUP'COM) et Directeur de l'Unité de Recherche en Réseaux Radio Mobile Multimédia (MEDIATRON), d'avoir accepté de rapporter ce travail, pour l'intérêt qu'il a accordé à discuter certains de nos résultats et pour ses remarques pertinentes. Je le remercie pour l'honneur qu'il nous fait en participant au jury de cette thèse.

Monsieur Toufik Ahmed, Professeur à l'Ecole Nationale Supérieure d'Electronique, Informatique, Télécommunications, Mathématique et Mécanique de Bordeaux (ENSEIRB-MATMECA) de l'Institut Polytechnique de Bordeaux (IPB), pour son acceptation de la lourde tâche de rapporteur, pour avoir apporté sa caution scientifique en jugeant ce travail et pour son honorable participation au jury de cette thèse.

Monsieur Dominique Dallet, Professeur à l'Ecole Nationale Supérieure d'Electronique, Informatique, Télécommunications, Mathématique et Mécanique de Bordeaux (ENSEIRB-MATMECA) de l'Institut Polytechnique de Bordeaux (IPB), pour l'intérêt qu'il porte à notre travail en acceptant d'examiner avec sa rigueur scientifique habituelle nos contributions de recherche.

Monsieur Mazen Saghir, Professeur Associé à l'Université de Texas A&M au Qatar (TAMUQ), pour avoir fait l'effort du voyage à Valenciennes et accepté de participer au jury de cette thèse en contribuant à la discussion de nos travaux avec sa grande compétence reconnue dans le domaine.

Monsieur Saïd Hanafi, Professeur à l'Université de Valenciennes, pour avoir accepté de diriger ma thèse. Je le remercie sincèrement pour son suivi attentif, ses conseils et nos discussions enrichissantes. Je lui suis reconnaissant pour tous les moyens qu'il a bien voulu mettre à ma disposition pour mener à bien dans d'excellentes conditions mes travaux de recherche au sein du groupe de recherche SIADE du laboratoire LAMIH de l'UVHC. Je le remercie aussi pour sa forte implication dans l'encadrement des travaux de ma thèse qui m'a permis de m'affranchir de multiples complexités. Qu'il trouve ici l'expression de mes respectueux remerciements et ma grande admiration pour son soutien et la gentillesse qu'il m'a toujours témoignée.

Monsieur Christophe Wilbaut, Maître de conférences à l'Université de Valenciennes, pour le coencadrement de ma thèse. Je lui présente ma gratitude pour ses conseils scientifiques et pour l'agréable cadre de travail.

Je remercie tous les enseignants chercheurs de l'Université de Texas A&M au Qatar et en particulier Monsieur Hussein Alnuweiri Professeur et Coordinateur du Département Electrique et Informatique du TAMUQ qui m'a aidé au bon déroulement de mes activités pédagogiques et de mes activités de recherche en m'offrant tout le support administratif nécessaire.

Il m'est indispensable de remercier autant Monsieur Chiheb Rebai, Maître de conférences à SUP'COM, pour sa présence et son assistance continue.

Je tiens à remercier aussi mes collègues au TAMUQ ainsi que le personnel administratif pour m'avoir soutenu et encouragé pour finaliser mes travaux de recherche.

Je ne pourrais jamais oublier d'exprimer ma gratitude à toute ma famille pour sa patience, son assistance et son sacrifice tout au long de mes études. En outre, je profite de cette occasion pour remercier tous mes amis, pour leur encouragement et leur présence, en particulier Amin, Wiem, Manel, Anis, Layla, Lassad et remercier toute personne ayant contribué de près ou de loin à la réussite de ce travail de thèse.

Résumé

Le standard IEEE 802.11 est devenu la référence d'excellence pour les réseaux locaux sans fil par son très large déploiement au niveau mondial. L'objectif des travaux de cette thèse est d'élaborer de nouveaux algorithmes pour contribuer à l'amélioration du schéma global de la Qualité de Service (*QoS*) dans les réseaux WiFi. Nous avons proposé un nouvel algorithme appelé *Modified Adaptive Auto Rate Fallback* qui permet d'obtenir des performances très intéressantes en termes de débit général de sortie et du nombre de paquets erronés tout en restant compatible avec le standard IEEE 802.11. Les expérimentations menées avec cette nouvelle technique de contrôle de débit confirment le gain en performance obtenu par des simulations sous la plateforme *Network Simulator* 2. Nous avons également amélioré la méthode d'accès de la norme 802.11e *« Enhanced Distributed Channel Access »*, en introduisant une nouvelle classification entre les stations mobiles. Des simulations sous la plateforme *Network Simulator* 2 ont prouvé une réduction considérable du taux de collisions dû à l'accès simultané par les stations mobiles et une amélioration du débit de transmission respectant ainsi les contraintes de la qualité de service liées aux applications multimédias. La dernière contribution porte sur le mécanisme de commutation entre les points d'accès (*Handover*) déployé dans les réseaux WiFi. Plus précisément, nous avons élaboré un nouveau mécanisme de *Handover* peu complexe et efficace appelé *Prevent-Scan Handoff Procedure*. Il permet de sélectionner un meilleur point d'accès candidat pour les prochaines transmissions d'une station mobile en mode infrastructure tout en réduisant à la fois le temps et le trafic engendré. Enfin, un nouveau simulateur a été développé afin d'analyser et de comparer les performances de la technique proposée avec des algorithmes de la littérature. Les résultats des simulations montrent une diminution de la latence et de la gigue par rapport aux approches récentes, et de plus notre méthode respecte la transmission des trames à *QoS* exigée.

Abstract

Due to their wide proliferation, *IEEE 802.11 Wireless Local Area Networks* (*WLANs*) are expected to support multimedia applications such as voice and broadband video transmissions, which normally have a strict bounded transmission delay. The aim of this thesis is to enhance the support of these applications, and so, to improve the *Quality of Service* (*QoS*) scheme in *IEEE 802.11 WLANs*. Firstly, we focus our work on the link adaptation procedure performed over WiFi networks. A new dynamic time-based link adaptation mechanism, called *Modified Adaptive Auto Rate Fallback*, is proposed. Simulations are performed by means of the *Network Simulator 2* and the results show the quality improvement of transmission link. The results also demonstrate that the proposed mechanism outperforms the basic solution in terms of providing support to both acknowledgment-based and time-based rate control decision. In the second part of this thesis, we introduce a new inter-node priority access scheme with the existing *Enhanced Distributed Chanel Access* of *IEEE 802.11e* networks. The new technique is based on the junction of a new inter-node priority with the existing inter-frame priority. The simulation results show that the proposed technique improves the basic *802.11e MAC* protocol in terms of providing support to both strict priority and weighted fair service. Compared to other solutions, the new model is easier to implement in real systems, has better aggregate throughput and is more stable. As a third part of this work, a novel handoff scheme for *IEEE 802.11 WLANs* is proposed to reduce handoff latency and to improve the *QoS* support in multimedia applications. The proposed handoff scheme, called *Prevent-Scan Handoff Procedure*, uses an early pre-scan phase to avoid probe wait delays during next Handoff occurrence. Through extensive simulations, we prove that the new handoff procedure decreases handoff latency considerably and accelerates handoff by minimizing the time during which the MS remains out of contact with its AP. Therefore, the inter-frame delay incurred is within multimedia applications' delay restrictions. These simulations are carried out over a new *IEEE 802.11* handoff simulator to provide performance evaluations. We prove that the new handoff technique achieves both fast and smooth handoff which is requested by multimedia applications.

Acronymes et Sigles

AARF	Adaptive Auto Rate Fallback
AC	Access Category
ACK	Acknowledgment
ADSL	Asymetric Digital Subscriber Line
AES	Advanced Encryption Standard
AF	Assured Forwarding
AIFS	Arbitration Inter Frame Spacing
AIFSN	Arbitration Inter Frame Spacing Number
AODV	Ad Hoc On demand Distance Vector
ARF	Auto Rate Fallback
AP	Access Point
APFH	Adaptive Preemptive Fast Handoff
BE	Best Effort
BEB	Binary Exponential Back off
BER	Bit Error Rate
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
CBR	Constant Bit Rate
CBQ	Class-Based Queuing
CCA	Clear Channel Assessment
CHCF	Controlled Hybrid Coordination Function
CDMA	Code Division Multiple Access
CONSER	Collaborative Simulation for Education and Research
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
CW	Collision Windows
DARPA	Defence Advanced Research Projects Agency
DCF	Distributed Coordination Function
DIFS	DCF Inter Frame Spacing
DS	Distributed System
DSCP	Differentiated Service Code Point
DSSS	Direct Sequence Spread Spectrum
DVMRP	Distance Vector Multicast Routing Protocol
EDCA	Enhancement Distributed Channel Access
EDCF	Enhancement Distributed Coordination Function
EF	Expedited Forwarding
ESS	Extended Service Set
ESSID	Extended Service Set
FHAP	Fast Handoff by Avoiding Probe Wait
FHSS	Frequency Hopping Spread Spectrum
GPRS	General Packet Radio Services
GSM	Global System for Mobile Communication

HCCA	Hybrid Coordination Function Controlled Chanel Access
HCF	Hybrid Coordination Function
IAPP	Inter Access Point Protocol
IETF	Internet Engineering Task Force
IFS	Inter Frame Spacing
IR	Infra Red
LLC	Logical Link Control
MAARF	Modified Adaptive Auto Rate Fallback
MAC	Media Access Control
MIMO	Multiple Input Multiple Output
NAM	Network Ani Mator
NAV	Network Allocation Vector
NG	Neighbour Graph
NIC	Network Interface Card
NSF	National Sciences Foundation
NS-2	Network Simulator 2
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnection
PC	Point Coordination
PCF	Point Coordination Function
PE	Persistance Factor
PHB	Peer Hop Behaviour
PIFS	PCF Inter Frame Spacing
PDA	Personal Digital Assistant
PSM	Power Saving Mode
PSHP	Prevent Scan for 802.11 Handoff Procedure
QAM	Quadrature Amplitude Modulation
QAP	Quality of Service AP
QoS	Quality of Service
RADIUS	Remote Authentification Dial In User Service
RSSI	Received Signal Strength Indicator
RTS	Request To Send
RTT	Round Trip Time
RTO	Retransmission Time Out
RTP	Real Time Transport Protocol
SAMAN	Simulation Augmented by Measurement and Analysis for Networks
SIFS	Short Inter Frame Spacing
SNR	Signal-to-Noise Ratio
SRM	Scalable Reliable Multicast
SSID	Service Set Identifier
SyncScan	Synchronized Scan
TC	Traffic Category
TCL	Tool Command Language
TCP	Transmission Control Protocol
TFRC	TCP Friendly Rate Control Protocol
TID	Traffic Identifier
TOS	Type of Service

TS	**T**raffic **S**tatement
UDP	**U**ser **D**atagram **P**rotocol
UMTS	**U**niversal **M**obile **T**elecommunication **S**tandard
U-NII	**U**nlicensed-**N**ational **I**nformation **I**nfrastructure
UP	**U**ser **P**riority
VBR	**V**ariable **B**it **R**ate
VCH	**V**irtual **C**ollision **H**andler
WCDMA	**W**ideband **C**ode **D**ivision **M**ultiple **A**ccess
WEP	**W**ired **E**quivalent **P**rivacy
WiFi	**Wi**reless **Fi**delity
WiMAX	**W**orldwide **I**nteroperability for **M**icrowave **A**ccess
WLAN	**W**ireless **L**ocal **A**rea **N**etwork
WMAN	**W**ireless **M**etropolitan **A**rea **N**etwork
WPAN	**W**ireless **P**ersonal **A**rea **N**etwork
WPA	**WiFi** **P**rotected **A**ccess
WWAN	**W**ireless **W**ide **A**rea **N**etwork

Table des Matières

Avant-propos iii
Résumé v
Abstract vi
Acronymes et sigles vii
Table des matières x
Table des figures xiv
Liste des tableaux xvii

Introduction Générale 1

Chapitre 1 : La technologie WiFi

I. Les réseaux sans fil 5
II. Le standard IEEE 802.11 8
1. Les normes 802.11 9
2. Les modes de connexion supportés 11
A. Le mode Ad-hoc 12
B. Le mode Infrastructure 12
3. Le modèle OSI 13
A. La couche Physique 13
B. La couche Liaison de Données 15
4. Les méthodes d'accès au medium dans les réseaux 802.11 17
A. La méthode DCF : Distributed Coordination Function 17
B. La méthode PCF : Point Coordination Function 21
5. La sécurité dans la norme 802.11 22
III. Conclusion 24

Chapitre 2 : La qualité de service dans les réseaux WiFi

I. Contexte et objectifs 25
II. Adaptation du débit physique 25
1. Estimation de la qualité du canal et la sélection du débit 26
2. Synthèse sur le choix du débit approprié 27
III. Différenciation de services dans les réseaux 802.11 29
1. Couplage de DiffServ avec le schéma de QoS dans les WLANs 31
2. Améliorations du Distributed Coordination Function (DCF) 33

A. DCF : Critiques 33
B. Hybrid Coordination Function (HCF) 35
C. Enhanced Distributed Function (EDCF) 36
IV. Handover dans le 802.11 38
1. Phase de détection/déclenchement 39
2. Phase de Désauthentification 39
3. Phase de recherche d'un nouveau point d'accès 39
A. Scan Actif 39
B. Scan Passif 40
4. Phase d'Authentification 41
5. Phase d'Association ou Réassociation 41
6. Handover en pratique 42
V. Conclusions 43

Chapitre 3 : Mécanisme d'adaptation du débit dans les réseaux WiFi

I. Contexte et objectifs 44
II. Mécanismes actuels d'adaptation du débit dans le 802.11 44
1. Auto Rate Fallback (ARF) 45
2. Adaptive Auto Rate Fallback (AARF) 47
3. Discussion 49
III. Nouvelle technique de contrôle adaptatif 50
1. Round Trip Time (RTT) 50
2. Intégration du paramètre RTT 51
3. Principe de l'algorithme MAARF 53
4. Paramétrage de MAARF 55
IV. Résultats et Interprétations 57
1. Optimisation des paramètres 58
2. Régimes des Tests 59
A. État de canal non stable 59
B. État de canal stable 60
C. Régime d'une station mobile 61
V. Synthèse des résultats 62

Chapitre 4 : Evaluation du MAARF

I. Objectifs 64
II. Network Simulator - 2 64
1. Présentation du NS-2 65

2. Network AniMator (NAM) 66
3. Fichiers "Trace" et le logiciel XGraph 66
III. Simulations et Analyses 67
1. Environnement de simulation 67
2. Résultats et Interprétations 68
A. Discussion des paramètres de l'algorithme MAARF 68
B. Débit théorique du lien physique 69
C. Débit réel observé sur le canal 73
D. Débit moyen observé 78
E. Taux de perte des paquets 80
F. Résumé des simulations menées 81
IV. Conclusion et synthèse 81

Chapitre 5 : Révision du schéma d'accès EDCA adopté par 802.11e

I. Contexte et Objectifs 83
II. Enhancement Distributed Channel Access (EDCA) 84
1. Définition du EDCA 84
2. Limites de l'EDCA 86
III. Nouvelle approche de classification inter-stations 88
1. Critères de la nouvelle classification 89
2. Mécanisme de la nouvelle classification 90
3. Implémentation de la nouvelle classification 92
4. Priorité tolérante pour les trafics de classes inférieures 95
IV. Simulations et analyses des résultats 96
1. Les scénarii adoptés 96
2. Simulations et résultats obtenus 97
A. Première évaluation : l'EDCA révisé par rapport l'EDCA classique 97
B. Deuxième évaluation : choix des paramètres k et n de l'EDCA révisé 99
C. Troisième évaluation : priorité tolérante aux trafics non-multimédias 101
D. Quatrième évaluation : taux de collision enregistrés 102
3. Résumé des simulations réalisées 103
V. Conclusion et synthèse 103

Chapitre 6 : Etude du Handover 802.11

I. Objectifs 105
II. Estimation du temps du Handover 105
III. Etat de l'art 108

1. Solutions exploitant les paramètres de la norme 108
2. Solutions pour réduire le temps du Scan 109
A. Déploiement d'un réseau de capteur 110
B. Exécution d'un Scan Synchronisé 111
C. Graphe de voisinage 113
D. Handover rapide en évitant l'attente de probe 117
E. Handover préemptif rapide et adaptatif 118
3. Gestion du Handover Inter-AP 120
4. Optimisation du Handover au niveau IP 123
A. Architecture du Mobile IPv6 123
B. Protocole Mobile IPv6 124
IV. Conclusion 125

Chapitre 7 : Nouvelle Technique de Scan Préventif PSHP

I. Contexte et objectifs 126
II. Prevent Scan Handoff Procedure 126
1. Nouvelle procédure d'association 128
2. Nouvelle phase de Pré-Scan 130
3. Fonctionnement du nouveau mécanisme PSHP 131
III. Implémentations et résultats 136
1. Paramétrage de l'algorithme 137
2. Résultats et analyses 140
IV. Conclusion 147

Conclusion générale 149

Annexe A 154

Bibliographie 161

Table des figures

Chapitre 1 : La technologie WiFi

Figure 1.1 : Les standards des réseaux sans fil 6
Figure 1.2 : Connexion sans fil 9
Figure 1.3 : Mode *Ad-hoc* 12
Figure 1.4 : Mode *Infrastructure* 13
Figure 1.5 : Les couches 1 et 2 de la norme 802.11 13
Figure 1.6 : Récapitulatifs des technologies et des débits possibles 14
Figure 1.7 : La gestion d'accès dans la norme 802.11 16
Figure 1.8 : Le mécanisme *CSMA/CA* du 802.11 19
Figure 1.9 : Paquets *RTS/CTS* et mise à jour des *NAVs* 21
Figure 1.10 : Le mode Point Coordination Function « *PCF* » 22

Chapitre 2 : La qualité de service dans les réseaux WiFi

Figure 2.1 : Champ *QoS* dans la trame *MAC* 802.11 33
Figure 2.2 : Evolution du débit et du délai en mode *DCF* [31] 35
Figure 2.3 : Différence entre une station 802.11 classique et une station 802.11e 36
Figure 2.4 : Les phases du Handover dans 802.11 38
Figure 2.5 : Scan actif 40
Figure 2.6 : Scan passif 40

Chapitre 3 : Mécanisme d'adaptation du débit dans les réseaux WiFi

Figure 3.1 : Diagramme de transition de l'algorithme *ARF* 46
Figure 3.2 : Diagramme de transition de l'algorithme *AARF* 47
Figure 3.3 : Comparaison entre *AARF* et *ARF* 48
Figure 3.4 : Définition du *Round Trip Time* « *RTT* » 51
Figure 3.5 : Transmissions des trames de données 52
Figure 3.6 : Choix des paramètres de l'algorithme : RTT_i^+ et RTT_i^- 53
Figure 3.7 : Diagramme de transition du nouvel algorithme *MAARF* 57
Figure 3.8 : Adaptations de débit avec différents paramètres du *MAARF* 59
Figure 3.9 : Adaptations de débit en régime transitoire 59

Figure 3.10 : Adaptations de débit en absence d'erreurs 60
Figure 3.11 : Adaptations de débit aux conditions instantanées du canal 61
Figure 3.12 : Débit moyen en fonction du nombre de paquets transmis 62
Figure 3.13 : Nombre de trames erronées par rapport au nombre de paquets transmis 63

Chapitre 4 : Evaluation du MAARF

Figure 4.1 : Optimisation des paramètres *g* et *h* du *MAARF* 69
Figure 4.2 : Débit théorique en absence d'erreur 70
Figure 4.3 : Débit théorique pour une transmission avec un taux d'erreur = 1% 70
Figure 4.4 : Débit théorique pour une transmission avec un taux d'erreur = 3% 71
Figure 4.5 : Débit théorique pour une transmission avec un taux d'erreur = 5% 72
Figure 4.6 : Débit théorique pour une transmission avec un taux d'erreur = 7% 72
Figure 4.7 : Débit réel pour une transmission sans erreur 74
Figure 4.8 : Débit réel pour une transmission avec un taux d'erreur = 1% 74
Figure 4.9 : Débit réel pour une transmission avec un taux d'erreur = 3% 75
Figure 4.10 : Débit réel pour une transmission avec un taux d'erreur = 5% 75
Figure 4.11 : Débit réel pour une transmission avec un taux d'erreur = 7% 76
Figure 4.12 : Débit réel pour une transmission avec un taux d'erreur = 10% 76
Figure 4.13 : Débit moyen en fonction de la distance 77
Figure 4.14 : Débit moyen en fonction de la taille des paquets 78
Figure 4.15 : Débit moyen en fonction du taux d'erreur 79
Figure 4.16 : Taux de perte des paquets en fonction du taux d'erreur canal 80

Chapitre 5 : Révision du schéma d'accès EDCA adopté par 802.11e

Figure 5.1 : Structure de l'*EDCA* adopté par le protocole *MAC* 802.11e 85
Figure 5.2 : Relation entre les temps *Inter-Frame IFS* dans 802.11e 86
Figure 5.3 : Collisions entre stations par l'*EDCA* classique du 802.11e 88
Figure 5.4 : Les collisions à résoudre par la nouvelle classification 90
Figure 5.5 : Schéma d'accès avec la nouvelle priorité inter-station 91
Figure 5.6 : Choix de la priorité inter-station dans l'*EDCA* révisé 94
Figure 5.7 : Schémas d'accès au canal par les cinq stations examinées du réseau 802.11e 98
Figure 5.8 : Discussion des paramètres *k* et *n* du mécanisme proposé 100
Figure 5.9 : Taux de collision enregistrés pour plusieurs configurations par type de trafic 102

Figure 5.10 : Taux de collision général en fonction du nombre de paquets 104

Chapitre 6 : Etude du Handover 802.11

Figure 6.1 : Réseau de capteurs déployé dans le WiFi 110
Figure 6.2 : Handover par interrogation de capteurs 111
Figure 6.3 : Fonctionnement de l'algorithme *SyncScan* 112
Figure 6.4 : Architecture du graphe de voisinage 114
Figure 6.5 : Les paquets échangés par un client *NG* 115
Figure 6.6 : Diagramme d'état du client *NG* 115
Figure 6.7 : Phase de découverte d'un nouvel *AP* avec *FHAP* 118
Figure 6.8 : Les zones de couverture d'un *AP* 119
Figure 6.9 : Gestion de l'authentification des stations par le serveur *RADIUS* 121
Figure 6.10 : Diagramme d'échange *IAPP* entre les *APs* 122
Figure 6.11 : Architecture du *Mobile IP* 124

Chapitre 7 : Nouvelle Technique de Scan Préventif PSHP

Figure 7.1 : Organigramme d'Association avec le meilleur *AP* 129
Figure 7.2 : Le nouveau mécanisme *PSHP* 131
Figure 7.3 : Diagramme d'état pour un client mobile exécutant *PSHP* 133
Figure 7.4 : Variation du *RSSI* d'un client en fonction de la distance 138
Figure 7.5 : La latence moyenne des Handovers en fonction de la charge du trafic 141
Figure 7.6 : Délais inter-trames obtenus avec le mécanisme *FHAP* 143
Figure 7.7 : Délais inter-trames par application de la technique *PSHP* 144
Figure 7.8 : Occurrence des Handovers dans *PSHP* 146
Figure 7.9 : Occurrence des Handovers en fonction de la charge du trafic 146

Liste des tableaux

Tableau 2.1 : Mappage entre les Priorités Utilisateurs *UP* et les Catégories d'Accès *AC* 36

Tableau 4.1 : Paramètres du modèle de simulation 68
Tableau 4.2 : Tableau récapitulatif des améliorations en débits moyens du *MAARF* 81

Tableau 5.1 : Valeurs du paramètre *CW[i]* du modèle *EDCA* révisé 96
Tableau 5.2 : Trafics adoptés par les stations lors des simulations 97
Tableau 5.3 : Trafics adoptés par les stations en l'absence de trafic multimédias 101
Tableau 5.4 : Taux d'utilisation du canal de transmission 102

Tableau 6.1 : Comparaison des phases du Handover 109
Tableau 6.2 : Table du graphe de voisinage 115

Tableau 7.1 : Paramètres de simulation 137
Tableau 7.2 : La latence moyenne de Handover avec différents types de scan 143
Tableau 7.3 : Probabilité moyenne de perte des paquets *VoIP* 146

Introduction générale

Point de départ et sujet de thèse

Nous assistons ces dernières années à une importante évolution dans la société de l'information, conduite par la commercialisation et l'émergence des appareils de communication (tels que les téléphones cellulaires, les ordinateurs portables, les assistants personnels, etc.) et la convergence des réseaux fixes et mobiles. L'utilisateur passe ainsi de l'âge de l'ordinateur personnel à l'âge de l'ubiquité du traitement à travers plusieurs infrastructures. Il a accès à l'information n'importe où et n'importe quand. Un utilisateur mobile peut consulter son courrier électronique, naviguer sur Internet dans les aéroports, les gares ou dans d'autres lieux publics.

Comme pour les réseaux mobiles, l'intégration des services multimédias dans les réseaux sans fil de la norme 802.11 a suscité ces dernières années un réel intérêt. Notons que les problématiques posées dans le contexte particulier des réseaux WiFi sont différentes et plus complexes que celles rencontrées dans le monde filaire. Pour cela, plusieurs travaux pour le support de la qualité de service ont été proposés. Ils sont globalement classés par couche de protocoles, et se focalisent essentiellement sur des problèmes liés à l'amélioration du débit général observé. De plus, le mécanisme standard de commutation entre les cellules dans la norme 802.11 (référencé par le terme *Handover* ou *Handoff* et adopté par les mobiles pour la recherche de la prochaine cellule) enregistre des délais d'exécution excessifs. Il serait alors incorrect de supposer que les protocoles actuels de la norme protègent les données bornées en temps de transmission à cause de leurs contraintes de *QoS* (*Quality of Service*).

Dans le cadre de nos travaux, nous nous sommes intéressés à la nécessité du respect de la qualité de service dans les réseaux mobiles WiFi normalisés IEEE802.11. La communication entre les couches ainsi que l'exploitation des informations échangées entre elles permettent de mieux garantir et de satisfaire les besoins des applications multimédias temps réel. Parallèlement, nous avons élaboré un protocole de commutation intercellulaire 802.11 qui permet de prendre en considération ces contraintes de *QoS* en termes de latence.

Un autre paramètre important dans les réseaux WiFi est le choix du débit physique adéquat. Dans les approches de contrôle du lien existantes, l'idée est de s'appuyer sur des paramètres,

comme le résultat des transmissions admises dans le canal, pour obtenir une décision sur le débit physique à déployer lors des prochaines émissions. Dans notre proposition, contrairement aux autres approches, nous agissons sur cette sélection par l'introduction de nouveaux paramètres afin d'assurer une meilleure connectivité, de minimiser le nombre de trames perdues dans le réseau, et de fournir par conséquent des liens sans fil stables.

La version 802.11e de la norme s'est intéressée à la différenciation de services par l'installation de plusieurs types de trafics. La séparation entre les trafics est réalisée simplement par un choix différent des valeurs des paramètres responsables de l'accès au medium. Cependant, la compétition inter-station pour le même type de trafic n'est pas gérée et elle aboutit à des pertes de trames lors des tentatives d'accès. Nous avons établi une nouvelle priorité par laquelle la concurrence d'accès au canal inter-catégorie est ordonnée et mieux gérée, et ainsi la transmission des applications multimédias sera privilégiée. Nous modifions le choix des valeurs des paramètres responsables de l'accès de manière à minimiser le nombre de collisions produites. Une nouvelle classification des stations actives dans le réseau permet de préserver les contraintes *QoS* exigées et de produire une meilleure utilisation du canal par la minimisation des pertes, ce qui a pour effet d'augmenter le débit général observé en sortie, ainsi que le respect des délais exigés par des trames bornées en temps de transmission (*Voix sur IP ou VoIP*).

Contributions et organisation de la thèse

Dans le premier chapitre nous commençons par une étude bibliographique sur les réseaux mobiles et plus précisément sur la norme IEEE802.11, son principe de fonctionnement et le protocole associé. Les parties significatives et importantes de cette norme utiles pour notre travail sont détaillées dans le deuxième chapitre. Plus spécifiquement, nous décrivons les extensions de la norme en particulier la méthode du choix du débit physique, la différenciation de service adoptée et les phases du mécanisme de Handover traditionnel.

Un nouveau mécanisme du choix du débit physique est exposé dans le chapitre 3. Après avoir présenté les approches actuelles ainsi que nos motivations, nous décrivons l'ensemble des extensions nécessaires à apporter au schéma d'adaptation du lien classique afin de tenir compte de la qualité instantanée du lien entre les stations mobiles. Cette nouvelle information sera extraite depuis le temps d'aller-retour calculé à partir des paquets déjà transmis. Nous tirons un

ensemble d'interprétations des premiers résultats théoriques observés lors de son application. Un ensemble de simulations, sous différents scenarii de mobilité et de trafics, est réalisé à l'aide de l'outil de simulation *Network Simulator 2*. Les résultats obtenus sont reportés dans le chapitre 4. Ils confirment, en termes de sélection instantanée du débit physique adéquat, une meilleure utilisation des ressources du réseau et un excellent débit moyen de sortie mesuré.

Dans le chapitre 5, nous montrons d'abord la nécessité d'une gestion de la qualité de service dans les réseaux mobiles sans fil et particulièrement dans les réseaux 802.11. Ensuite, nous présentons les principaux travaux effectués dans cet objectif à savoir la nouvelle version 802.11e de la norme. Sur cette dernière, nous apportons des modifications au niveau des paramètres décisionnels pour l'accès au canal en ajoutant une nouvelle priorité autre que celle déjà existante. Nous montrons que cette nouvelle technique, combinée avec l'ancienne, assure un total support des applications temps réel. Un ensemble de simulations est réalisé pour illustrer l'amélioration obtenue.

Concernant le troisième volet de cette thèse, nous le consacrons au mécanisme de commutation intercellulaire, qui présente le problème majeur de ce type de réseau sans fil, et qui forme actuellement, le centre d'intérêt des chercheurs pour l'amélioration du support des applications multimédias. Dans le chapitre 6, un état de l'art détaillé est exposé sur les dernières techniques accomplies sur le mécanisme *Handover 802.11* en mode infrastructure. La deuxième partie de ce chapitre est consacrée à la mesure de la latence totale réalisée par ce mécanisme. Cette dernière étude conduite montre l'incapacité du dispositif actuel à supporter les applications à contraintes temporelles strictes (comme la *Voix sur IP*).

Une nouvelle et innovante approche de commutation entre les cellules du réseau IEEE802.11 est proposée dans le chapitre 7. Son principe est basé sur le déploiement d'une nouvelle sous phase appelée *pré-scan* pour contrôler et guider le choix de la station mobile du prochain point d'accès auquel s'affilier. Cette phase de pré-scan permet de réduire le temps du la phase de découverte des points d'accès voisins.

Des simulations de cette nouvelle technique de Handover sont effectuées pour montrer la valeur ajoutée d'une telle proposition ainsi que son apport lorsqu'une transmission multimédia est acceptée sur un canal WiFi. Pour conduire de telles expériences, nous intégrons cette nouvelle méthode dans un nouveau simulateur des Handovers dans les réseaux sans fil 802.11.

Ce dernier a été développé pour ce fait et propose une implémentation complète du Handover traditionnel utilisé par la norme 802.11 ainsi que d'autres techniques récemment publiées afin de les comparer avec notre technique. Les outils adoptés dans cette plate-forme sont détaillés dans l'annexe A.

Certains des travaux présentés ici sont encore en cours, d'autres ont déjà fait l'objet de publications [74, 75, 76, 77, 78, 79, 80].

CHAPITRE

1 La technologie WiFi

Dans la seconde moitié des années 90, plusieurs normes pour les réseaux sans fil à portée limitée ont été développées (visant des usages à l'échelle du bureau ou du bâtiment). Leur arrivée a soulevé un engouement nouveau pour les réseaux radio multi-sauts, qui étaient jusqu'alors le domaine exclusif des militaires. Dans ce chapitre, nous allons présenter ces différentes normes, et en particulier nous focaliserons notre étude sur la norme 802.11 connue aussi sous le nom commercial de ***WiFi*** (*Wireless Fidelity*), qui est désormais utilisée dans la plupart des travaux appliqués de la communauté ***WLAN*** (*Wireless Local Area Network*).

I. Les réseaux sans fil

Les réseaux sans fil sont devenus ceux les plus utilisés actuellement dans le marché des réseaux locaux. Ils sont basés sur une liaison utilisant les ondes radioélectriques (radio et infrarouges) comme support de communication et remplaçant ainsi l'usage ennuyeux des câbles. Il existe plusieurs technologies : nous les distinguons par la fréquence d'émission utilisée, le débit alloué ainsi que la portée des transmissions. De nos jours, divers équipements sont accessibles aux entreprises et au grand public permettant des connexions entre les périphériques sans fil. Plusieurs communautés collaborent afin de standardiser les technologies de ces réseaux, en leur permettant en même temps d'être concurrentes et complémentaires.

En observant rapidement les technologies les plus répandues, nous remarquons que chaque niveau topologique dispose d'une ou de plusieurs norme(s) accessible(s) comme l'illustre la *Figure* 1.1.

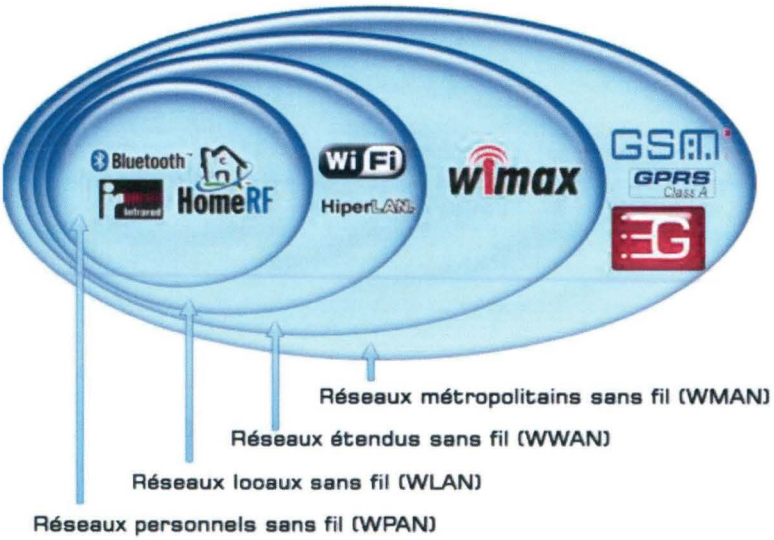

Figure 1.1 : Les standards des réseaux sans fil

Nous distinguons essentiellement quatre types de réseaux sans fil:

- ***WPAN*** (*Wireless Personal Area Network*) : Ce type de réseau concerne essentiellement les connexions entre les périphériques ou la constitution de micro réseaux. C'est le cas par exemple, avec les ***PDA*** (*Personal Digital Assistant*), les téléphones mobiles et bien sûr les micro-ordinateurs qui échangent des informations entre eux. Le protocole le plus répandu pour cette catégorie de connexions point à point est le Bluetooth.

- ***WLAN*** (*Wireless Local Area Network*) : Ce type de réseau sans fil est l'équivalent du *LAN* habituel. Il passe par une ou plusieurs borne(s) d'accès, ce qui permet à l'utilisateur de se connecter à un réseau filaire existant (exemple Internet). Si le réseau dispose de plusieurs bornes d'accès, l'utilisateur peut alors exécuter le *« roaming »*, c'est-à-dire se déplacer d'une zone de couverture à une autre tout en restant connecté au réseau.
- ***WMAN*** (*Wireless Metropolitan Area Network*) : Ce type de réseau utilise le même matériel qui est nécessaire pour constituer un *LAN* mais peut couvrir une plus grande surface comme un réseau urbain de l'ordre de quelques kilomètres.

- ***WWAN*** (*Wireless Wide Area Network*) : Ce type de réseau est basé sur les cellules, il est utilisé pour la téléphonie mobile (*GSM, GPRS, UMTS*, etc.).

Par contre, ces transmissions radio induisent plusieurs problèmes parmi lesquels nous pouvons citer :

- ✓ Un débit souvent plus faible qu'un réseau câblé.
- ✓ Une dégradation rapide du signal en fonction de la distance qui induit pour un émetteur l'impossibilité de détection d'une éventuelle collision au même moment. En effet, le medium utilisé est dit *half-duplex*, ce qui correspond à un medium sur lequel l'émission et la réception sont impossibles en même temps.
- ✓ L'inévitabilité des interférences : les transmissions radios ne sont pas isolées et le nombre de canaux disponibles est limité, ce qui force le partage. De plus les émetteurs travaillent à des fréquences trop proches. Les interférences peuvent donc être de natures diverses : des bruits naturels (pluie, vents, etc.), parasites (magnétiques, électriques, etc.), des phénomènes d'atténuation, de réflexion et de chemins multiples dus à l'environnement, etc.
- ✓ La limitation de l'énergie par l'autonomie des batteries. En effet, les applications relatives aux réseaux sans fil ont un caractère nomade portable. Emettre ou recevoir des données consomme de l'énergie, donc il faut prévoir dans ce cas des mécanismes d'économie d'énergie puisque les mobiles ne vont pas être connectés directement au réseau électrique.
- ✓ Une faible sécurité : il est très facile d'espionner passivement un canal radio.
- ✓ Une topologie de réseaux non fixe : la mobilité des nœuds provoque le changement de la topologie du réseau.

Cependant, l'utilisation des réseaux sans fil offre plusieurs avantages. Parmi les plus importants nous pouvons évoquer :

- ➢ La portabilité : un ordinateur portable ou un ordinateur de poche suffit pour se connecter.
- ➢ Le choix du lieu de connexion, sous contrainte d'être toujours sous la couverture du réseau.
- ➢ La flexibilité : la connexion est indépendante de la marque ou des caractéristiques techniques des appareils connectés. Seules les cartes réseaux doivent garantir une compatibilité avec la norme à laquelle elles font référence.
- ➢ La facilité : pas de câble signifie moins d'encombrement. Les appareils sur le marché tendent à se connecter automatiquement.

- La mobilité : les utilisateurs peuvent se déplacer sans interruption de la connexion au réseau.
- Le prix : il tend à baisser suivant l'évolution du marché. Il est quasiment impossible de trouver de nos jours un ordinateur portable sans carte réseau sans fil intégrée.

Cette technologie apporte également des avantages aux responsables du déploiement de ces types de réseaux, à savoir :

- Moins de câble à mettre en place, donc une diminution de l'investissement en coûts ainsi qu'en charge de travail lors de l'installation.
- Facilité et souplesse de déploiement : une machine supplémentaire peut se connecter facilement au réseau sans réservation d'un espace tel qu'une prise *RJ45* dans les réseaux filaires.
- Le prix : une solution sans fil pourrait être largement moins chère pour une entreprise.

Aujourd'hui, de nouvelles technologies portant sur le principe de communications sans fil sont apparues et ont gagné une large popularité par rapport aux « anciens » réseaux câblés. Nous pouvons citer par exemple : les *WWAN* (*GPRS, GSM*), les réseaux satellites, les réseaux locaux sans fil tels que les *WPAN* (*Bluetooth, HomeRF*) ou les *WLAN* (*WiFi, HiperLAN*). Après cette brève introduction, nous allons nous focaliser essentiellement dans la suite de ce chapitre sur la technologie WiFi.

II. Le standard IEEE 802.11

WiFi (*Wireless Fidelity*) est un réseau local radio, il correspond à la norme IEEE 802.11. Cette norme de réseau informatique sans fil a été définie par le consortium IEEE (Institute of Electrical and Electronics Engineers) en 1999 [1].

Il a d'abord été conçu pour fournir des accès à haut débit pour des utilisateurs nomades dans les entreprises, puis dans des lieux de passage à large public (tels que des gares, des hôtels, des aéroports, des trains,...). Il a ainsi permis de mettre à la portée de tous un vrai système de communication sans fil pour la mise en place des réseaux informatiques hertziens.

Le WiFi est une technologie standard d'accès sans fil à des réseaux locaux *WLAN*. Le principe consiste à établir des liaisons radio rapides entre des terminaux et des bornes reliées aux réseaux à haut débit. Grâce à ces bornes WiFi, l'utilisateur se connecte à Internet ou au système d'informations de son entreprise et accède à de nombreuses applications reposant sur le transfert de données. Cette technologie a donc une réelle complémentarité avec les réseaux **ADSL** (*Asymetric Digital Subscriber Line*) ainsi que les réseaux d'entreprise (comme illustré dans la *Figure* 1.2).

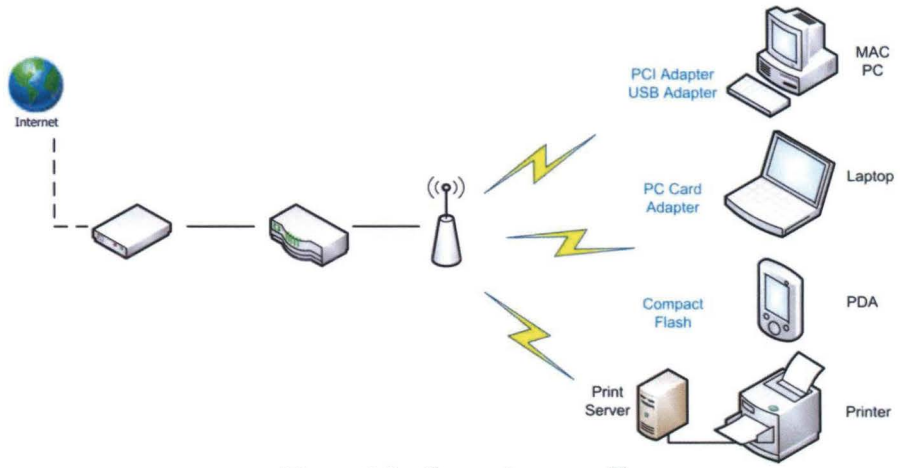

Figure 1.2 : Connexion sans fil

Ce standard a été développé pour favoriser l'interopérabilité du matériel entre les différents fabricants ainsi que pour permettre de futures évolutions compatibles avec la norme. Ainsi, les consommateurs peuvent mélanger des équipements de différents fabricants afin de satisfaire leurs besoins de connexion, ce qui a permis à cette norme un développement conséquent qui lui vaut d'être la plus utilisée actuellement et d'être la plus fournie par tous les fabricants d'équipements de réseaux et de téléphonie.

1. Les normes 802.11

La norme IEEE 802.11 [1] est en réalité la norme initiale qui offre des débits de 1 ou 2 Mb/s. Des évolutions ont ensuite été apportées à cette norme originale afin d'optimiser le débit (802.11a, 802.11b et 802.11g), la sécurité (802.11i), l'interopérabilité ou de manière à gérer des services de base supplémentaires comme la qualité de service (802.11e). Nous exposons ci-après une description des différentes évolutions de la norme 802.11 ainsi que leurs caractéristiques.

- **802.11a** [2]: Elle permet d'obtenir un haut débit (54 Mb/s théoriques, 30 Mb/s réels). Cette norme spécifie 8 canaux radio dans la bande de fréquence ***U-NII*** (*Unlicensed-National Information Infrastructure*) de 5 GHz utilisant une modulation de type ***OFDM*** (*Orthogonal Frequency Division Multiplexing*). L'inconvénient majeur de cette norme est qu'elle est incompatible avec le 802.11b.
- **802.11b** [3]: C'est la norme la plus répandue actuellement. Elle propose un débit théorique de 11 Mb/s (≈ 6 Mb/s de débit réel) avec une portée pouvant aller jusqu'à 300 mètres dans un environnement dégagé. La plage de fréquences utilisée est la bande ***ISM*** (*Industrie Scientifique Medicale*) 2.4 GHz, avec 3 canaux radio disponibles. Cette norme est définie par une modulation ***DSSS*** (*Direct Sequence Spread Spectrum*), garantissant un accès au medium par la méthode ***CSMA/CA*** (*Carrier Sense Multiple Access / with Collision Avoidance*) avec une détection de la porteuse.
- **802.11c :** Des modifications concernant la gestion de la couche ***MAC*** (*Medium Access Control*) ont été apportées. Cette révision a amélioré les procédures de connexion en pont entre les ***APs*** (*points d'accès*).
- **802.11d** [4]: Son but est de permettre une utilisation internationale des réseaux locaux 802.11. Elle consiste à adapter les couches physiques des différents équipements pour échanger des informations sur les plages de fréquences et les puissances conformes aux réglementations des pays.
- **802.11e** [5]: Cette norme s'est intéressée à l'amélioration de la qualité de service (*QoS*) au niveau de la couche liaison de données. Ainsi, elle a pour but de définir les besoins des différents paquets en termes de bande passante et de délai de transmission de telle manière à permettre notamment une meilleure transmission de la voix et de la vidéo sur ce type de canal.
- **802.11f** [6]: Cette version de la norme permet à un utilisateur itinérant de changer de point d'accès de façon transparente lors d'un déplacement, indépendamment des marques des points d'accès. Cette possibilité de passage d'une cellule à une autre sans interruption de la communication est appelée itinérance (ou *roaming*).
- **802.11g** [7]: Elle offre un haut débit théorique allant jusqu'à 54 Mb/s sur la bande de fréquence *ISM* 2.4 GHz. De plus, elle offre une compatibilité avec la norme 802.11b, ce qui signifie que des matériaux conformes à la norme 802.11g pourront fonctionner en 802.11b.

La norme 802.11g utilise aussi la modulation ***OFDM*** (*Orthogonal Frequency Division Multiplexing*).

- **802.11h** [8]: Elle vise à adapter la couche *MAC* pour rendre compatibles les équipements de la norme 802.11 avec des infrastructures utilisant le réseau *HiperLAN2* [9] et d'assurer la conformité avec la réglementation européenne en matière de fréquences utilisées et de l'économie d'énergie associée.
- **802.11i** [10]: Le but de cette norme est d'améliorer la sécurité des transmissions (gestion et distribution dynamique des clés, chiffrement des informations et authentification des utilisateurs). Elle s'appuie sur le chiffrement ***AES*** (*Advanced Encryption Standard*).
- **802.11j** [11]: Le but de la norme 802.11j est de rendre compatible 802.11a avec la réglementation japonaise.
- **802.11k** [12]: Cette norme présente des spécifications sur l'environnement radio afin d'améliorer les performances et supporter les nouvelles applications émergeantes (Voix sur IP, Vidéo sur IP ainsi que de nouvelles applications basées sur la localisation).
- **802.11IR** : La norme *802.11IR* a été élaborée afin d'utiliser des signaux infrarouges. Cette norme est désormais dépassée techniquement.
- **802.11n** [13]: La norme 802.11n propose des débits beaucoup plus rapides (plus de 100 Mb/s) en utilisant la technologie de transmission ***MIMO*** (*Multiple Input Multiple Output*).
- **802.11r** [14]: Traite le changement de cellules rapide en s'attachant au niveau du contenu, afin de ne pas perdre d'informations lors de la transmission de flux continus comme la vidéo ou la voix sur ***IP*** (*Internet Protocol*).
- **802.1s** [15]: Cette norme propose d'exploiter toutes les stations terminales d'équipements Wifi afin d'acheminer des données d'un point à un autre.

2. Les modes de connexion supportés

Le standard IEEE 802.11 [1] définit deux modes de connexion ou deux types d'architecture :

- *Le mode Ad-hoc* dans lequel les clients sont connectés les uns aux autres sans présence de points d'accès (absence de bus commun).
- *Le mode Infrastructure* dans lequel les clients sans fil seront connectés à des points d'accès. Il s'agit généralement du mode par défaut et le plus répandu des cartes Wifi 802.11g.

A. Le mode Ad-hoc

Dans ce mode, les stations sans fil se connectent les unes aux autres sans passer par aucun point d'accès afin de constituer un réseau point à point, c'est-à-dire un réseau dans lequel chaque machine joue en même temps le rôle de client et de serveur (comme illustré dans la *Figure* 1.3).

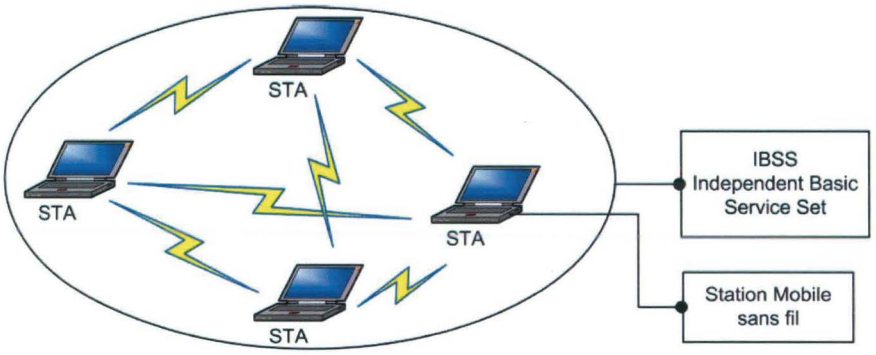

Figure 1.3 : Mode Ad-hoc

L'ensemble formé par les différentes stations est appelé ***IBSS*** (*Independant Basic Service Set*). Un *IBSS* est un réseau sans fil constitué au minimum de deux stations communicantes sur la même bande et n'utilisant pas de point d'accès, c'est une transmission directe. Un *IBSS* est identifié par un ***SSID*** (*Service Set IDentifier*).

B. Le mode Infrastructure

En mode Infrastructure, chaque station se connecte à un point d'accès via une liaison sans fil. L'ensemble formé par le point d'accès et les stations situées dans sa zone de couverture est appelé ***BSS*** (*Basic Service Set*) et constitue par conséquence une cellule. Dans ce cas, les stations communiquent entre elles en passant par un point d'accès. Chaque *BSS* est identifié par un ***BSSID*** (*Basic Service Set IDentifier*) composé de 6 octets (48 bits), et correspondant à l'adresse MAC du point d'accès (voir *Figure* 1.4).

Plusieurs points d'accès (ou plusieurs *BSS*) peuvent être reliés entre eux par une liaison généralement filaire (de type Ethernet) appelée ***DS*** (*Distribution System*). Le réseau ainsi construit constitue un ***ESS*** (*Extended Service Set*) qui englobe l'ensemble des cellules et leurs points d'accès associés. Il constitue un ensemble de services étendu. Le système de distribution *DS* peut être aussi bien un réseau filaire qu'un réseau sans fil, métropolitain ou mondial mais pas un autre *WLAN* local.

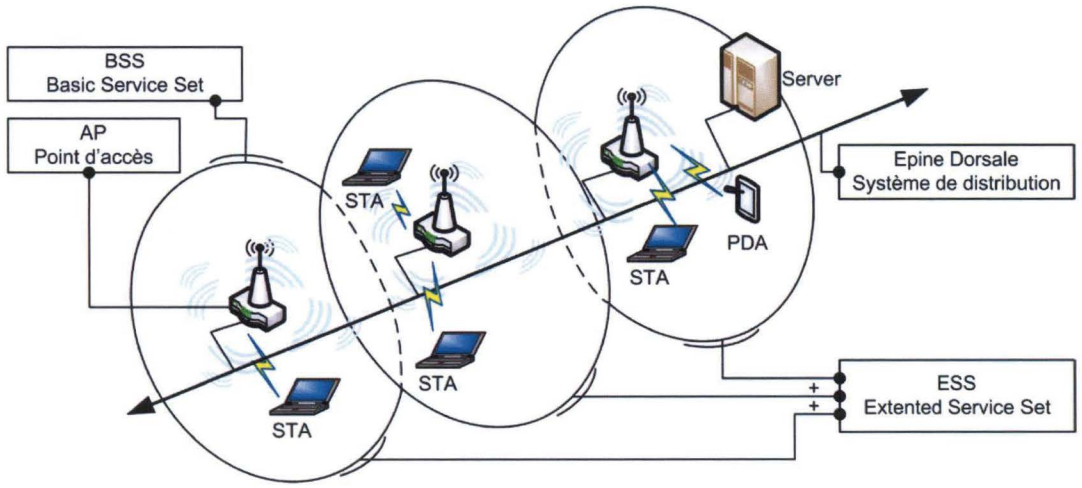

Figure 1.4 : Mode Infrastructure

Pour une architecture en mode Infrastructure à plusieurs cellules, les bornes de connexion (ou les points d'accès APs) sont connectées entre elles. Les terminaux peuvent alors se déplacer au sein de la cellule et garder une liaison directe avec le même point d'accès, ou changer de cellule et s'affilier avec une nouvelle borne. C'est ce que nous appelons le *roaming* .

3. Le modèle OSI

Comme toutes les normes définies par le comité IEEE 802, la norme 802.11 couvre les deux premières couches du modèle OSI, c'est-à-dire la couche physique et la couche liaison de données, comme indiqué dans la *Figure* 1.5.

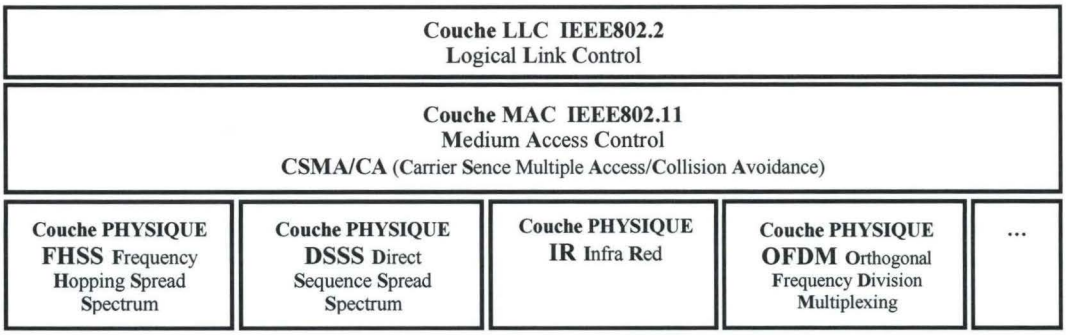

Figure 1.5 : Les couches 1 et 2 de la norme 802.11

A. La couche Physique

La couche physique ([16], [17]) définit le type de la modulation utilisée pour les ondes radioélectriques ainsi que les caractéristiques de la signalisation pour la transmission de données.

Cette couche est responsable de la transmission réelle des bits transformés en un signal sur le canal de communication. Elle définit la conversion de l'information entre le format numérique et analogique ainsi que le choix de la technique physique de transmission.

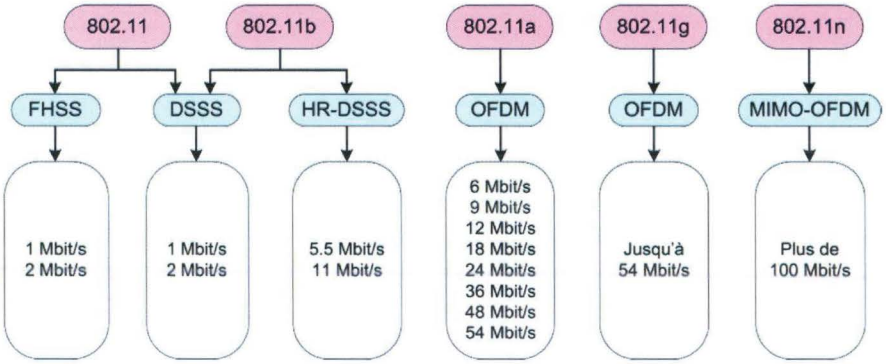

Figure 1.6 : Récapitulatifs des technologies et des débits possibles

Les principales technologies de transmissions proposées par la couche physique sont : ***IR*** (*Infra Red*), ***FHSS*** (*Frequency Hopping Spread Spectrum*), ***DSSS*** (*Direct Sequence Spread Spectrum*), ***OFDM*** (*Orthogonal Frequency Division Multiplexing*). La *Figure* 1.6 récapitule les technologies supportées ainsi que les débits associés.

- **IR :** La technologie infrarouge a pour caractéristique principale d'utiliser une onde lumineuse pour la transmission de données. Il est possible grâce à cette technologie d'obtenir des débits allant de 1 à 2 Mb/s.

- **FHSS :** en français « *étalement de spectre par saut de fréquence* », consiste à découper la large bande de fréquence, puis de transmettre en utilisant une combinaison de canaux connue de toutes les stations de la cellule courante. Dans la norme 802.11, la bande de fréquence *ISM* (2.4 - 2.4835 GHz) permet de créer 79 canaux de 1 MHz. La transmission est ainsi réalisée en émettant successivement sur un canal puis sur un autre pendant une courte période de temps (d'environ 400 ms), ce qui permet à un instant donné de transmettre un signal plus facilement reconnaissable sur une fréquence donnée.

- **DSSS :** en français « *étalement de spectre à séquence directe* », consiste à transmettre pour chaque bit une séquence de 11 bits appelée *"chipping"* pour le codage des données. Ainsi chaque bit valant 1 est remplacé par la séquence de bits (10110111000) et chaque bit valant 0

par son complément (01001000111). Dans la norme 802.11, la bande de fréquence *ISM* (2.4 - 2.4835 GHz) est divisée en 14 canaux de 22 MHz. Les canaux adjacents se recouvrent partiellement, seuls trois canaux sur les 14 étant entièrement isolés. Les données sont transmises intégralement sur l'un de ces canaux de 22 MHz, sans saut.

- **OFDM :** en français *« multiplexage par division orthogonale de fréquences »*, consiste à diviser la bande de fréquence en bandes secondaires qui transmettent simultanément des fractions de données. Plus le nombre de canaux est élevé, plus les données transmises en parallèle sont nombreuses, et plus la bande passante est élevée. Selon les conditions de bande passante, *OFDM* peut utiliser des méthodes de modulation de phase et d'amplitude.

B. La couche Liaison de Données

Elle définit l'interface entre le bus de la machine et la couche physique. Elle s'appuie notamment sur une méthode d'accès très proche de celle utilisée dans le standard Ethernet puisqu'il y a une utilisation multiple du même support de communication par tous les usagers du réseau. Vue sa complexité en nombre de fonctionnalités requises, cette couche est divisée en deux sous-couches : ***LLC*** (*Logical Link Control*) pour le contrôle de la liaison logique et ***MAC*** (*Media Access Control*) pour définir les attributs de la méthode d'accès au support.

i. La sous-couche LLC

La norme 802.11 [1] utilise les mêmes propriétés que la sous-couche *LLC* 802.2 [18]: un adressage sur 48 bits, simplifiant ainsi le pontage entre les réseaux sans fil et câblés. Donc, de cette façon, il est possible de relier un réseau 802.11 *WLAN* à tout autre réseau local appartenant à une norme de l'IEEE.

ii. La sous-couche MAC

La sous-couche *MAC* comprend les dernières tâches du modèle ***OSI*** (*Open System Interconnection*) avant de convertir les données numériques et passer à leur transmission réelle sur le support. En d'autres termes, elle consiste à acheminer les informations entre la couche physique et la sous-couche *LLC*. Dans la norme 802.11, la sous-couche *MAC* est très proche dans sa conception de celle de la norme 802.3. En effet, elle est conçue pour supporter de multiples utilisateurs sur un support partagé en faisant détecter l'état du support (libre ou occupé) par

l'émetteur avant d'y accéder. La sous-couche *MAC* définit deux méthodes d'accès au support: ***PCF*** (*Point Coordination Function*) et ***DCF*** (*Distributed Coordination Function*).

Ces deux méthodes sont basées sur l'utilisation des temporisateurs. Chaque trame est délimitée par un espace temps. Cet espace permet la gestion d'accès au support en temporisant l'envoi des trames. Selon le type des temporisateurs utilisés, on peut différencier la priorité d'accès (voir *Figure* 1.7). Ces temporisateurs, notés ***IFS*** (*Inter Frame Spacing*) ou *« l'espace temps entre les trames »*, correspondent à des intervalles de temps entre l'émission de deux trames. Plus l'*IFS* est court plus l'accès est prioritaire. Il existe trois types d'espaces temps *Inter-Frame* différents selon 802.11 :

1. ***SIFS*** (*Short IFS*) : c'est le plus court des *IFS*, donc le plus prioritaire. Il est utilisé pour la transmission d'un même dialogue : données et ***ACK*** (*acknowledgment* ou appelé aussi *acquittement* en français) accusé de réception de la station réceptrice et le reste des données (ou les trames d'information de la station émettrice) restent prioritaires jusqu'à leur transmission totale.
2. ***PIFS*** (*PCF IFS*) : espace inter trame utilisé pour les trames *PCF* (accès contrôlé) par le point d'accès. Permet un accès prioritaire de cet *AP* sur les stations du réseau. Sa valeur correspond à un *SIFS* plus un temps (appelé *slot-time*).
3. ***DIFS*** (*DCF IFS*) : temporisateur inter trame pour l'accès distribué utilisé par les stations pour accéder au support en mode *DCF*. Ce temps est le moins prioritaire par rapport aux autres espaces temps. Il est utilisé pour initialiser une nouvelle transmission sur le réseau.

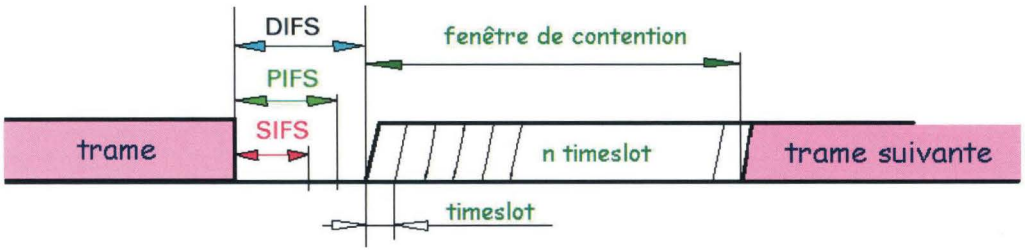

Figure 1.7 : La gestion d'accès dans la norme 802.11

Ces temporisateurs permettent la gestion d'accès au support en temporisant l'envoi de trames. Les stations actives sur le réseau et voulant transmettre des données, doivent donc attendre ces différents *IFS* (suivant le cas) avant d'accéder au canal de communication.

4. Les méthodes d'accès au medium dans les réseaux 802.11

A. La méthode DCF : Distributed Coordination Function

Cette méthode est généralement la plus déployée dans ce type de réseau local sans fil et peut être utilisée par tous les mobiles. Elle permet un accès équitable au canal radio sans aucune centralisation de la gestion de l'accès. C'est une méthode totalement distribuée et aléatoire. Elle est utilisée par les deux types d'architectures réseau : le mode infrastructure et le mode Ad-hoc (sans point d'accès). Le *DCF* est basé sur la méthode d'accès ***CSMA/CA*** (*Carrier Sense Multiple Access with Collision Avoidance*). Nous détaillerons, dans ce qui suit, ce mécanisme déployé à chaque nouvelle transmission sur le support de communication.

i. Le mécanisme CSMA/CA du 802.11

Dans un réseau Ethernet 802.3 [19], la méthode d'accès utilisée par les machines est le ***CSMA/CD*** (*Carrier Sense Multiple Access with Collision Detection*). Ce dernier consiste à écouter le canal et à ne procéder à la transmission d'une trame que si ce dernier est libre. Dans le cas où le canal est occupé par une autre communication, la station voulant émettre diffère son accès. Si le medium est libre, la station commencera la transmission réelle des signaux en même temps que l'écoute simultanée du canal. Une fois que l'émetteur écoute des signaux différents de ceux qu'il est en train de transmettre conjointement, il détecte une anomalie et juge qu'une collision s'est produite avec une ou plusieurs autres trames émises en même temps que la sienne. L'émetteur démarre ensuite une procédure suite à cette collision afin d'informer les destinataires par une succession de bits à '0' (appelés des bits de bourrage) indiquant que cette trame doit être erronée. Un nouveau temps d'attente doit être calculé et respecté avant la prochaine tentative d'accès au canal. Le calcul de cette durée aléatoire est réalisé par l'utilisation de l'algorithme de *backoff* (sera détaillé ultérieurement). Cette durée est aléatoire et n'est pas déterminée à l'avance afin que les stations émettrices évitent de prochaines collisions en attendant des durées différentes les unes des autres.

Cette méthode d'accès ***CSMA/CA*** (*Carrier Sense Multiple Access with Collision Avoidance*) est celle qui est actuellement déployée dans les réseaux 802.11. L'ancienne version *CSMA/CD* n'est pas applicable dans un environnement radio pour deux raisons :

- Les liaisons radio utilisées ne sont pas de type full-duplex (nous ne pourrons jamais écouter et émettre en même temps).
- Une machine qui écoute la porteuse n'est pas certaine d'écouter toutes les stations connectées au point d'accès (cas de la station cachée).

Pour ces deux raisons le mécanisme *CDMA/CD* est modifié par une nouvelle technique d'accès *CSMA/CA* parfaitement adaptée à cet environnement de transmission.

La méthode d'accès *CSMA/CA* utilise plusieurs techniques pour palier cette impossibilité d'écoute en émission. Tout d'abord un système d'accès au support basé sur des temporisateurs, un système d'acquittement positif. Ensuite, une bonne gestion de reprise après collision par application de temps d'attente aléatoire. Enfin, une technique optionnelle permettant de sécuriser la transmission des données et d'éviter des collisions avec des nœuds cachés. La norme 802.11 utilise la nouvelle méthode d'accès *CSMA/CA*, qui permet d'écouter le canal et d'éviter au maximum les collisions, en imposant un accusé de réception systématique des paquets *ACK*. Cette méthode s'appuie sur l'évitement des collisions puisque ces dernières sont très difficiles à gérer dans un tel environnement et ne sont détectées que par le récepteur du message.

Avant toute émission, la station écoute le canal de transmission. Si le réseau est occupé, la transmission est retardée. Si le réseau est libre pendant un *DIFS*, elle attendra avant d'émettre encore un autre temps aléatoire. Cette durée supplémentaire est calculée par le même algorithme *Backoff* (celui utilisé par le *CSMA/CD*). Elle diffère d'une station à une autre et de la fréquence de collision. Elle est déployée afin d'éviter que deux stations attendent le même *DIFS* lorsque le canal est libre, et par la suite débutent simultanément la transmission de leurs trames respectives. Cet algorithme diminue considérablement le risque d'avoir des collisions, dans le cas où plusieurs stations veulent émettre en même temps. La station qui aura le plus petit temps aléatoire d'attente après l'écoulement de *DIFS* occupera le canal et commencera son émission.

L'algorithme de *Backoff* utilisé pour le calcul de ce temps est très simple à mettre en œuvre. Le temps d'attente aléatoire varie entre 0 et *CW* (***CW**: Collision Windows* ou *la fenêtre de collision*). Il est calculé de cette manière : *Attente = Random (0, CW) * slot-time*. Chaque fois que le support est libre, l'attente est décrémentée de 1 *slot-time*. Dés qu'il atteint la valeur 0, alors la station commence l'émission. Ainsi un accusé de réception *ACK* est émis par la station réceptrice pour chaque paquet de données reçu correctement, après un *SIFS*, comme illustré dans

la *Figure* 1.8. Dans le cas d'une autre collision consécutive suite à un même temps aléatoire d'attente retourné par l'algorithme, la taille de la fenêtre (autrement dit la valeur de *CW*) va doubler afin de diminuer les chances que de telles collisions se répètent pour les prochaines tentatives. L'incrémentation de cette fenêtre est arrêtée après 10 doublements de sa valeur (donc la valeur *CW* est constante dès la dixième tentative). Après un nombre précis (généralement égal à 16) de tentatives consécutives de transmission de la même trame d'informations, la station émettrice abandonne la diffusion de cette dernière et informe les couches supérieures.

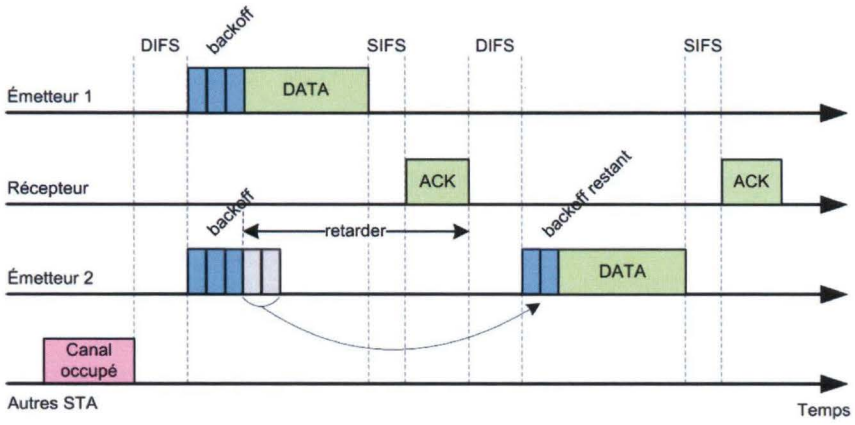

Figure 1.8 : Le mécanisme CSMA/CA du 802.11

Il est très important de mentionner que l'algorithme de *Backoff* limite les risques de collision mais ne les supprime pas complètement. Nous pourrons avoir le cas d'une collision lorsque deux stations auront le même temps aléatoire d'attente ou en l'absence d'*ACK* retourné à l'émetteur.

Certes, le *CSMA/CA* est considéré comme une bonne solution de partage d'accès au médium dans un réseau sans fil 802.11. Ce mécanisme d'accusé de réception explicite gère aussi très efficacement les interférences et d'autres problèmes radio. Cependant, face à certains cas d'utilisation, cette méthode devient impuissante et insuffisante. Il s'agit du problème des nœuds cachés dans lequel les émetteurs qui ne peuvent pas du tout s'entendre veulent atteindre un même récepteur. Dans ce cas, l'un ne détecte pas l'activité de l'autre. En général, à cause d'un obstacle, ils croient que le réseau est libre et transmettent dés qu'ils ont des données disponibles, ce qui génère une collision automatiquement.

ii. Le mécanisme RTS/CTS (Request To Send/Clear To Send)

Pour réduire la probabilité d'avoir deux stations ne pouvant pas se voir (pas de détection de signal) qui entrent en collision, la norme 802.11 définit sur la couche *MAC* un mécanisme optionnel utilisant les paquets de contrôle ***RTS*** (*Request To Send*) et ***CTS*** (*Clear To Send*).

Si le réseau est libre pendant une période *DIFS*, la station émettrice attend une durée aléatoire supplémentaire. Après l'écoulement de cette durée, la station transmet un paquet *RTS*, qui contient les informations suivantes : l'adresse source, l'adresse destination et la durée de la transmission. Le récepteur (un point d'accès dans le mode Infrastructure) répond par un paquet *CTS* qui inclura les mêmes informations sur la durée totale réservée pour cette transmission actuelle, juste après une période *SIFS*. À la réception de ce paquet, la station commence l'émission des données après un *SIFS* dont la bonne réception est confirmée par un paquet *ACK*. L'apport de ce mécanisme est que chaque station active sur le même réseau va mettre à jour son propre vecteur réseau ***NAV*** (*Network Allocation Vector*) et ne procédera à aucune transmission pendant la période annoncée, comme illustré dans la *Figure* 1.9. Les *NAV*s des stations contiendront les informations nécessaires sur la durée de la transmission actuelle occupante le canal de communication. Donc les paquets *RTS* et *CTS* permettent de réserver le canal pendant la durée de transmission des données.

La mise à jour des *NAV*s de tous les nœuds n'est pas simultanée : les stations proches de l'émetteur (à sa portée) recevront le paquet *RTS* et elles procéderont dès cet instant à la mise à jour de leurs propres *NAV*s. Les nœuds qui seront cachés à l'émetteur (ne détectant pas son signal) appliqueront la nouvelle mise à jour seulement après la réception de la trame de réponse *CTS* du destinataire du message.

Ce nouveau mécanisme, additionné au standard IEEE 802.11, élimine considérablement le problème des stations cachées. Cependant, nous devons noter que cette technique admet des coûts supplémentaires sur le temps total d'occupation du canal ainsi que sur le trafic engendré en conséquence.

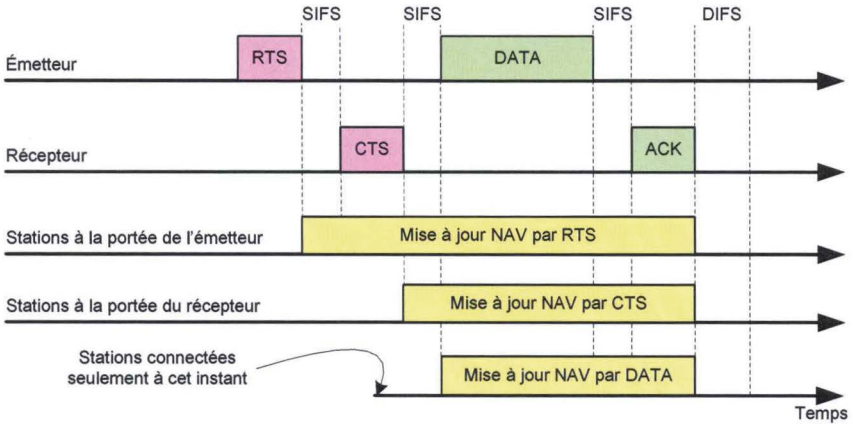

Figure 1.9 : Paquets RTS/CTS et mise à jour des NAVs

B. La méthode PCF : Point Coordination Function

Dans cette méthode, les stations de base (ou les points d'accès) ont la charge de la gestion de l'accès au canal pour les mobiles qui leurs sont rattachés (dans leurs zone de couverture), c'est une méthode totalement centralisée. Elle est utilisée seulement par le mode Infrastructure et ne fonctionnant qu'avec la présence des *APs*. Ce mode est optionnel dans la norme 802.11. Il consiste à définir un ***PC*** (*Point Coordinator*) au niveau de l'*AP*. Ce *PC* pourra ensuite donner la main à tour de rôle à chaque station afin de lui allouer le droit de transmettre. C'est le principe du polling qui permet l'élimination des contentions. Avec cette méthode d'accès, une trame peut être émise immédiatement après seulement un *PIFS*, c'est à dire sans attente supplémentaire après la libération du canal. Le fonctionnement de la méthode d'accès *PCF* est décrit ci-dessous.

Nous rappelons qu'en pratique le temps inter-espace *PIFS* est plus court que celui du *DIFS*. Ces temps sont fixés de telle sorte que le mode *PCF* soit prioritaire sur l'accès au canal si celui-ci est occupé par rapport au mode distribué *DCF*. Si le support est libre pendant *PIFS*, le *PC* doit transmettre une trame *Beacon* pour indiquer le passage en mode *PCF* contenant la longueur de sa période. À la réception de cette trame, les stations mettent à jour leurs vecteurs *NAV*s et ne transmettront plus pendant cette durée. Après *SIFS*, le *PC* peut transmettre des trames de données aux stations. Ce dernier peut aussi envoyer des trames *CF_Poll* (au lieu des trames *Beacon*) pour autoriser une station à transmettre des trames de données. Ces trames sont acquittées. Si les acquittements n'arrivent pas, alors une retransmission est effectuée après *PISF*. La trame *CF_End* annonce la fin de la période *PCF* (voir *Figure* 1.10).

La méthode d'accès au medium *PCF* garantit une transmission à un rythme régulier, permettant de synchroniser les flux (images, sons ou autres) ou de travailler en temps réel. Cette méthode a plutôt été destinée à des échanges à caractère multimédia nécessitant une certaine qualité de service.

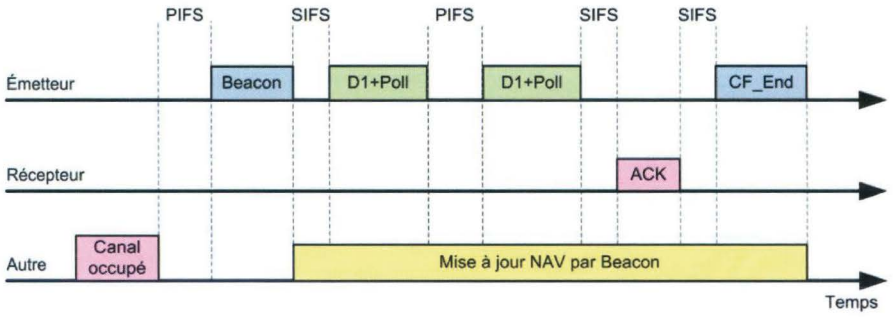

Figure 1.10 : Le mode Point Coordination Function « PCF »

Ce mode est appliqué lorsqu'une station admet des données urgentes à transmettre. Ces données seront alors prioritaires et la station concernée n'acceptera pas de temps d'attente considérables pour que le canal soit libre. Aussi, en choisissant ce mode, les données voyagent instantanément et sans risque de collision puisque le medium de communication sera réservé à cette transmission.

5. La sécurité dans la norme 802.11

Le problème spécifique des réseaux radio est que le canal est accessible à tous les équipements dotés du matériel nécessaire et se trouvant dans la zone de couverture. Donc, il est très courant et facile d'avoir une personne non autorisée qui écoute passivement les communications radio. Afin de se prémunir contre ces écoutes passives, le chiffrement des données est donc nécessaire si nous recherchons à avoir une certaine confidentialité.

Nous exposons dans ce qui suit une brève description de ce problème majeur des réseaux locaux sans fil ainsi que les remèdes actuellement appliqués. Cette étude ne sera pas parmi nos objectifs de recherche mais il s'agit simplement d'un balayage à titre informationnel de la norme étudiée. Dans la littérature [20], il existe diverses méthodes de sécurisation permettant d'éviter les interférences et l'espionnage en se basant sur les techniques suivantes :

- La liste de contrôle d'accès permet de spécifier les adresses *MAC* des utilisateurs autorisés à utiliser le réseau sans fil.

- L'identificateur de réseau *SSID* qui permet de donner au réseau un nom unique et non visible lors de la découverte des réseaux, de manière à ce que seules les stations autorisées puissent le voir et y accéder.
- Le cryptage ***WEP*** (*Wired Equivalent Privacy*) [20] est une méthode de chiffrement qui peut être utilisée de manière optionnelle. Le système repose sur une clé secrète qui doit être partagée par l'émetteur et le récepteur. La clé sert à créer un nombre pseudo-aléatoire d'une longueur égale à la longueur de la trame. Chaque transmission de données est ainsi chiffrée en utilisant le nombre pseudo-aléatoire comme masque grâce à l'opérateur booléen OU Exclusif (entre le nombre pseudo-aléatoire et la trame). De nos jours, cette technique n'est plus largement déployée puisqu'elle a perdu son efficacité (sur Internet il existe quelques logiciels gratuits permettant en l'espace de quelques heures de s'infiltrer et de se connecter sur un réseau 802.11 utilisant ce type de cryptage).
- Le cryptage ***WPA*** (*WiFi protected Access*) [20] est une solution de sécurisation proposée par la « *WiFi Alliance* », afin de combler les lacunes du *WEP*. Il ne supporte que les petits réseaux en mode infrastructure. Le fonctionnement de *WPA* repose sur la mise en œuvre d'un serveur d'authentification, permettant d'identifier les utilisateurs sur le réseau et de définir leurs droits d'accès. Il est beaucoup plus performant que l'ancienne version et il est bien plus difficile de casser la clé *WAP* puisqu'elle repose sur un algorithme plus complexe.

La sécurité est indéniablement le point faible des réseaux sans fil. La facilité de déploiement d'un réseau sans fil conduit à négliger les failles de sécurité que recèle la communication sans fil et donc à oublier la mise en place des mesures de sécurité adéquates. Par exemple dans la norme 802.11b, les différents utilisateurs du réseau utilisent un identifiant de réseau *SSID*, généralement codé par défaut et stocké dans le registre, pour identifier les appareils d'un même réseau. Par ailleurs, le protocole de connexion utilise le chiffrement *WEP*, vieux de 2 ans avec une clé de 40 bits alors que tous les spécialistes de sécurité informatique s'accordent à dire aujourd'hui qu'elle devrait être de 128 bits minimum. Deux standards sont en développement pour corriger cette faille de sécurité : le 802.11i qui gère le chiffrement et la gestion de clés utilisateurs et le 802.1x qui fait partie du 802.11i, et qui définit la sécurisation d'accès au réseau via l'usage de certificats et l'authentification mutuelle entre les clients et les points d'accès.

Les normes WiFi sont une révolution pour les réseaux sans fil et présentement les plus déployées. Leurs évolutions ne s'arrêteront pas, les principaux défauts des versions précédentes

de 802.11 disparaîtront dans les futures versions. Ainsi, la sécurité, les débits, la qualité de service et l'optimisation du mécanisme de Handover représentent actuellement de vastes domaines de recherche pour l'amélioration du standard 802.11.

III. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art des réseaux sans fil et plus spécifiquement sur le standard 802.11 le plus répandu actuellement sur le marché ainsi que les détails relatifs à son fonctionnement. Nous allons dans un second chapitre présenter les champs de recherche courants dans la norme 802.11. Par ailleurs, nous dévoilerons les éventuelles difficultés rencontrées lors du déploiement de ce type de *WLAN* pour effectuer des transmissions multimédias (appelées aussi à *QoS* exigée) devenues très demandées par les usagers de la toile vu la croissance des technologies dans le domaine du multimédia.

CHAPITRE

2 La qualité de service dans les réseaux WiFi

I. Contexte et objectifs

Nous focaliserons notre étude sur les problèmes rencontrés dans cette norme de communication face aux besoins des utilisateurs qui doivent être résolus en terme de qualité de service (*QoS*). Ce dernier terme est très général et englobe plusieurs aspects sur la qualité suivant le cas d'utilisation. Habituellement, pour le domaine des réseaux la *QoS* est similaire à une communication : sans perte, sans retransmission, adoptant le meilleur débit disponible et respectant les contraintes temporelles de livraison (le cas de la Voix et de la Vidéo). Dans ce chapitre, nous présentons les mécanismes déployés par le standard pour satisfaire ces exigences à savoir l'adaptation du débit physique, la gestion du trafic et la différenciation de service ainsi que la technique de commutation entre les cellules (ou entre les *APs*).

II. Adaptation du débit physique

Partant du principe que les réseaux sans fil sont basés sur une liaison radioélectrique (radio et infrarouge) ils sont par conséquent sensibles aux différentes variations de l'état du canal et à la coupure d'une connexion ainsi qu'à la non-commodité aux applications multimédias (c'est-à-dire pas de respect des obligations liées à ce type de données). Puisque les systèmes qui utilisent les couches physiques IEEE802.11g [7] offrent des débits allant de 6 à 54 Mb/s, plusieurs travaux ont été réalisés afin d'élaborer des mécanismes qui permettent d'adapter le débit. Ces travaux (qui seront détaillés et commentés ultérieurement) consistent à choisir le bon débit physique

correspondant à un état de canal donné pour une bonne sélection du débit afin d'éviter les problèmes liés aux connexions WiFi.

Nous présenterons dans cette partie les problèmes les plus importants qui devraient être pris en considération et qui sont à l'origine de la conception d'un tel mécanisme d'adaptation du débit :

- Une variation du canal suite à une erreur de transmission qui engendre des retransmissions multiples ou même une coupure de la transmission.
- La sensibilité du canal aux interférences dues au bruit, au phénomène perturbateur, au bruit additif et aléatoire, au bruit électromagnétique, à l'effet doppler, à un obstacle accidentel et aux phénomènes naturels.
- La durée d'émission, qui a un impact sur l'autonomie des terminaux puisque en présence d'erreur la durée de communication est prolongée et l'énergie consommée devient plus importante.
- La mobilité réduite qui conduit à un changement des distances entre les stations, d'où l'utilité des protocoles appropriés de gestion de mobilité lors d'une connexion WiFi.

Selon la qualité instantanée du canal, un changement de débit est nécessaire pour réussir une bonne transmission. Ce changement a pour objectif d'obtenir de meilleures performances de communication.

1. Estimation de la qualité du canal et la sélection du débit

Tout d'abord nous rappelons que la norme IEEE802.11 [2, 3, 7], avec ses diverses versions a/b/g, permet l'utilisation de plusieurs débits physiques allant de 1Mbit/s jusqu'à 54Mbit/s dans le cas du 802.11g. L'adaptation du lien est un processus de commutation ou de choix dynamique entre les différents débits physiques de données correspondant à l'état instantané du canal. Elle vise à sélectionner le débit idéal correspondant à un état de canal donné. Ce meilleur débit peut être plus grand ou encore plus petit que l'actuel. Le point le plus important est que ce nouveau débit choisi correspond au mieux aux conditions instantanées du medium de communication. Il existe deux critères pour bien évaluer cette adaptation : la première est l'estimation de la qualité du canal, ensuite en deuxième lieu une bonne sélection du débit adéquat.

Une estimation implique la mesure de la variation en fonction du temps de l'état du canal afin de considérer une prédiction de sa qualité instantanée. Ceci engendre un choix des paramètres indicateurs de l'état du canal qui peuvent être composés du taux du rapport signal sur bruit noté ***SNR*** (*Signal to Noise Ratio*), du taux d'erreur binaire, noté ***BER*** (*Bit Error Rate*), et de la mesure de la puissance du signal reçu notée ***RSSI*** (*Received Signal Strength Indicator*). Ces paramètres physiques expriment des mesures instantanées exploitées par la carte 802.11 après l'achèvement de la dernière transmission sur le canal. Ces retenues estiment la qualité actuelle du support de communication, et par ailleurs, peuvent prédire celle des prochaines transmissions.

Au sujet de la seconde variante, concernant la sélection du débit, elle consiste à exploiter les prédictions de la qualité du canal pour choisir une meilleure valeur du débit physique. En effet un choix optimal minimiserait les retransmissions et le taux de pertes. Nous définissons ainsi le temps moyen de transfert d'un message. Toute erreur d'estimation de la qualité de canal engendrerait une dégradation des performances du mécanisme de contrôle de débit. Des évaluations imprécises, résultantes d'un mauvais choix des indicateurs de l'état du canal, engendrent des jugements non adaptés aux conditions instantanées du canal. Ainsi, il est avantageux d'utiliser une meilleure information disponible quand une évaluation se produit. Cette meilleure information est alors déterminée par le récepteur. Il est également important que les évaluations soient déployées instantanément une fois produites et avant qu'elles ne deviennent invalides. De plus, il est plus rentable de réduire au minimum le délai entre le moment de l'évaluation et le temps de transmission du paquet.

A ce niveau, lorsque nous concevrons un schéma ou un plan d'adaptation du lien, il y a deux questions fondamentales qui se posent concernant le débit physique de transmission : comment choisir le débit adéquat et suivant quels critères ?

2. Synthèse sur le choix du débit approprié

La capacité de supporter de multiples taux permet aux stations de sélectionner le débit de transmission approprié selon la qualité de service exigée et selon les conditions instantanées du canal afin d'améliorer la performance globale du système de communication.

Si, par exemple, une station veut communiquer avec une autre très éloignée, alors la valeur du rapport Signal/Bruit (*SNR*) observée dans le récepteur doit être très basse si le débit employé est important. Dans une telle situation, l'utilisation d'un débit plus petit est exigée pour établir

une meilleure communication. Cependant, si la qualité de canal est suffisamment bonne, alors il est souhaitable de transmettre en utilisant une plus grande valeur de débit. Ceci est indispensable pour mieux supporter les services et maximiser les performances (le débit moyen du système ainsi que le taux d'utilisation du canal radio). Par contre, si le débit choisi est excessivement élevé, des retransmissions supplémentaires seront produites et causeront une détérioration du débit moyen observé en sortie ou même une coupure totale de communication. En résumé, le choix d'un débit trop conservateur engendre aussi une dégradation des performances du système. Par conséquent une adaptation du lien dynamique est obligatoire pour réaliser les meilleures performances du système de communication possible.

De ce fait, lors d'une transmission, une station émettrice peut augmenter ou diminuer la valeur de son débit physique en utilisant une des deux approches suivantes :

a. A l'aide d'une estimation exacte du canal, la station émettrice connaît l'instant où les conditions du canal sont améliorées pour s'accommoder à un débit plus important, et adapter son taux de transmission en conséquence. Cependant, ce type d'approche [21, 22] exige des modifications et un effort de mise en œuvre supplémentaire au standard 802.11. D'autres travaux de recherche [23] ont présenté un plan d'adaptation du lien intéressant basé sur l'estimation du canal sans exiger de changements au standard en utilisant des mesures du *RSSI* ainsi que du nombre de trames retransmises. Toutefois, depuis que ce plan opère sous l'hypothèse que tous les échecs de la transmission sont dus aux erreurs du canal, il ne fonctionne plus efficacement dans un environnement multiutilisateur. Dans ce type d'environnement, plusieurs transmissions peuvent échouer en raison des collisions produites et non seulement à cause d'une mauvaise qualité du support de communication.

b. Une façon alternative pour exécuter l'adaptation du lien est que la station émettrice prend la décision de l'adaptation du débit en se basant uniquement sur l'information retournée par le récepteur. Dans un réseau *WLAN* de type 802.11, un acquittement (*ACK*) est envoyé par le récepteur après la récupération des données avec succès. C'est seulement après avoir reçu correctement une trame *ACK* que l'émetteur annonce un succès de transmission des données correspondantes. D'autre part, si un *ACK* reçu est erroné ou si aucun *ACK* n'est reçu, l'émetteur préjuge de l'échec de transmission des données correspondantes et procède à la réduction du débit. En outre, la station émettrice peut augmenter son débit de transmission pour s'adapter aux conditions du canal après la réception d'un nombre précis d'*ACK* positifs

et consécutifs. Ces approches [24, 25] n'exigent pas de changements non conformes aux standard 802.11, et il est donc très facile de les déployer avec les cartes réseaux 802.11 existantes.

D'autres travaux liés à cette problématique de recherche seront présentés en détails et commentés dans le chapitre 3. De plus une nouvelle approche sera exposée et comparée avec celles actuellement déployées par la norme. Les résultats de simulation de cette dernière seront également présentés afin de montrer l'apport de notre nouvelle technique par rapport à celles présentement utilisées. Enfin une étude portant sur le choix des valeurs des paramètres du nouveau mécanisme sera exposée et interprétée.

III. Différenciation de services dans les réseaux 802.11

Offrir à des hôtes mobiles un accès ubiquitaire à l'Internet devient de plus en plus important en raison de l'émergence de nouvelles applications, telles que l'accès mobile à l'information, les communications multimédia temps-réel, les jeux en réseau, etc. Il est donc nécessaire d'avoir des schémas de différenciation non seulement dans tous les domaines qui constituent le réseau *Internet*, mais aussi dans les réseaux d'accès où se situent souvent les goulots d'étranglement. D'où la nécessité de gérer la qualité de service dans les réseaux d'accès 802.11 qui de nos jours sont très étudiés. Dans [26] les auteurs ont montré que les mécanismes de la couche *MAC* du standard IEEE 802.11 sont considérés insuffisants pour accomplir une qualité raisonnable dans des scénarii où un énorme trafic de type background charge le réseau de transmission. Donc, des améliorations pour la gestion de *QoS* dans ces types de réseaux, sont largement étudiées et évaluées.

Avant de présenter les différentes approches proposées pour améliorer les services offerts par un réseau, il serait plus intéressant d'étudier en premier lieu les différents critères qui influencent la *QoS* sur les réseaux d'une façon générale. La caractérisation de la qualité de service dans les réseaux est généralement exprimée par les facteurs suivants :

- **Le délai :** c'est le temps écoulé entre l'envoi d'un paquet par l'émetteur et sa réception par le destinataire. Le délai tient compte du temps de propagation le long du chemin parcouru par les paquets et du temps de transmission induit par la mise en file d'attente des paquets dans les systèmes intermédiaires (les routeurs par exemple). La plupart des applications, et surtout les

applications temps réel, sont très sensibles aux valeurs élevées de délais et exigent un délai limité pour qu'elles puissent fonctionner correctement.

- **La gigue :** la gigue est la variation du délai de bout en bout. Certaines applications (audio et vidéo) subissent une distorsion du signal avec de grandes valeurs de gigue.
- **Le débit :** c'est en fait le taux de transfert maximum pouvant être maintenu entre deux points terminaux (un émetteur et un récepteur). Ce facteur est influencé non seulement par les capacités physiques des liens, mais aussi par les autres flux partageant ces liens.
- **La fiabilité :** c'est le taux moyen d'erreur d'une liaison. Actuellement les supports de transmission deviennent de plus en plus fiables et ce taux devient de plus en plus négligeable par rapport aux pertes dues aux situations de congestion des réseaux.
- **Le taux de perte :** les pertes sont généralement dues, soit à la non-fiabilité des liens physiques, soit aux mécanismes de gestion des files d'attente dans les routeurs intermédiaires qui se trouvent souvent obligés d'éliminer des paquets pour faire face à des situations de congestion. Ce facteur est très important. En effet, certaines applications ne tolèrent pas des valeurs élevées de ce facteur.

Plusieurs techniques ont été proposées pour introduire la différentiation de la qualité de service dans 802.11b. En effet, des chercheurs [27] ont proposé de travailler sur les *IFSs* (*les temps d'inter-frame*) pour différencier les classes de service : une station avec un trafic prioritaire utilise des *IFS*s plus courts qu'une station ayant un trafic de basse priorité. Utiliser des *IFS*s plus courts revient à réduire la durée d'attente et par là même à augmenter la probabilité d'accès au support. Il est à noter que toutes ces techniques présentent des compromis du genre ratio de bande passante utilisée, degrés de différentiation entre les classes de priorités hautes et faibles qu'il faut prendre en compte. Ensuite, d'autres travaux [28] ont été réalisés sur la modification des *CW* (*fenêtres de contention*) respectives à chaque classe de service appropriée CW_{min} et CW_{max} en se basant sur des modèles mathématiques.

Pratiquement toutes ces approches proposent des solutions au niveau lien où il faut modifier la couche *MAC* pour les intégrer, ce qui rend assez difficile leur faisabilité dans le cadre réel et opérationnel actuel. Dans des conditions idéales (toutes les stations sont en vue directe, distance raisonnable entre le point d'accès et les stations, pas d'obstacles, etc.), un réseau 802.11b offre un débit efficace de l'ordre de 5 Mb/s. La bande passante restante est utilisée par les signaux de contrôle de *CSMA/CA*. Pire encore, sachant que les performances de *CSMA/CA* dépendent aussi

de la charge imposée par les clients du BSS, nous pourrons nous retrouver dans la pratique face à une situation de saturation du canal (*CSMA/CA* passe son temps à "backoffer") entraînant qu'aucune transmission ne soit possible : il faut alors éliminer de l'information pour éviter ce genre de scénario.

1. Couplage de DiffServ avec le schéma de QoS dans les WLANs

Définie par l'***IETF*** (*Internet Engineering Task Force*), ***DiffServ*** [29] se base sur la différentiation de Services en introduisant une étiquette dans chaque paquet qui détermine le traitement que celui-ci obtiendra du réseau. Le principe de ce dernier est d'étiqueter les différents flux des classes de service, par l'utilisation d'une vignette ajoutée dans l'entête *IP* des paquets. *DiffServ* [30] agit au niveau des agrégats de trafic en segmentant le trafic total en plusieurs classes (classes de services) dont le traitement est ensuite différentié dans les équipements d'interconnexion.

DiffServ définit 3 types de classes de services, l'administrateur d'un réseau ayant la liberté de n'implémenter que les classes qu'il juge nécessaires. Les différents types de classes sont les suivants :

- ✓ **EF (Expedited Forwarding)** : c'est la classe d'excellence. Les paquets marqués *EF* doivent être acheminés avec un délai, une gigue et un taux de perte minimum. Des moyens techniques (contrôle d'accès, sur réservation, etc.) doivent être mis en œuvre pour assurer le bon fonctionnement de celui-ci.
- ✓ **AF (Assured Forwarding)** : quatre classes *AF* ont été définies, chacune d'elles comporte 3 sous classes. Les paquets sont marqués AF_{xy} tel que $x \in [1, 4]$ est le numéro de la classe *AF* et $y \in [1, 3]$ est la précédence à l'écartement. Les paquets d'une même classe empruntent toujours la même file d'attente pour éviter le déséquencement. La précédence à l'écartement définit la priorité relative de rejet (ou un traitement spécifique) des paquets en cas de congestion.
- ✓ **BE (Best Effort)** : c'est l'équivalent de l'Internet actuel où aucun traitement particulier ne vient améliorer le relayage des paquets appartenant à cette classe.

Au total nous dénombrons 14 comportements possibles, notés ***PHB***s (*Per Hop Behaviour*). Un *PHB* est la manière avec laquelle un routeur traitera les paquets entrants (c'est à dire la mise

en file d'attente plus le traitement en cas de congestion). Un *PHB* est déterminé à partir des ***DSCPs*** (*Differentiated Service Code Point*) codés directement dans le champ ***TOS*** (*Type of Servive*) du paquet *IP*. L'interopérabilité entre le champ *DSCP* et le schéma de *QoS* dans IEEE802.11e consiste à faire correspondre les niveaux de priorité entre *DiffServ* et 802.11e. Deux problèmes se posent pour ce couplage. Le premier consiste à déterminer les champs dans la trame *MAC* 802.11e qui reflètent la différence de service définie par le *DSCP* de *DiffServ*. Le second est de définir la compatibilité et les similarités entre les services *DiffServ* et les services du standard 802.11e. Il est important aussi de définir à quel instant le marquage des trames pour la *QoS* est effectué.

Dans *DiffServ*, le champ *DSCP* indique non seulement la nature du trafic (important, moins important, pas important) reflétant le niveau de priorité à la perte, mais aussi la classe de services à laquelle appartient le paquet. Par conséquent, il nous faut aussi dans le réseau 802.11 des indicateurs pour identifier la nature du trafic et le niveau de priorité de chaque trame. Dans 802.11e, une interface de programmation applicative spécifique a été définie pour demander la qualité de service à partir de la couche *MAC*. Elle permet d'identifier si la trame requiert une qualité de service ou non (voir *Figure* 2.1) en indiquant la valeur appropriée sur quatre bits (1000) dans le champ « *Subtype* ». Cette valeur ne définit pas le niveau de priorité de la trame qui est essentiel pour la mise en correspondance dans les différentes files d'attente. Nous utilisons trois bits dans le champ *QoS* Control pour spécifier le trafic à travers le sous-champ ***TID*** (*Traffic Identifier*).

Considérons le sens ascendant où le trafic quitte le réseau d'accès pour aller vers le réseau Internet (appelé aussi *backbone*). Dans ce cas, il est nécessaire de marquer les trames avant de les envoyer. Le nombre limité de bits (3) pour identifier les catégories de trafic nous impose d'avoir seulement 3 classes de services avec 3 niveaux de priorité dans chacune pour le comportement *AF*. Deux combinaisons sont prises pour les trafics *Expedited Forwarding* et *Best Effort*. Ceci rentre bien dans la conception du trafic que nous avons définie (trafic important, moins important et pas important). Le troisième niveau de priorité de toutes les classes de service *AF* est codé avec "000" et le trafic concerné est considéré comme du *Best Effort*.

Le couplage défini permet d'harmoniser la *QoS* dans les réseaux d'accès et la *QoS* dans le *backbone*. La notion de précédence ou la priorité à la perte est ainsi identique pour l'architecture

DiffServ et 802.11. En d'autres termes, le couplage permet à une classe de service d'observer le même comportement dans les deux niveaux de réseaux. Ceci améliore la gestion de bout en bout de la qualité de service des classes de services. Ainsi, pour le réseau global constitué de réseaux d'accès et de plusieurs domaines *DiffServ*, la qualité de service de bout en bout est déterminée par la partie du réseau qui offre les performances les moins bonnes.

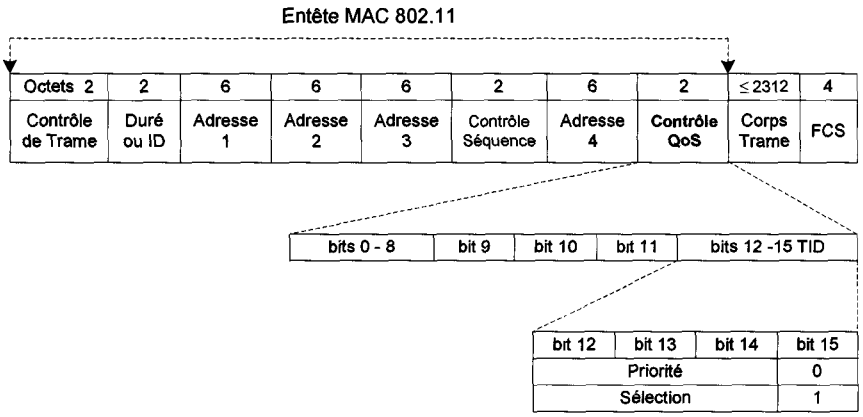

Figure 2.1 : Champ QoS dans la trame MAC 802.11

2. Améliorations du Distributed Coordination Function (DCF)

A. DCF : Critiques

La fonction de coordination distribuée *DCF* permet un accès au médium avec contention. Il s'agit donc d'un service de type *"au mieux"* (*Best Effort*) qui ne peut supporter aucun autre service exigeant de la *QoS*.

En effet la méthode *DCF* pose le problème d'un coût d'accès qui croit proportionnellement avec le nombre de stations. Elle est basée sur le principe *CSMA/CA* précédemment expliqué. Si une seule station émet, le temps de transmission sera (pour un réseau 802.11b à 11Mbit/s en négligeant les temps de propagation) comme illustré dans l'*Equation* 1 suivante :

$$eq.1 \qquad T_{single} = t_{pr} + t_{tr} + SIFS + t_{ACK} + DIFS$$

où t_{pr} est le temps de préambule (144μs), t_{tr} est le temps de transmission de la trame (taille/débit), *SIFS*=10μs, t_{ACK} est le temps de transmission de l'acquittement (210μs), et *DIFS*=30μs. La proportion r de la bande passante effectivement disponible sera comme l'*Equation* 2 qui suit :

$$eq.2 \qquad r = \frac{t_{tr}}{T_{single}}$$

Pour une taille de trame de 1500 octets de données (1534 octets au total) la proportion r est égale à $\frac{11.1ms}{15.1ms} = 0.735$.

Une seule station émettant sur un canal radio à 11Mbit/s aura donc une bande passante effective de 8.08 Mb/s. Par contre, si plusieurs stations cherchent à accéder au canal, une station peut trouver le canal occupé, ou entrer en collision avec la transmission d'une autre station. Dans de tels cas, la station attend pendant un intervalle de temps aléatoire distribué uniformément dans [0, *CW*-1] × *slot-time*, puis retente d'émettre. L'unité de temps réseau utilisée est le *slot-time* = 50μs. Chaque fois qu'une station choisit une durée d'attente et rentre ensuite en collision, elle double la valeur de *CW* jusqu'à un maximum de CW_{max}. En pratique, la valeur de *CW* varie entre CW_{min} = 4 et CW_{max} = 256. Les différentes valeurs des paramètres indiqués sont données pour la couche physique *DSSS*. Donc, pour *m* stations, le délai augmente à cause des collisions, et donc l'efficacité se dégrade comme le montre l'*Equation* 3 suivante :

$$eq.3 \qquad T_{multiple\ (m)} = T_{single} + w_{cw(m)}$$

où $w_{CW(m)}$ est la longueur moyenne de la fenêtre de contention pour *m* stations. La proportion de bande passante effectivement utilisable dépendra donc aussi du nombre de stations comme dans l'*Equation* 4 suivante :

$$eq.4 \qquad r_{(m)} = \frac{t_{tr}}{T_{multiple\ (m)}}$$

Nous en concluons donc que pour fournir de la qualité de service sur un lien 802.11, le nombre d'hôtes autorisés à utiliser le canal doit être limité. La méthode *DCF* est conçue pour donner aux hôtes mobiles une probabilité égale d'accès au lien. Pour offrir des performances différentes aux sources de trafic au niveau des hôtes, il faut les rendre configurables, afin que les sources de faible priorité aient une allocation de ressources différente de celles dont bénéficient les sources de haute priorité.

Dans [31], les auteurs démontrent l'insuffisance du *DCF* pour la transmission des paquets multimédias. En effet, *DCF* peut seulement supporter le service *Best Effort* et pas n'importe quel autre type de garanties *QoS*, comme ceux demandés par les services temps-bornés tels que la voix sur *IP* (*VoIP*) ou les conférences de l'audio/video, qui exigent une bande passante spécifiée, un délai et une instabilité connue, mais qui peuvent tolérer des pertes. Cependant, dans le mode

DCF, toutes les stations d'une même *BSS* utilisent les ressources et accèdent au canal avec les mêmes priorités. Il n'y a aucun mécanisme de différenciation pour garantir une bande passante, un délai du paquet et une stabilité précise pour les stations prioritaires porteuses de flux multimédias. Dans [31], les auteurs ont démontré par une simple simulation (voir *Figure* 2.2), qu'il n'y a aucun moyen de garantir les exigences *QoS* pour des trafics prioritaires (audio, vidéo...), à moins que le contrôle de l'admission soit utilisé.

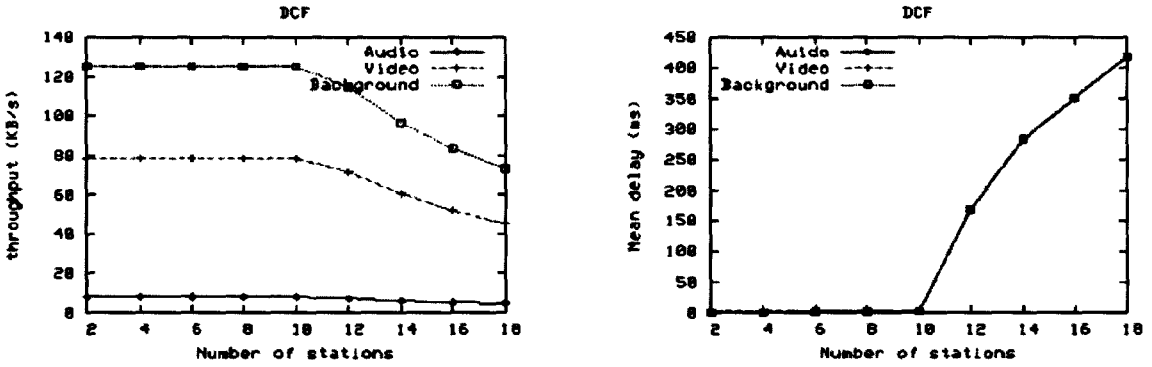

Figure 2.2 : Evolution du débit et du délai en mode DCF [31]

Nous constatons, à partir de la figure ci-dessus, qu'aucune garantie n'est allouée pour les flux multimédias par rapport à ceux du Background. Le débit ainsi que le délai moyen sont similaires pour les différents types de trames et aucune classification n'est planifiée.

B. Hybrid Coordination Function (HCF)

Afin de supporter *DiffServ* dans les réseaux sans fil, le standard 802.11e [27] a défini un nouveau mécanisme appelé *HCF*. Ce dernier, est composé de deux modes d'accès : ***EDCF*** (*Enhancement DCF*) et ***CHCF*** (*Controlled HCF*). La nouveauté du *HCF* se résume dans l'introduction de quatre catégories d'accès ***ACs*** (*Access Categories*) et huit types de trafics ***TSs*** (*Traffic Streams*) dans la file. Lorsque la trame arrive au niveau *MAC*, elle est étiquetée par un type de trafic *TID* bien spécifié suivant la *QoS* demandée (une valeur entre 0 et 15). Les trames admettant une valeur du *TID* entre 0 et 7 seront mappées dans les quatre files *ACs* en s'accordant à l'accès *EDCF* [28]. Pour celles qui admettent une valeur du *TID* qui varie entre 8 et 15, elles seront mappées dans les huit files *TSs* suivant la méthode *HCF* pour les canaux contrôlés.

C. *Enhanced Distributed Function (EDCF)*

Avec cette amélioration au niveau *QoS*, les stations ont 4 catégories *ACs* afin de supporter 8 priorités d'utilisateurs ***UPs*** (*User Priorities*). Par conséquent, plusieurs *UPs* peuvent se trouver dans la même file *AC*. Avec ce choix de partage (4 *AC* et 8 *UP*), le nombre d'*ACs* est moins important que celui des *UPs*, ce qui entraîne une diminution du champ dédié à la *QoS*, et donc également de la taille de l'entête *MAC* (voir *Tableau* 2.1). Chaque file *AC* travaille comme étant une station indépendante des autres *ACs*, et admet ses propres paramètres de *backoff*. La procédure d'allocation de priorité du *EDCF* [27, 31] est illustrée dans la *Figure* 2.3.

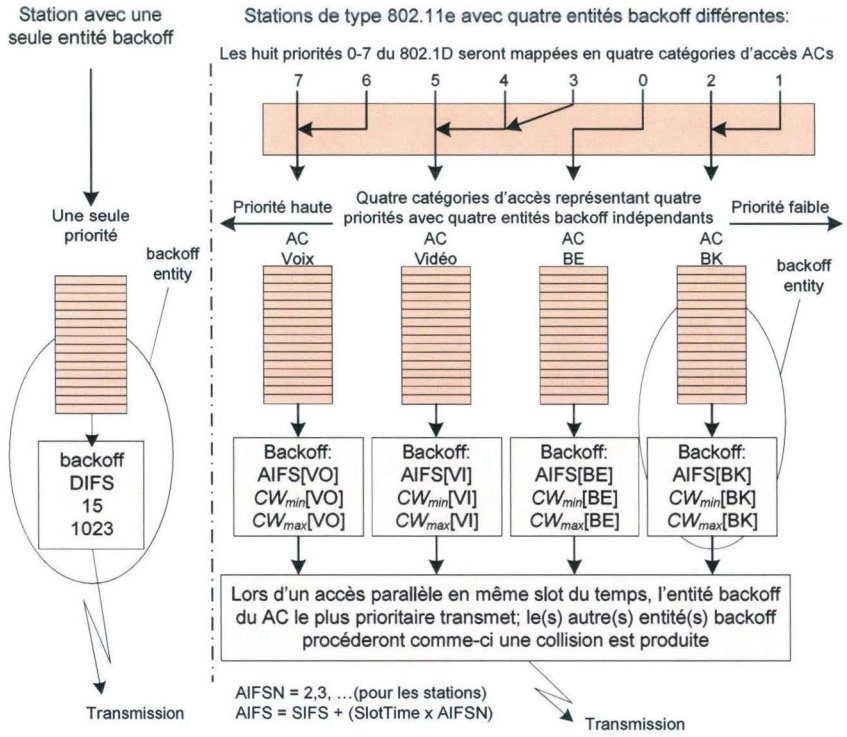

Figure 2.3 : Différence entre une station 802.11 classique et une station 802.11e

802.1D UP (User Priority)	***802.1D Designation***	***802.11e AC (Access Category)***	***Service type***
1	Background (BK)	0	Background
2	Not defined	0	Background
0	Best Effort (BE)	1	Best Effort
3	Excellent Effort (EE)	1	Video Probe
4	Controlled load (CL)	2	Video
5	VI (Video *<100 ms* latency and jitter)	2	Video
6	VO (Voice *<10 ms* latency and jitter)	3	Voice
7	Network Control (NC)	3	Voice

Tableau 2.1 : Mappage entre les Priorités Utilisateurs UP et les Catégories d'Accès AC

Dans l'*EDCF* deux méthodes sont introduites pour supporter la différenciation de service. La première est l'utilisation d'espaces différents entre trames *IFSs* pour les différents *ACs*. Un nouveau type d'*IFS* est né : ***AIFS*** (*Arbitration Inter-Frame Space*) et utilisé dans *EDCF* (au lieu du *DIFS* dans *DCF*). L'*AIFS* d'une catégorie *AC* est déterminé par :

$$eq.5 \qquad AIFS[AC] = AIFSN[AC] \times SlotTime + SIFS$$

où la valeur du ***AIFSN*** (*AIFS Number*) varie suivant la catégorie *AC*.

Lorsque $AIFSN[AC] = 1$, la file de priorité correspondante aura une valeur du $AIFS[AC]$ égale au *PIFS*. C'est pour cette raison que la valeur d'$AIFSN[AC]$ commence à 2 comme illustré dans la *Figure* 2.3. La file de priorité la plus haute aura un $AIFS[AC]$ de valeur égale au *DIFS*. Les autres files moins prioritaires auront un $AIFS[AC]$ plus grand à respecter avant d'accéder au canal. Lorsqu'une trame arrive dans une file d'un *AC* vide et que le canal reste libre pendant $AIFS[AC] + SlotTime$, elle est transmise immédiatement. Dans le cas contraire (c.-à-d. canal occupé), chaque trame qui arrive dans une des files *ACs* doit attendre la libération du canal puis elle diffère sa transmission pendant $AIFS[AC] + SlotTime$. Donc, la file qui admet l'$AIFS[AC]$ le plus petit est en effet celle qui admet la priorité la plus haute.

La deuxième méthode introduite dans *EDCF*, comporte l'allocation de différentes tailles de fenêtre *CW* pour les différents *ACs*. La plus petite taille de fenêtre *CW* est assignée à la file *AC* la plus prioritaire pour s'assurer, dans la plupart des cas, que les files de priorités hautes occuperont le canal devant ceux de priorités basses. Avec ces nouveaux paramètres, l'*EDCF* est supposé améliorer les performances du *DCF* dans des conditions de congestion.

Les valeurs par défaut du $AIFSN[AC]$, $CW_{min}[AC]$ et $CW_{max}[AC]$ sont annoncées par le point d'accès ***QAP*** (*QoS AP*) dans la trame *beacon*, et le standard 802.11e alloue au *QAP* la possibilité d'adapter ces paramètres dynamiquement aux conditions du réseau. Cependant, la manière d'adapter ces paramètres aux conditions du canal n'a pas été définie par le standard et ouvre un grand axe de recherche. Dans le chapitre 5, nous allons étudier la dernière amélioration apportée à la norme 802.11 appelée ***EDCA*** (*Enhanced Distributed Channel Access*) ainsi que ses faiblesses. Ensuite, nous exposerons notre nouvelle approche de différentiation de service basée sur ce dernier et améliorant considérablement l'accès ainsi que le schéma de transmission des flux multimédias dans les réseaux 802.11.

IV. Handover dans le 802.11

Un troisième volet de recherche, concernant toujours l'amélioration de la *QoS*, est abordé dans cette thèse. Nous nous intéressons à un problème important des réseaux 802.11 lorsque les nœuds sont des équipements mobiles (le cas d'une couverture WiFi dans les universités, les cybercafés, les aéroports, etc.). Ce thème reste parmi les moins examinés et étudiés, et il existe peu de travaux de recherche qui lui sont dédiés dans la littérature. Par ailleurs, ce mécanisme est devenu un vrai obstacle de connexion face aux énormes demandes des industriels d'étendre la couverture réseau de leurs sites. C'est souvent le cas lorsqu'un utilisateur veut rester connecté au réseau tout en se déplaçant dans la zone de couverture garanti par les *APs*. Dans un réseau 802.11 en mode infrastructure, les stations sont affiliées à un point d'accès qui fait partie de cette infrastructure. Dans un contexte où les stations sont mobiles, il est parfois nécessaire qu'une station change de cellule, ce qui signifie qu'elle doit s'affilier à un autre point d'accès. Dans la norme 802.11 [1], nous constatons l'absence d'une réelle gestion du Handover comme celle du réseau GSM. Nous trouvons uniquement les conditions à partir desquelles une station va décider de changer de point d'accès d'affiliation ainsi que les différentes étapes qui vont lui permettre de choisir et de s'affilier à un nouveau point d'accès.

Le mécanisme du Handover, tel qu'il a été présenté par la norme, peut être divisé en 5 étapes comme le montre la *Figure* 2.4 : phase de détection, phase de désauthentification, phase de découverte ou de scan, phase d'authentification et finalement la phase de réassociation. Les deux dernières phases constituent le Handover (ou appelé aussi *Handoff*) effectif. Comme l'illustre la *Figure* 2.4, les deux premières phases sont liées et souvent réduites à une seule phase appelée phase de détection. De même pour les deux dernières qui sont réunies ensemble pour former la phase du *handoff* effectif.

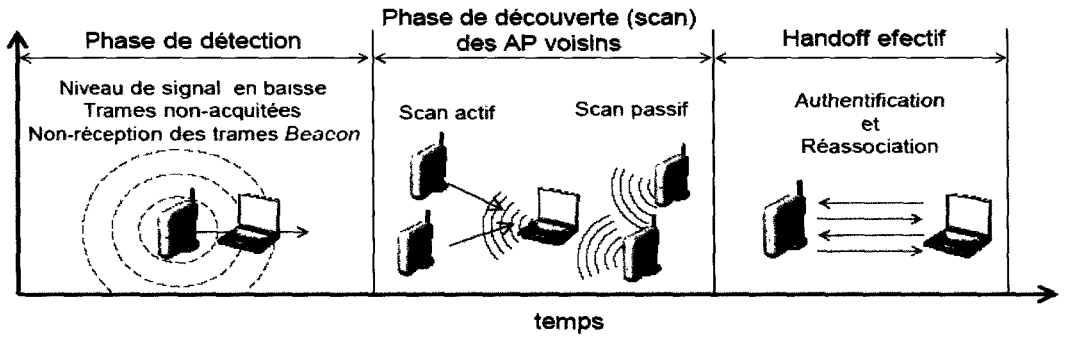

Figure 2.4 : Les phases du Handover dans 802.11

1. Phase de détection/déclenchement

Le changement de cellule est en principe décidé suite à une détérioration de la qualité de la liaison radio définie grâce à la puissance du signal, au rapport signal sur bruit (*SNR*) ou à une perte de connexion avec l'*AP*. Pour chaque trame reçue, la carte réseau est capable de mesurer la puissance du signal reçu à partir de la valeur de son indicateur *RSSI* utilisé pour ce fait. Il doit être mesuré par le circuit sur une carte réseau sans fil appelé ***NIC*** (*Network Interface Card*) selon la norme 802.11. La valeur de ce dernier est une valeur numérique entière pouvant aller de 0 à 255 (une valeur sur 1-octet), avec une valeur de 0 pour un signal très faible ou non existant et 255 pour un signal très satisfaisant. À partir de ces paramètres, la station repère les dommages sur la qualité du support et entamera la procédure du Handover.

2. Phase de Désauthentification

Une trame de gestion appelée *''Désauthentication''* est envoyée, soit par la station avant de changer de canal de communication, ce qui permet au point d'accès de mettre sa table d'affiliation à jour, soit par le point d'accès pour demander à une station de quitter la cellule. En général, cette trame est générée par la station mobile puisqu'elle détecte plus rapidement la détérioration de la qualité du medium.

3. Phase de recherche d'un nouveau point d'accès

Une fois que la décision concernant le Handover est prise, la station doit chercher un nouveau point d'accès *AP*. Ce dernier doit offrir une qualité de lien plus performante dont la puissance du signal reçu est meilleure selon les mêmes critères que le point d'accès précédent. La qualité du lien est un indicateur composite qui varie en fonction de la puissance du signal reçu, corrélée avec le taux d'erreurs et de retransmissions. Deux méthodes sont définies par la norme pour rechercher de potentiels points d'accès : le scan Actif et le scan Passif

A. Scan Actif

Dans ce mode la station va prendre l'initiative de rechercher un nouveau point d'accès à rejoindre. La station cherche à joindre de potentiels *APs* voisins en envoyant des trames sondes appelées *''Probe Request''*. Si un point d'accès reçoit ce type de trame, il va répondre avec une autre trame de gestion appelée *''Probe Response''* (voir la *Figure* 2.5).

Le temps d'attente sur un canal est contrôlé par deux *timers*. Le premier est le temps minimum d'écoute sur un canal, *MinChannelTime*. Si pendant ce temps la station ne détecte aucune activité sur le canal, celui-ci est déclaré inactif et elle balaiera le canal suivant. Par contre, si elle détecte une activité quelconque sur le canal, la station est obligée d'attendre les éventuelles trames de réponse pendant un temps plus long, désigné par *MaxChannelTime*. Grâce à la mesure du signal effectuée à la réception de cette trame, la station jugera si ce point d'accès potentiel est un bon candidat ou non. C'est après comparaison des résultats de tous les *''Probe Response''* reçus que la station entamera la phase d'affiliation à un nouveau point d'accès. Bien évidemment, celui des *APs* visités offrant la meilleure qualité enregistrée, sera choisi pour achever le Handover actuel.

B. Scan Passif

La station n'émet plus et change de canal à intervalles réguliers selon le paramètre *ChannelTimer* (voir la *Figure* 2.6). Il est nécessaire de rester sur chaque canal de la liste des canaux *ChannelList* pour un délai supérieur au délai *inter-beacon* des *APs*. Tous les canaux sont scrutés, puis nous retrouvons la phase de *Probe* (utilisée dans le scan Actif) mais seulement pour l'*AP* choisi. Le groupe 802.11k [32] travaille sur l'amélioration du choix du prochain l'*AP* en tenant compte de la charge du réseau et non plus seulement de la puissance du signal.

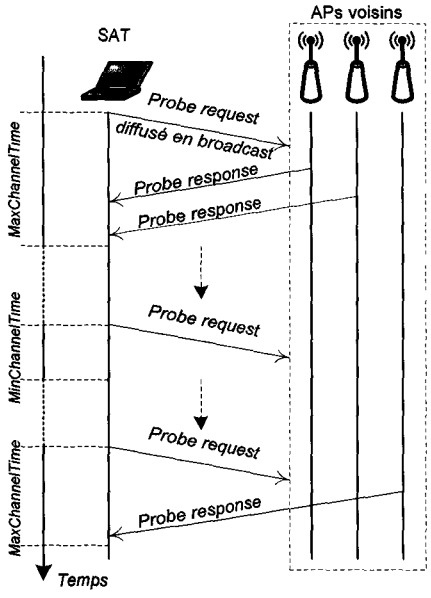

Figure 2.5 : Scan actif

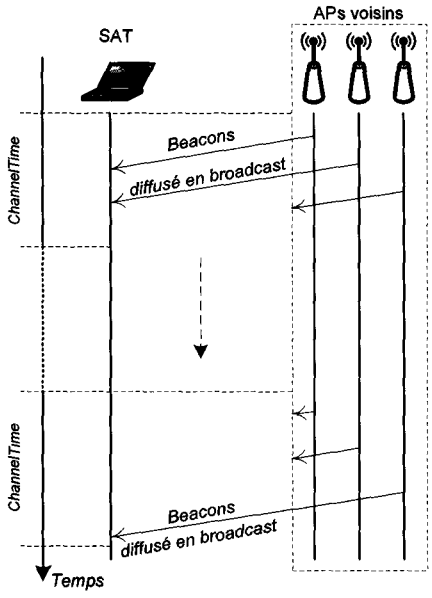

Figure 2.6 : Scan passif

4. Phase d'Authentification

Une fois que la phase de choix du prochain *AP* candidat ayant les meilleures caractéristiques de transmission est effectuée, la station débute la procédure d'authentification. Les versions initiales du protocole 802.11 spécifient deux méthodes : *Open System Authentication* et *Shared Key Authentication*. La première est très simple : la station envoie une première trame et le point d'accès lui répond en lui indiquant l'acceptation ou le rejet. Le mécanisme de contrôle d'accès se base que sur l'adresse *MAC* de la station présentant une demande d'authentification.

La deuxième méthode suppose l'existence d'un secret partagé entre la station et le point d'accès. Ce secret est représenté par une clé ***WEP*** (*Wired Equivallent Privacy*), utilisée aussi pour le chiffrement éventuel de trames de données. Un échange supplémentaire de deux trames (*défi – réponse*) est rajouté lors de la phase d'authentification, dans lequel la station doit déchiffrer un texte fourni par le point d'accès. La méthode de *Shared-Key Authentication* exige donc l'échange de quatre messages:

1. Le client demande l'authentification à l'*AP* en envoyant une trame *''Challenge-Request''*.
2. L'*AP* envoie un nombre aléatoire au client à travers une trame *''Challenge-Response''*.
3. Le client signe ce nombre aléatoire en utilisant le cryptage *WEP* et envoie de nouveau une trame de réponse à l'*AP*.
4. L'*AP* vérifie que le nombre aléatoire a été signé par la bonne clé ou non, en calculant la signature lui-même et en la comparant avec la valeur reçue. Une fois que la clé a été vérifiée par l'*AP*, il authentifie le client en lui envoyant un message d'approbation.

Une étude [33] a montré que le contrôle d'accès basé sur les adresses *MAC* des stations peut être facilement contourné avec des outils logiciels qui permettent de reconfigurer l'adresse *MAC* des interfaces sans fil, ce qui montre les limites de cette méthode.

5. Phase d'Association ou Réassociation

Une fois la station authentifiée, le processus d'association se déclenche. Celui-ci consiste en un échange d'informations sur la cellule : l'*ESSID* et les vitesses de transmission supportées. Seulement après le processus d'association, la station sera affiliée réellement avec le *BSS* et pourra transmettre et recevoir des trames de données. Ce processus est de type *« hand-check »* requête puis réponse à la requête. Il est réalisé par l'échange d'une trame *''Association Request''*

envoyée par la station qui souhaite s'affilier ou s'associer. Le point d'accès *AP* actuel répondra lui-même à cette demande par l'envoi d'une trame *''Association response''*, qui contient un code statut par lequel il précise si l'association a été acceptée et accomplie ou non. Le processus de réassociation est similaire à celui de l'association initiale. Il comporte deux ou quatre trames d'authentification en fonction de la méthode utilisée (*Open system* ou *Shared-key*) et deux trames d'association appelées *''Reassociation Request''* et *''Reassociation Response''*. La seule différence par rapport à l'association initiale est la spécification de l'ancien point d'accès abandonné par la station dans la trame *''Reassociation Request''*. Cet élément peut être utilisé pour un échange supplémentaire de messages entre les deux points d'accès (l'ancien et le nouveau), conformément aux spécifications d'un document de l'*IETF* - la recommandation 802.11f [6]. Cette recommandation consiste principalement à la spécification d'un nouveau protocole de communication entre les points d'accès appelé *IAPP* (*Inter Access Point Protocol*. Ainsi, le nouvel *AP* envoie à l'ancien un message *''IAPP MOVE Notify''* et la réponse consiste en un message *''IAPP MOVE Response''*. Le but de ce message est l'échange des informations concernant la station mobile, comme par exemple des informations de sécurité permettant une authentification plus rapide au nouveau point d'accès. Une fois l'affiliation terminée, l'identifiant de la station, c'est-à-dire l'adresse *MAC* de la carte réseau, sera contenu dans la table d'affiliation du nouveau point d'accès. Avec cette information, le point d'accès transmettra dans sa cellule (donc sur son interface radio *NIC*) toutes les trames venant de l'infrastructure destinée à cette station. Avec cette nouvelle révision du standard, les données destinées à une station qui vient juste de réaliser un Handover ne seront plus perdues mais plutôt enregistrées et transférées vers le nouvel *AP* pour atteindre la station concernée.

6. Handover en pratique

Comme nous l'avons déjà mentionné, le standard 802.11 ne spécifie pas un mécanisme pour implémenter la phase de détection d'un Handover. De même pour la phase du scan, la norme contient une description générique des algorithmes déployés. Elle ne spécifie ni la manière de choisir l'un ou l'autre ni les valeurs des différents *timers* utilisés par ces algorithmes. En conséquence, nous nous attendons à avoir des comportements différents pendant les phases de détection et du scan, dont les durées peuvent également varier considérablement d'une utilisation à une autre. Toutes ces raisons présentées plus haut, nous poussent à explorer davantage cette

section du 802.11 peu abordée mais très importante lorsque nous parlons de support des trames multimédias nécessitant, en plus du débit, un respect strict des temps de transmission. Cependant, en adoptant le schéma actuel du Handover, il ne sera jamais possible de transférer des flux bornés en temps sur un nœud mobile en mouvement et exécutant des Handovers entre les différents *APs* de la zone de couverture. Dans le Chapitre 6, nous démontrerons ces concepts par une étude détaillée portant sur le temps du Handover et nous présenterons un état de l'art approfondi sur les travaux récents effectués visant la réduction du temps du Handover. Dans le chapitre 7, nous exposerons une nouvelle solution respectant le support de flux à *QoS* exigée ainsi que les résultats acquis par son application. Ces résultats sont obtenus par le biais d'une nouvelle implémentation conçue pour la simulation des Handovers 802.11 ; et seront comparés avec ceux établis par la méthode standard de la norme 802.11.

V. Conclusions

Cet état de l'art montre qu'un certain nombre de problèmes ont été mis en évidence avec l'utilisation des applications multimédias 802.11 dans les réseaux 802.11. Ces travaux assez récents, les plus spécifiques datant de 2003, soulèvent une problématique bien particulière, mais aucun ne propose une solution intégrale pour le support de ces types d'applications dans le cadre des réseaux *WLAN*s de type 802.11. Nous étudierons de façon précise ces approches qui, dans la majorité des cas, apportent des modifications sur les mécanismes et les algorithmes existants pour améliorer les performances. Cependant, ces changements ne seront ni conformes à la norme 802.11 ni approuvés par le standard IEEE.

Dans cette thèse, nous avons mené un travail complet sur les performances du 802.11 face aux applications multimédias qui émergent du réseau Internet. Nous avons tenté de dégager les scénarii de base qui permettent de lister les différents problèmes soulevés et d'y apporter des explications ciblées. Grâce à des simulations, certains des phénomènes observés dans la littérature ont été retrouvés, tandis que d'autres caractéristiques ont été dégagées.

Enfin, contrairement à la majorité des travaux existants, plusieurs nouvelles contributions sur le 802.11 actuel sont présentées dans cette thèse pour le support ainsi que le respect d'une certaine *QoS* exigée. Ce travail a ainsi permis d'une part de connaître les performances réelles du 802.11, et d'autre part d'éclairer les problématiques sous un angle nouveau.

CHAPITRE

3 Mécanisme d'adaptation du débit dans les réseaux WiFi

I. Contexte et objectifs

Le WiFi a prouvé jusqu'à maintenant qu'il permettait un déploiement facile ainsi qu'une mise en place rapide et modulaire d'un réseau dans des conditions où il est impossible de tirer des câbles. Néanmoins, le 802.11 *WLAN* présente aussi des insuffisances introduites dans le chapitre précédent au niveau du choix du débit physique et des algorithmes capables de gérer les états instantanés du canal afin de satisfaire les applications nécessitant une qualité de service assez importante et stricte.

Dans le présent chapitre, nous exposons les principaux algorithmes déployés par la norme 802.11 pour ce fait. Ensuite, nous proposons une nouvelle technique d'adaptation de débit afin d'améliorer la décision sur les conditions instantanées du canal. Elle repose sur d'autres informations et paramètres que le nombre d'acquittements positifs, à savoir le facteur temps d'aller-retour des paquets. Ces nouveaux paramètres rendent la décision sur le choix du débit physique plus rapide et appropriée aux conditions instantanées du support de transmission très sensible.

II. Mécanismes actuels d'adaptation du débit dans le 802.11

En 1997, Kamerman et Monteban [34] ont défini un algorithme d'adaptation de lien physique pour les interfaces sans fil 802.11, nommé ***ARF*** *« Auto Rate Fallback »*. Ensuite, une amélioration de cette approche ***AARF*** *« Adaptive Auto Rate Fallback »* a été réalisée par Manshaei et al. [35] en 2004. Cette nouvelle version avait comme objectif d'obtenir plus de

performances par rapport à l'algorithme existant dans des conditions stables du canal de transmission. D'autres travaux [21, 22, 23, 24, 25], déjà mentionnés dans le chapitre précédent, ont traité le même sujet. Cependant, nous détaillerons seulement le fonctionnement des deux mécanismes d'adaptation de débit : *ARF* et *AARF*, puisqu'ils ont été conçus et déployés jusqu'à présent par les cartes réseaux 802.11.

1. Auto Rate Fallback (ARF)

Auto Rate Fallback [10] fut le premier algorithme de « contrôle de débit » publié et très vite adopté. Il a été conçu pour optimiser le débit physique sur la deuxième génération des *WLANs* (les versions 802.11a et g qui admettent plusieurs sauts de débits physiques). Le principe de l'*ARF* s'appuie simplement sur le nombre d'acquittements reçus par une station émettrice pour décider du prochain débit à déployer pour la transmission des prochaines trames. Cette technique ne s'appuie pas sur des informations étrangères, comme par exemple une mesure physique de la qualité du canal via des indicateurs (mesure du *SNR*). De ce fait, il est facile à mette en œuvre et complètement compatible avec la norme 802.11. En effet, *ARF* incrémente le débit d'émission physique R_i pour passer à un autre débit supérieur R_{i+1} parmi ceux alloués par la norme après un nombre fixe de transmissions réussies égal à 10, ou après l'expiration d'un temporisateur T chargé au départ. En d'autres termes, le mécanisme *ARF* décide d'augmenter le débit utilisé à un instant donné en un autre plus rapide s'il juge que les conditions du canal sont améliorées. Au contraire des autres algorithmes recensés dans la littérature, *ARF* ne s'appuie pas sur des mesures directes de la couche physique lors de la réception d'une trame. Il juge simplement l'amélioration de l'état du canal en comptant fondamentalement le nombre de succès consécutifs de transmissions réalisées actuellement ou l'expiration du temporisateur T. L'utilisation de ce temporisateur est très utile dans le cas où les conditions sont excellentes et favorables pour adopter un débit plus important, mais que le compteur de transmissions réussies et successives n'atteindra jamais la valeur souhaitée (10). Ce cas est très courant dans ce type de réseau sans fil où l'échec d'une transmission n'indique pas seulement que le débit actuel est inadéquat.

La transmission juste après une remontée de débit doit être effectuée avec succès, sinon le débit décroît immédiatement (retour à un débit plus petit en valeur), et le temporisateur T sera réinitialisé à zéro. Suite à deux échecs d'émissions consécutifs, l'algorithme décroît le débit jusqu'à atteindre de nouveau un nombre d'acquittements (*ACKs*) positifs et successifs égal à 10

ou l'expiration du temporisateur T déjà rechargé. De cette manière, l'algorithme détecte la détérioration de la qualité du canal par deux transmissions consécutives échouées (deux *ACKs* négatifs), et choisit de revenir à un débit antérieurement utilisé. Cette même décision est prise après un seul échec de transmission précédée d'une augmentation de débit. Dans ce cas, le mécanisme estime que le nouveau débit adopté ne sera pas adéquat aux prochaines transmissions. La *Figure* 3.1 présente un diagramme de choix des débits qui résume le fonctionnement de l'*ARF*.

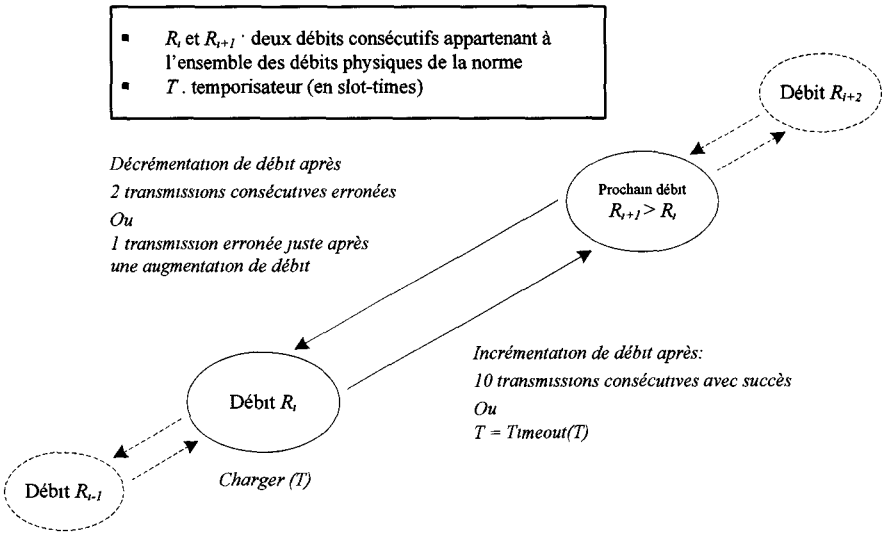

Figure 3.1 : Diagramme de transition de l'algorithme ARF

Bien qu'*ARF* augmente le débit de la transmission à fréquence fixe (après 10 transmissions consécutives) jusqu'à atteindre un meilleur taux (la valeur du débit maximal), ce modèle présente des inconvénients :

- Le processus peut être coûteux, vu que l'échec d'une transmission (produit par une remontée de débit exigée par le mécanisme *ARF*) diminue le débit moyen observé. Plus précisément, si l'état du canal est constant (garde les mêmes caractéristiques), *ARF* essaiera par défaut d'utiliser un débit plus élevé toutes les dix transmissions réussies, ce qui induit des tentatives de changement de débit inutiles et qui engendre des paquets erronés et diminue ainsi considérablement l'efficacité de cet algorithme.
- D'autre part, *ARF* est incapable de stabiliser les variations de débits. Si les conditions du canal changent et se détériorent brusquement, l'*ARF* sera incapable de réagir à la même vitesse pour s'adapter à l'état actuel. Il faudra que plusieurs transmissions erronées soient

effectuées pour qu'il atteigne la valeur du débit souhaité. Par conséquent, cet algorithme ne peut plus s'accommoder rapidement avec les changements instantanés du support de communications.

2. *Adaptive Auto Rate Fallback (AARF)*

Pour surmonter ces défaillances, une nouvelle approche nommée *Adaptive Auto Rate Fallback* (*AARF*), a été proposée dans [35]. Elle repose sur l'historique de la communication et vise à réduire les variations de débit inutiles causées par une fausse interprétation de l'état du canal. Ainsi, cette méthode contrôle le temps de décision en employant le mécanisme ***BEB*** (*Binary Exponentiel Backoff*) précédemment présenté dans le chapitre 1 (le même utilisé par les mécanismes d'accès *CSMA/CD* et *CSMA/CA*).

Lorsque la transmission d'un paquet échoue juste après une montée en débit, un débit plus faible est choisi durant la prochaine tentative d'émission. En plus de cette action, le nombre de transmissions consécutives réussies n requis pour commuter à un débit plus élevé pour la prochaine tentative sera multiplié par deux (avec une valeur limite qui ne dépasse pas $n_{max} = 50$). Similairement à l'ancienne version, lors d'une réduction de débit causée par une transmission erronée de deux trames consécutives, cette valeur est réinitialisée à $n_{min} = 10$. Le schéma de la *Figure* 3.2 suivante explique brièvement le fonctionnement de l'*AARF*.

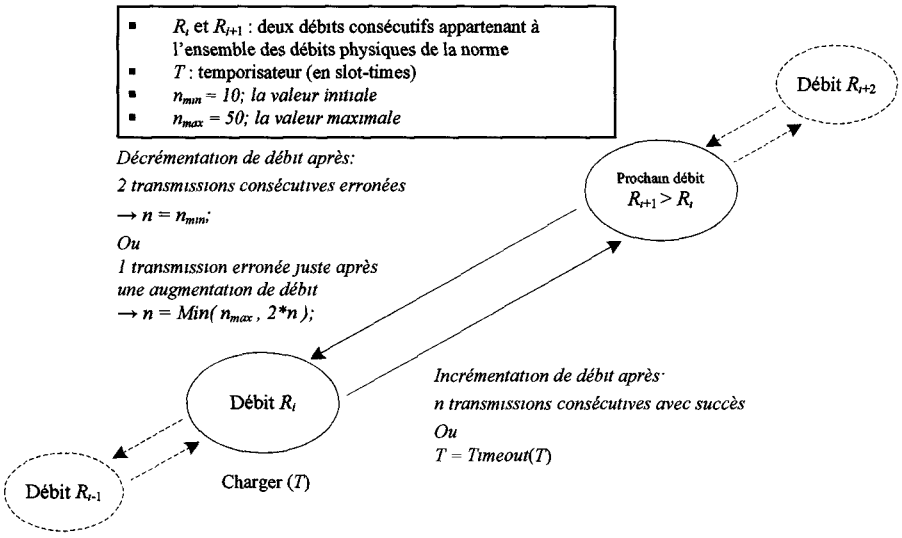

Figure 3.2 : Diagramme de transition de l'algorithme AARF

Par conséquent, cette nouvelle version contrôle dynamiquement le nombre d'*ACKs* positifs nécessaires pour la prochaine remontée en débit. Cependant elle garde les caractéristiques de l'ancienne implémentation en cas de changements brusques produits sur l'état du canal. La grande et unique différence de comportement entre *AARF* et *ARF* est perceptible dans le cas d'un canal stable.

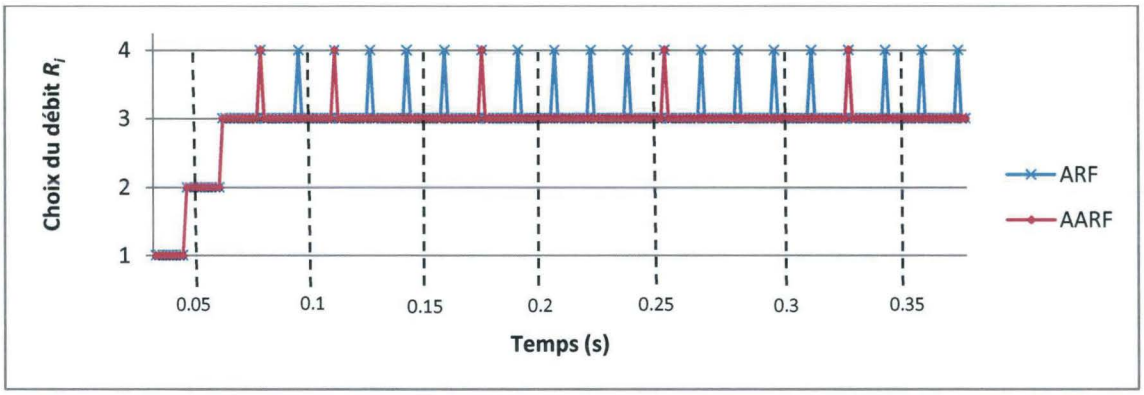

Figure 3.3 : Comparaison entre AARF et ARF

La *Figure* 3.3 illustre le comportement des deux approches *ARF* et *AARF* pendant une période de temps égale à 0.4s permettant la transmission de 230 trames de données. Les débits R_i adoptés dans cette expérience sont 1, 2, 5.5 et 11Mbit/s du 802.11b. Durant cette expérience, nous choisissons des conditions de canal favorisant l'utilisation d'un débit physique R_3 (5.5Mbit/s) pour la transmission des données. Nous remarquons que l'on assiste à une augmentation de la période entre deux tentatives successives erronées pour un état stable du canal de transmission pour le mécanisme d'adaptation *AARF*. En effet juste après une remontée de débit, si la transmission d'une trame est erronée l'incrémentation du débit ne sera établie qu'après un intervalle de temps plus large. Tandis que l'*ARF* essaie d'incrémenter le débit actuel vers un autre plus élevé toutes les dix transmissions successives réussies. Nous pouvons noter dans l'illustration que dans l'intervalle de temps [0.2s, 0.25s], *AARF* n'effectue aucune tentative, alors que *ARF* en réalise trois. De même, l'algorithme *AARF* diminue considérablement les erreurs produites suite à des mauvaises décisions (3/4 des erreurs liées à des mauvaises interprétations ont été éliminées par rapport à celles engendrées par le mécanisme *ARF*).

3. Discussion

Nous avons montré lors de cette étude menée sur les algorithmes *ARF* et *AARF* de sélection de débit actuellement déployés dans les cartes WiFi qu'ils ne permettent pas une prise de décision précise lorsque le canal est relativement bruité. Malgré les améliorations apportées au schéma de transmission, ces deux modèles n'on pas encore abouti à la perfection, puisqu'ils ne peuvent pas réagir instantanément aux changements brusques de l'état du canal. De plus, un intervalle de temps important est nécessaire pour atteindre un débit maximal. Ces mécanismes ne représentent donc pas des solutions optimales pour l'adaptation du lien physique dans un milieu bruité.

De ce fait, nous proposons une nouvelle technique d'adaptation de débit afin d'améliorer la décision en fonction des conditions instantanées du canal en nous basant sur d'autres paramètres existants et non exploités (à savoir le facteur temps d'aller-retour des paquets) tout en respectant les exigences de la norme et en restant conforme au standard.

En effet, lors d'une variation lente de la qualité du support, *AARF* est mieux adapté que *ARF*, puisqu'il procède à l'élimination des accroissements inutiles de débit. Et par la suite, il diminue largement le nombre de paquets perdus tout en s'appuyant sur l'historique des changements de débit déjà réalisés précédemment. Cependant, cette amélioration reste insuffisante puisque le critère de décision dépend seulement de la nature des acquittements (*ACKs*) tandis que ce paramètre ne fournit plus d'information suffisante sur l'état instantané du canal. Il en résulte une forte latence pour atteindre le débit maximum à un moment donné. En d'autres termes, lors d'une décision portant sur le choix du débit, un acquittement négatif (ou la non réussite d'une transmission) est interprété uniquement par une détérioration de la qualité du medium. Toutefois, ce phénomène peut être causé par plusieurs autres anomalies réseaux (destination non trouvée, collision produite avec une autre trame de données, *CRC* erroné, etc.). Analogiquement dans le cas contraire, où tous les *ACKs* reçus seront positifs et successifs, les deux mécanismes doivent attendre au minimum plusieurs multiples de dix transmissions (10**d*) pour atteindre le débit souhaité (nécessitant *d* sauts de débit pour être admis). Nous suggérons alors une nouvelle architecture, compatible avec le standard 802.11, établie à partir de l'algorithme *AARF* et baptisée ***MAARF*** « *Modified Adaptive Auto Rate Fallback* ».

III. Nouvelle technique de contrôle adaptatif

L'idée de base de la nouvelle méthode appelée *Modified Adaptive Auto Rate Fallback* (*MAARF*) est d'introduire un nouveau paramètre qui coopère avec le nombre d'*ACKs* afin d'améliorer la dynamique de réaction du mécanisme, et par la suite de fournir une prédiction suffisamment précise pour mieux s'adapter aux changements instantanés du canal. Dans la présente section, nous définissons tout d'abord les principaux paramètres qui sont à la base de la réalisation de ce mécanisme. Ensuite, nous décrivons le principe de son fonctionnement et donnons les résultats obtenus lors de son application en les comparant avec les techniques usuelles.

1. Round Trip Time (RTT)

Les protocoles du transport fiables tels que ***TCP*** (*Transport Control Protocol*) [36] sont principalement conçus pour fonctionner dans les réseaux traditionnels où les pertes de paquets sont dues principalement à la congestion. Cependant, les réseaux sans fil introduisent des erreurs supplémentaires engendrées par les variations non contrôlées du canal de transmission.

Face aux problèmes de congestions, *TCP* répond à toute perte en invoquant des algorithmes de contrôle de congestion tels que *Slow Start, Congestion Avoidance et Fast Retransmission.* Ces techniques ont été introduites dans différentes versions de la machine protocolaire *TCP* (*TCP Reno, TCP Tahoe,* etc.).

Ces algorithmes réactifs consistent à contrôler la largeur de la fenêtre de congestion en fonction d'erreurs observées. La solution proposée par la version *TCP Vegas* [37, 38] pour les systèmes préventifs, adoptée ensuite par le protocole *TCP*, consiste à modifier la taille de la fenêtre de congestion après estimation de l'état de la connexion par une mesure du délai de transmission et réception d'un segment *TCP* notée ***RTT*** (*Round Trip Time*). Ce dernier paramètre *RTT* est, par définition, le délai moyen qui s'écoule entre l'instant d'émission t_e d'un segment *TCP* jusqu'à l'instant de réception de l'*ACK* t_r correspondant (voir *Figure* 3.4).

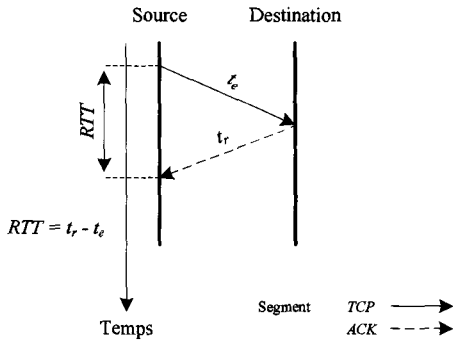

Figure 3.4 : Définition du Round Trip Time RTT

Si les *RTTs* observées sont plus importantes en valeur, le protocole *TCP* en déduit que le réseau commence à être congestionné, et réagit par conséquent par une diminution de la taille de la fenêtre de congestion (le nombre de trames à envoyer ainsi que leurs tailles). Si les valeurs des *RTTs* enregistrées baissent, le protocole détecte une amélioration des conditions du medium utilisé pour les communications, et que le réseau n'est plus surchargé et congestionné. Par conséquent, il procède à l'incrémentation de la taille de la fenêtre, et ainsi à une bonne exploitation des performances allouées par le canal.

2. Intégration du paramètre RTT

Une information intéressante et instantanée sur le canal est déduite après chaque transmission d'une trame de données : c'est l'observation et le calcul du *RTT*. Cette mesure est intégrée dans le nouvel algorithme de contrôle *MAARF* afin de corriger le débit de transmission de données en fonction de la capacité observée du canal. Le nouveau mécanisme *MAARF,* dont le but est de prévoir et de minimiser les pertes inutiles de données, est basé sur le choix d'une valeur de débit convenable de la trame suivante transmise en fonction de la valeur de *RTT* mesurée. L'algorithme effectue alors une correspondance entre la valeur du *RTT* observée et la décision sur le choix du débit.

Par ailleurs nous définissons deux types de *RTT*. La première est la valeur observée directement sur le canal, suite à l'envoi d'une trame, appelé *RTT instantanée* et notée par RTT^*. La deuxième, notée RTT_i, est une valeur théorique déduite en fonction du débit d'envoi R_i utilisé et de la taille d'une trame de données. Dans le cas d'une transmission sans perte d'une $Trame_i$ qui se traduit par la réception de l'acquittement associé ACK_i, une valeur du RTT^* est calculée. Nous introduisons un temporisateur de recouvrement, appelé *« Retransmission Time Out »* et

noté RTO_i, qui détectera la bonne réception ou la perte d'une trame de données. En effet, l'émetteur détecte la perte de $Trame_{i+1}$ en l'absence d'acquittement retourné jusqu'à l'expiration de ce temporisateur RTO_{i+1}. Dans ce cas, une retransmission de la dernière $Trame_{i+1}$ émise s'avère obligatoire (voir *Figure* 3.5).

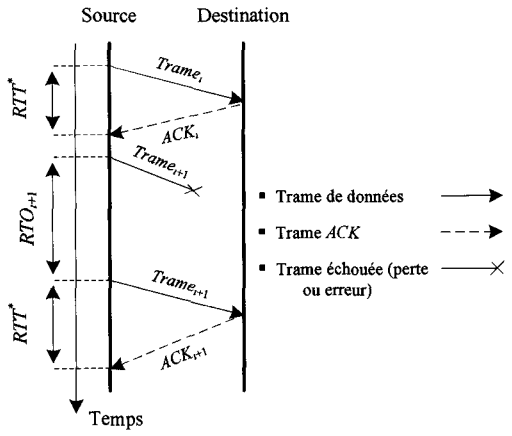

Figure 3.5 : Transmissions des trames de données

Similairement, nous introduisons autour de la valeur de RTT_i théorique correspondant à un débit R_i un intervalle défini par $[RTT_i^+, RTT_i^-]$. Si la valeur de RTT^* observée sur le canal appartient à cet intervalle (c.-à-d. suffisamment proche de la valeur théorique prévue RTT_i), alors les variations de l'état du support seront considérées comme non significatives et ne requièrent aucun changement du débit R_i actuel. En d'autres termes, si la valeur du RTT^* appartient à l'intervalle $[RTT_i^+, RTT_i^-]$ la qualité du lien est jugée stable. Par conséquent, il sera préférable que l'émetteur transmette avec la même valeur du débit R_i actuel puisque la valeur du RTT^* est jugée très proche de RTT_i, celle espérée théoriquement. En dehors de cette fenêtre, les variations du canal correspondent à :

- Une amélioration, si la valeur de RTT^* est inférieure à celle du RTT_i^+ puisque l'*ACK* d'une trame transmise est arrivée plus tôt que prévue, donc les conditions sont améliorées et le risque de perte est minimal.
- Une dégradation, si la valeur de RTT^* est supérieure à celle du RTT_i^- puisque l'*ACK* d'une trame transmise, a mis plus de temps pour arriver à la source, et par la suite nous pouvons juger que le risque de perte de données augmente.

La station doit alors changer de débit d'émission et l'adapte selon ces deux interprétations de l'état instantané du canal. Rappelons que $RTT_{i+1} < RTT_i < RTT_{i-1}$ puisque $R_{i+1} > R_i > R_{i-1}$.

Donc, nous aurons $RTT_i^+ < RTT_i^-$. Le choix des paramètres RTT_i^+ *et* RTT_i^- pour chaque RTT_i théorique n'est pas arbitraire (voir les *Equations* 1 et 2 ci-dessous).

$$eq.1 \qquad RTT_i^- = (RTT_{i-1} + RTT_i)/2$$

$$eq.2 \qquad RTT_i^+ = (RTT_{i+1} + RTT_i)/2$$

Comme illustré également dans la *Figure* 3.6, RTT_i^- sera le milieu de l'intervalle $[RTT_{i-1}, RTT_i]$ et la valeur RTT_i^+ celui de l'intervalle $[RTT_i, RTT_{i+1}]$.

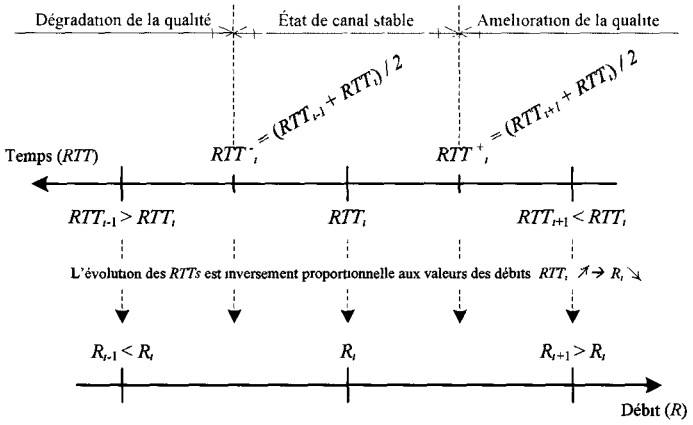

Figure 3.6 : Choix des paramètres de l'algorithme : RTT_i^+ *et* RTT_i^-

3. Principe de l'algorithme MAARF

Pour chaque trame envoyée avec succès, nous comparons l'écart entre les deux valeurs RTT^* instantané et RTT_i théorique. Plus spécifiquement, nous testons si la valeur du RTT^* a excédé ou non les bornes RTT_i^+ et RTT_i^-. Toutefois, la décision est prise après observation de plusieurs échantillons de RTT^* :

- Tant que la valeur du nombre de transmissions avec succès n (initialisé à n_{min}) n'est pas atteinte, nous effectuons les tests suivants :
  - Si la valeur observée du RTT^* est inférieure à RTT_i^+ ($RTT^* < RTT_i^+$) durant h transmissions successives et si le débit maximal (54 Mbit/s) n'est pas encore atteint, alors RTT^* instantané est jugé largement inférieur à RTT_i et plutôt proche de RTT_{i+1}. Subséquemment, nous commutons à un débit plus élevé R_{i+1} pour la prochaine tentative (nous interprétons que les caractéristiques du canal s'améliorent).
  - Si la valeur du RTT^* est supérieure à RTT_i^- ($RTT^* > RTT_i^-$) pendant g fois de suite et si le débit minimal (6 Mbit/s) n'est pas encore atteint, ceci implique que la valeur actuelle

du RTT^* instantané est beaucoup plus grande que celle du RTT_i attendue et plutôt proche à la valeur de RTT_{i-1}. Ainsi, nous détectons un début de détérioration de la qualité du lien et nous réduisons par conséquent le débit (vers R_{i-1}) à la prochaine tentative d'envoi.

- Si la valeur du RTT^* reste comprise entre les deux bornes théoriques RTT_i^+et RTT_i^- ($RTT_i^+ < RTT^* < RTT_i^-$), alors le débit reste invariant R_i (nous jugeons que les caractéristiques du support de transmission sont constantes).

➢ Similairement à l'algorithme *AARF*, lorsque le nombre de transmissions consécutives avec succès atteint la valeur souhaitée (qui peut être à un instant donné 10, 20, 40 ou 50), nous commutons à un débit supérieur sans prendre en considération les valeurs observées des RTT^*.

Aussi, de même que *AARF*, lorsque la transmission échoue (un paquet perdu, pas d'acquittement reçu pendant une valeur supérieure à RTO_i), nous modifions les valeurs des paramètres décisionnels (n, g et h) comme suit :

➢ Si l'erreur de la transmission est survenue juste après une incrémentation de débit, ce dernier sera décrémenté. De plus, comme illustré dans l'*Equation* 3, la valeur du nombre de transmissions réussies à respecter pour la prochaine remontée sera multipliée par deux avec une valeur limite égale à n_{max}.

$$eq.3 \qquad n = Min(2 * n;\ n_{max}) \tag{3}$$

➢ Si nous détectons deux erreurs consécutives, nous réduisons le débit actuel, tout en réinitialisant la valeur du nombre des transmissions réussies à la valeur minimale ($n = n_{min}$) pour la prochaine remontée.

Nous avons introduit, similairement au paramètre n désignant le nombre de transmissions successives réussies à respecter pour chaque remontée en débit, la loi de *backoff* pour le contrôle des paramètres h et g. Rappelons que ce sont deux nouveaux paramètres calculent respectivement le nombre de fois où ($RTT^* < RTT_i^+$) et ($RTT^* > RTT_i^-$). Ces deux variables seront dynamiques comme le paramètre n déjà utilisé et subiront des règles de variations et des limites supérieures et inférieures pour maintenir une décision rigoureuse d'incrémentation ou de décrémentation du débit.

- Dés qu'une erreur de transmission survient juste après une remontée suite à une décision causée par une interprétation de la valeur du RTT^*, le débit sera diminué et la valeur de h (comme illustré dans l'*Equation* 4) sera multipliée par deux, tant que la limite supérieure h_{max} n'est pas atteinte.

$$eq.4 \qquad h = Min(2 * h; h_{max})$$

Autrement dit, pour les prochaines transmissions avec succès et la condition ($RTT^* < RTT_i^+$) vérifiée, nous utiliserons la nouvelle valeur de h pour la décision qui portera sur l'incrémentation du débit.

- De même, si une erreur de transmission est perçue juste après une décrémentation de débit causée par une décision basée sur des comparaisons des valeurs enregistrées du RTT^* avec celle du RTT_i, alors le débit sera élevé vers son ancienne valeur. De même, la valeur de g (voir *Equation* 5) sera multipliée par deux, tant que la limite g_{max} n'est pas atteinte.

$$eq.5 \qquad g = Min(2 * g; g_{max})$$

Ceci veut dire que pour les prochaines transmissions nous ne prendrons la décision d'une décrémentation de débit liée aux interprétations du RTT^* qu'après atteinte de la nouvelle valeur doublée de g.

- Ces deux nouveaux paramètres h et g seront aussi réinitialisés, identiquement au paramètre n, après deux transmissions erronées consécutives comme suit :

$$h = h_{min} \ , \ g = g_{min}$$

4. Paramétrage de MAARF

Dans cette section nous détaillons les différents paramètres que nous avons déployés pour concevoir l'algorithme *MAARF*. Tout d'abord la norme 802.11 définit dans ses versions 802.11a et 802.11g, différents sauts de débit physique dont les valeurs peuvent atteindre 54 Mbit/s. Ainsi, nous définissons :

- R_i : le débit actuel qui varie parmi les valeurs suivantes {6, 9, 18, 12, 24, 36, 48, 54} représentées en Mb/s.

- RTT^* : c'est la valeur observée lors de l'envoi d'une trame (retournée à partir du canal de transmission).
- RTT_i : c'est le temps qui sépare l'envoi d'une trame jusqu'à la réception de l'acquittement associé. Ce temps est calculé à partir de l'occupation du canal et n'englobe pas les temps d'attente de l'émetteur avant d'accéder au medium. Il est donné par l'*Equation* 6 suivante :

$$eq.6 \qquad RTT_i = t_{em.Trame} + t_{propa\,g} + t_{trait.Récepteur} + SIFS + t_{em.ACK} + t_{propag} + t_{trait.Emetteur}$$

avec : t_{em} *le temps nécessaire pour l'émission des données (Trame ou ACK)*

t_{propag} *le temps de propagation sur le canal de transmission*

t_{trait} *le temps de traitement de chaque trame reçue*

En pratique ce temps est seulement représenté par le temps d'émission de la trame de données comme illustré dans l'*Equation* 7. Cette approximation est réalisée parce que les autres temps mentionnés sont très négligeables en valeurs par rapport à celui choisi.

$$eq.7 \qquad RTT_i \approx t_{em.Trame} = \frac{Taille_{trame}}{R_i}$$

- RTO_i (*Retransmission Time Out*) : c'est un temporisateur contrôlant la reprise après une perte de trame et dont la valeur est affectée dynamiquement en fonction du RTT_i (voir *Equation* 8).

$$eq.8 \qquad RTO_i = 2 * RTT_i$$

- RTT_i^- *et* RTT_i^+ : ce sont deux valeurs choisies pour chaque débit R_i utilisé. Ces deux paramètres sont relatifs à la marge de RTT_i comme il a été défini dans les *Equations* 1 et 2 de ce chapitre.
- h : variable responsable de l'incrémentation du débit, elle appartient à l'intervalle $[h_{min}, h_{max}] = [4,16]$.
- g : variable responsable de la décrémentation du débit, elle appartient à l'intervalle $[g_{min}, g_{max}] = [2,8]$.

- n : c'est un ancien paramètre qui a été déjà utilisé et initialisé par *AARF*. Il représente le nombre d'acquittements successifs transmis avec succès, il appartient à l'intervalle [n_{min} , n_{max}] = [10, 50].

Enfin, nous présentons une illustration détaillée du fonctionnent de l'algorithme *MAARF* dans le diagramme de la *Figure* 3.7.

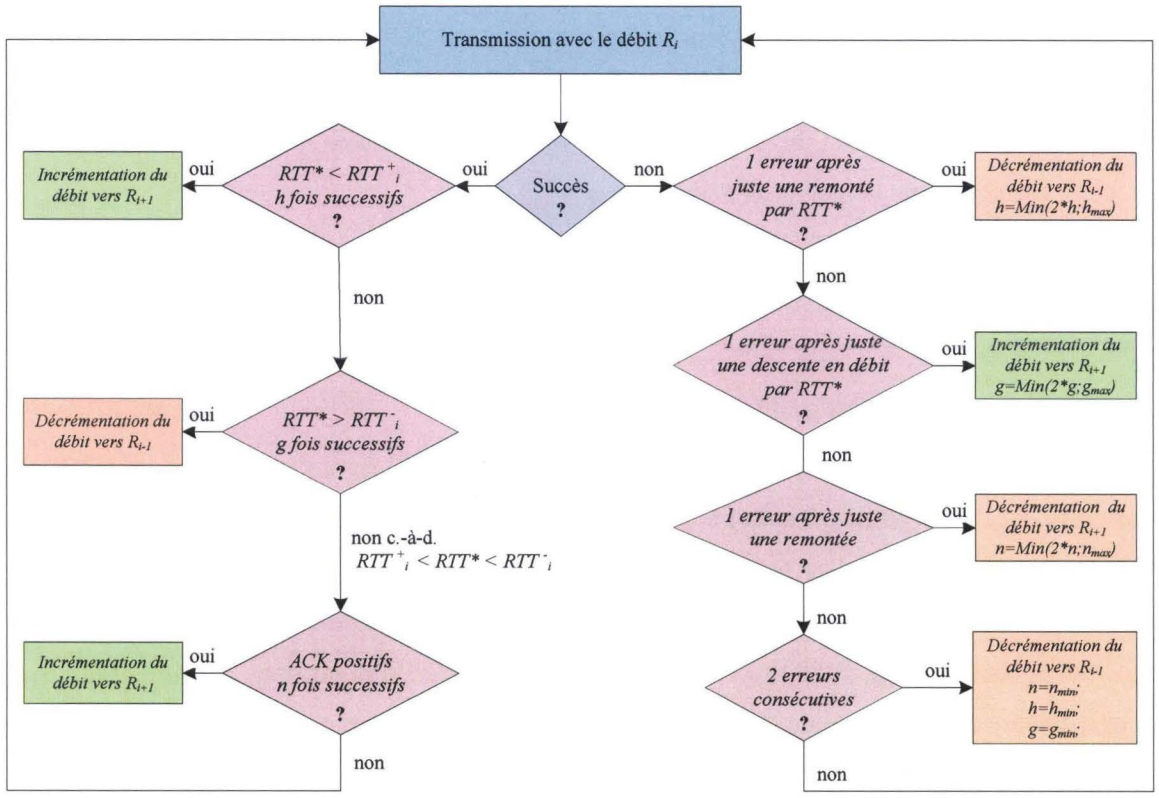

Figure 3.7 : Diagramme de transition du nouvel algorithme MAARF

IV. Résultats et Interprétations

Les algorithmes ont été implémentés en langage C sous un environnement de travail basé sur le système d'exploitation Unix (compilateur gcc/ terminal MAC) afin d'être facilement intégrables dans le simulateur réseau.

Nous avons effectué différents tests avec la configuration suivante :

- Le nombre de trames envoyées est de l'ordre de 100 trames (représentant environ 0.5 seconde).

- La taille de chaque trame de données est égale à la taille minimale d'une trame 802.11g (= 1200 octets).
- Un débit initial R_i de 6Mbit/s (pouvant atteindre 54Mbit/s).
- Le non retour d'un *ACK* traduit un échec de transmission : paquet perdu, *RTO* expiré ou erreur détectée par le CRC.
- Les *ACKs* retournés par le récepteur indiquent un succès de transmission : retour d'un *ACK* positif dans un temps inférieur à *RTO*.
- La valeur du *RTT* actuel (noté RTT^*) est lue après chaque réception d'un *ACK* et retournée aléatoirement dans la plage $[RTT_{i-1}, RTT_{i+1}]$.

Plusieurs cas de figures ont été envisagés afin d'évaluer les performances de notre algorithme par rapport à l'ancienne version *AARF*.

1. Optimisation des paramètres

Cette première expérience est destinée à l'étude et à l'optimisation des paramètres décisionnels de l'algorithme : *h* (comptant le nombre de fois pendant lesquelles nous avons $RTT^* < RTT_i^+$) et *g* (comptant celles pour lesquelles nous avons $RTT^* > RTT_i^-$). Nous discutons les valeurs de h_{min} *et* g_{min}. Dans la *Figure* 3.8, nous montrons les résultats d'implémentation de l'algorithme *MAARF* pour différentes valeurs de ces paramètres. Ces résultats expriment le débit physique adopté dans le canal de transmission en fonction du numéro du paquet pour différentes configurations du nouveau mécanisme.

En choisissant de faibles valeurs de *g* et de *h* ($h_{min} = g_{min} = 1$), l'algorithme procédera à des décisions trop rapides d'incrémentation ainsi que de décrémentation du débit de transmission. Il devient ainsi trop sensible pour déterminer les variations exactes du canal. Néanmoins, en choisissant des valeurs initiales importantes de nos paramètres *g* et *h* ($h_{min} = 10$ *et* $g_{min} = 4$) très proches de h_{max} *et* g_{max}, l'algorithme ne réagit plus efficacement pour des déviations de qualité significatives. Par conséquent, nous notons que les meilleurs paramètres *h et g* initiaux à choisir, de sorte que notre algorithme donne les meilleurs résultats, sont respectivement 4 et 2.

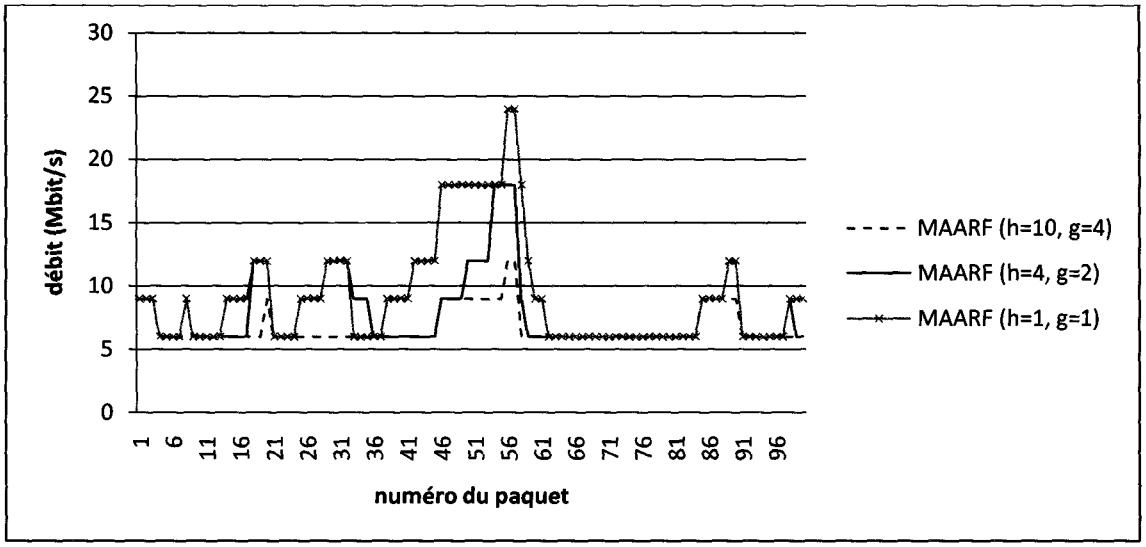

Figure 3.8 : Adaptations de débit avec différents paramètres du MAARF

2. Régimes des Tests

A. État de canal non stable

Nous comparons ici notre nouvelle technique à celle de l'*AARF* (utilisée dans 802.11) en présence d'un régime transitoire de l'état du canal (les conditions s'améliorent et se dégradent aléatoirement pendant plusieurs intervalles de temps différents). Dans la *Figure* 3.9, nous présentons le graphique correspondant.

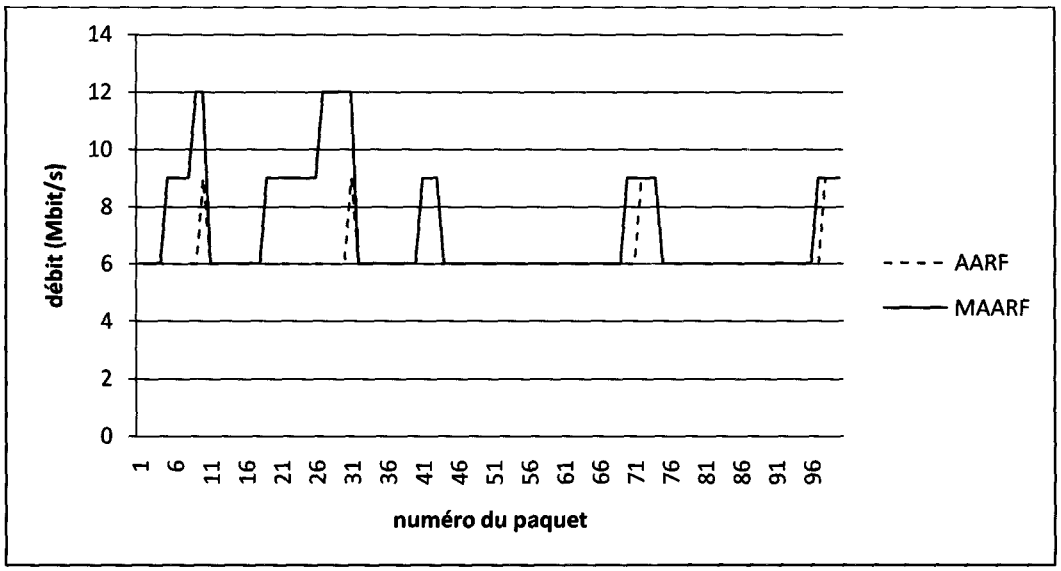

Figure 3.9 : Adaptations de débit en régime transitoire

Le graphique de la *Figure* 3.9 montre que la réaction contre les variations du canal est beaucoup plus efficace dans *MAARF* que dans l'*AARF*. En effet le nouvel algorithme détecte la disponibilité du canal, il choisit ainsi le bon débit à partir de la $4^{éme}$ trame, alors que *AARF* n'atteint ce débit qu'à partir de la $10^{éme}$ trame. Nous constatons aussi que la dynamique de notre algorithme *MAARF* contre les perturbations est similaire à celle d'*AARF* lorsque la qualité de canal se dégrade. Cette propriété a été obtenue par l'optimisation des paramètres de l'algorithme (voir *Figure* 3.8).

Enfin, le débit moyen enregistré des deux algorithmes, en régime transitoire, est de l'ordre de 6.01 Mbit/s pour *AARF* et de 7.60 Mbit/s pour celui du nouvel algorithme soit une amélioration significative en débit moyen de l'ordre de 26%.

B. État de canal stable

Nous supposons, dans ce cas, que seuls des acquittements positifs seront retournés vers l'émetteur des trames (des transmissions de paquets sans pertes : cas d'un canal parfait). Les valeurs du RTT^* enregistrées sur le medium seront ainsi proches de celles du RTT_i théorique correspondantes et qui s'améliorent au cours du temps.

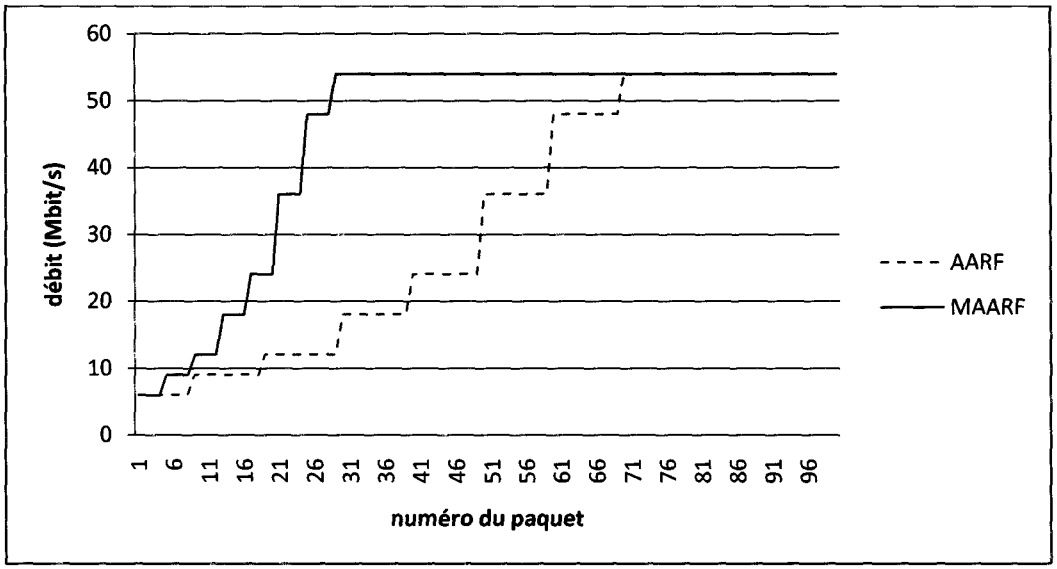

Figure 3.10 : Adaptations de débit en absence d'erreurs

Nous remarquons que la différence entre les débits moyens des deux techniques est beaucoup plus importante (32.04 Mbit/s dans le cas de *AARF* et de 45.48 Mbit/s pour celui du nouvel algorithme). L'amélioration du débit moyen ainsi obtenu est de l'ordre de 41%.

Selon les résultats obtenus dans la *Figure* 3.10, le débit maximal (54Mbit/s) est atteint plus rapidement par le nouvel algorithme que dans l'ancienne version *AARF* puisqu'il détecte rapidement l'amélioration du canal (dès la trame N° 28) et profite ainsi des grandes valeurs de débit possible. Alors que la technique *AARF* avec les mêmes conditions du support de communication admet une dynamique plus plate et atteint la valeur maximale du débit beaucoup plus tard (seulement à partir de la trame N° 70). Ceci est du au fait que cette dernière doit attendre, au minimum et dans les meilleures conditions, 10 *ACK* positifs à chaque saut de débit.

C. Régime d'une station mobile

Une quatrième simulation sur l'adaptation de débit dans un canal en régime sinusoïdal est menée. Dans le cas d'une station mobile dans 802.11, les variations de la qualité du support sont très rapides et totalement différentes. Ceci est traduit par des intervalles où les conditions du canal s'améliorent rapidement, séparés par ceux où l'état se dégrade brusquement. Nous constatons dans la *Figure* 3.11 que la technique *MAARF* adapte le même débit que celle d'*AARF*, mais sa dynamique de montée est plus agile et prédictive des conditions du medium de communication. Toutefois, les deux méthodes passent à un débit plus faible, presque au même instant, lorsque des erreurs de transmission surviennent.

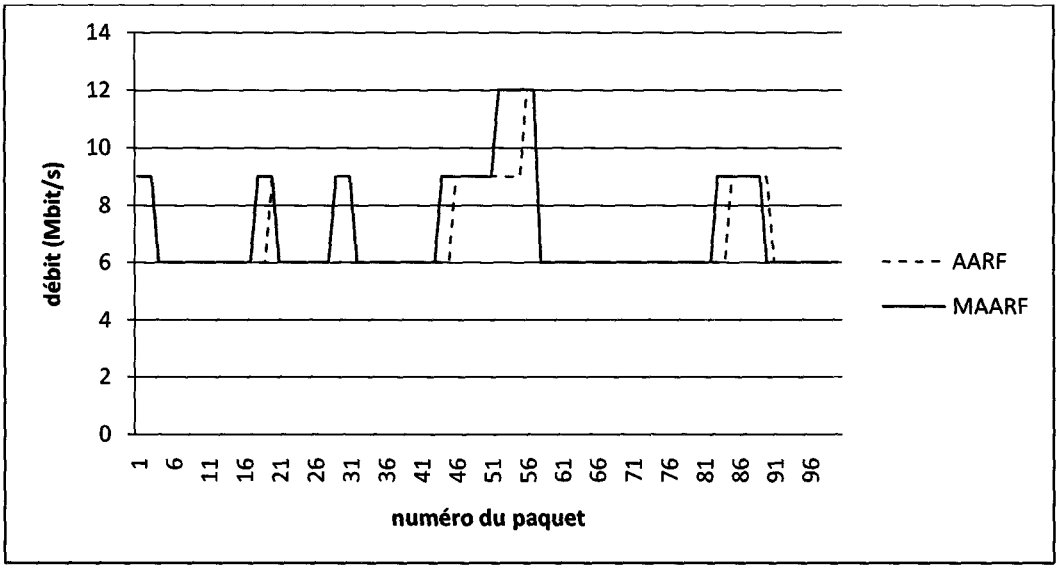

Figure 3.11: Adaptations de débit aux conditions instantanées du canal

Le débit moyen obtenu est égal à 6.72 Mbit/s pour le mécanisme *AARF* et 7.89 Mbit/s pour le nouvel algorithme. En conséquence, une amélioration de 17% est retenue entre ces deux mécanismes de contrôle du lien physique.

V. Synthèse des résultats

Les résultats obtenus lors de l'implémentation du nouvel algorithme *MAARF* ont montré qu'il est possible :

- D'estimer le canal grâce aux valeurs observées du RTT^*.
- De détecter la perte de paquet avant qu'elle ne se produise.
- De prendre les décisions nécessaires plus rapidement que les mécanismes actuels.

Nous avons montré au cours de cette étude qu'il n'est plus nécessaire d'attendre 10 acquittements consécutifs ou plus pour décider une modification de débit théorique comme le font les algorithmes classiques. Dans les *Figures* 3.12 et 3.13 nous comparons les résultats obtenus par l'application du nouveau mécanisme *MAARF* et de l'actuel *AARF*, respectivement sur le débit moyen calculé et le nombre de trames erronées, en fonction du nombre de paquets transmis pour les mêmes conditions du canal.

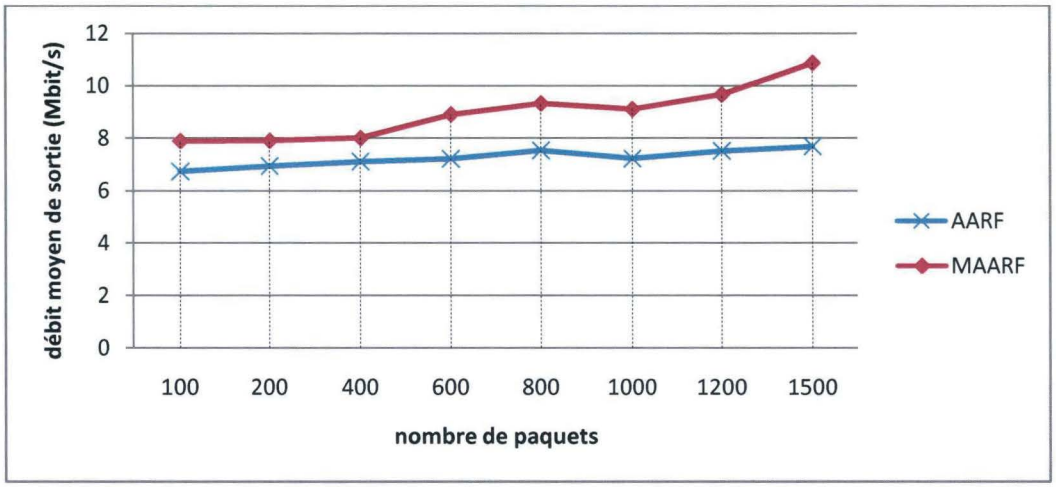

Figure 3.12 : Débit moyen en fonction du nombre de paquets transmis

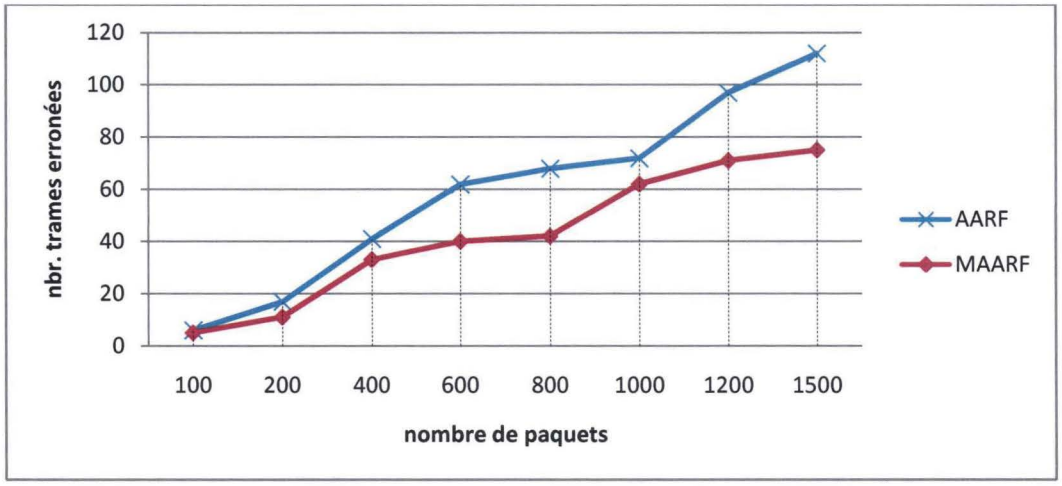

Figure 3.13 : Nombre de trames erronées par rapport au nombre de paquets transmis

Nous montrons par les expériences théoriques menées, que les améliorations sont très nettes et distinguées en débit général de sortie et en nombre de paquets erronés. Cependant les résultats obtenus ne permettent pas de tirer des conclusions définitives concernant l'algorithme proposé. Cette insuffisance nous amène à confirmer les performances de l'algorithme *MAARF* par la simulation de ce dernier sur la plateforme du standard ***NS-2*** [39] (*Network Simulator 2*) utilisée largement par la communauté scientifique des réseaux. Dans le chapitre 4 nous allons effectuer différentes simulations sous cette plateforme afin de comparer les résultats avec ceux déjà obtenus par les autres techniques actuelles, et dégager les performances ainsi que les apports de cette nouvelle technique.

CHAPITRE

4 Evaluation du *MAARF*

I. Objectifs

Jusqu'à maintenant, nous avons montré que le *MAARF* obtient de bonnes performances. Ce présent chapitre est consacré à valider ces résultats par des simulations de notre algorithme sous *NS-2* (*Network Simulator 2*) [39]. Le choix du simulateur n'a pas été aléatoire et nous a conduit à ce dernier, puisqu'il présente une plateforme qui permet de tester à moindre coût les nouveaux protocoles sur des architectures de test standard. Nous allons l'exploiter pour valider l'efficacité des mécanismes de contrôle de débit. En conséquence, nous implémenterons le nouveau mécanisme élaboré puis réaliserons des simulations pour les trois algorithmes d'adaptation de lien (*ARF, AARF et MAARF*) sous différents scénarii.

II. Network Simulator - 2

Dans cette section, nous décrivons brièvement le simulateur *NS-2* utilisé, le logiciel de visualisation *NAM* (*Network AniMator*) fourni avec ce dernier, ainsi que l'application ***Xgraph*** dédiée à la création des graphiques à partir des fichiers « *trace* » contenant les résultats des scénarios de tests. Etant donné que cet outil appartient à la famille des logiciels libres « *open source* », il s'avère pénible de trouver un document standard guidant son utilisation dans la littérature. De ce fait, nous jugeons essentiel d'introduire sommairement ce produit ainsi que ses extra-outils déployés.

1. Présentation du NS-2

Le *network simulator 2* [39] a été originellement développé en tant que variante du « *REAL network simulator* » en 1989, et a considérablement évolué au cours des années. Actuellement, son implémentation est supportée par le ***DARPA*** (*Defense Advanced Research Projects Agency*) à travers le projet ***SAMAN*** (*Simulation Augmented by Measurement and Analysis for Networks*) et par la ***NSF*** (*National Science Foundation*) à travers ***CONSER*** (*Collaborative Simulation for Education and Research*). *NS-2* est un simulateur de réseaux informatiques à événements discrets développé dans un but de recherche. Il est bâti autour d'un langage de Programmation appelé ***TCL*** (*Tool command Language*) et dont il est une extension. Il est fourni avec divers outils d'analyse complémentaires, eux-mêmes écrits en *C/C++* ou *TCL/Tk*. Son utilisation est particulièrement adaptée aux réseaux de commutation de paquets et à la réalisation de simulations de petites tailles. Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage point à point ou point à multipoint, des protocoles de transport, de session, de réservation, des services intégrés, des protocoles d'application comme *HTTP*, etc. De plus, le simulateur possède une palette de systèmes de transmission (couche 1 de l'architecture *TCP/IP*), d'ordonnanceurs et de politiques de gestion de files d'attente pour effectuer des études de contrôle de congestion. La liste des principaux composants actuellement disponibles dans *NS-2* par catégorie est :

- **Application :** Web, FTP, Telnet, générateur de trafic (CBR[1], VBR[2], etc.)
- **Transport :** TCP, UDP, RTP[3], SRM[4], TFRC[5]
- **Routage :** statique, dynamique (à base de vecteur distance) et routage multipoint (DVMRP[6], PIM[7])
- **Gestion de file d'attente :** RED, DropTail, Token bucket
- **Discipline de service :** CBQ[8], SFQ[9], DRR[10], Fair queueing
- **Système de transmission :** CSMA/CD, CSMA/CA, lien point à point

---

[1] CBR: *Constant Bit Rate*
[2] VBR: *Variable Bit Rate*
[3] RTP: *Real Time Transport Protocol*
[4] SRM: *Scalable Reliable Multicast*
[5] TFRC: *TCP-Friendly Rate Control Protocol*
[6] DVMRP: *Distance Vector Multicast Routing Protocol*
[7] PIM: *Protocol Independent Multicast*
[8] CBQ: *Class-Based Queuing*
[9] *SFQ: Start-time Fair Queuing*
[10] *DRR: Deficit Round Robin*

Du point de vue utilisateur, la mise en œuvre de ce simulateur se fait via une étape de programmation qui décrit la topologie du réseau et le comportement de ses composants. L'utilisateur entame ensuite l'étape de simulation et termine par l'interprétation des résultats. Cette dernière étape peut être prise en charge par un outil annexe, appelé *NAM* (*Network AniMator*), qui permet une visualisation et une analyse des éléments simulés. Tout le package est sous une Licence GNU libre : il est donc développé, corrigé et étendu au fur et à mesure de l'apparition de nouveaux besoins (ou de la découverte d'erreurs). Chaque extension sera accessible via une nouvelle version du simulateur.

2. Network AniMator (NAM)

Le *Network Animator* ([40], [41]) est un outil d'animation basé sur *Tcl/TK* et utilisé dans *NS* afin de visualiser le déroulement de la simulation des réseaux en fonction du temps, ainsi que les tracés de données. Le modèle théorique du *Nam* a été non seulement créé pour lire un large ensemble de données d'animation, mais aussi suffisamment extensible pour être utilisé quelque soit le type de réseau simulé (fixe, mobile ou mixte), ce qui permet de visualiser tout type de situation possible.

3. Fichiers "Trace" et le logiciel XGraph

Les résultats obtenus avec *NS-2* se présentent sous la forme d'un fichier texte décrivant précisément l'avancement des paquets dans le réseau en simulation. En effet, chaque ligne représente un paquet envoyé ou reçu par un nœud du réseau. Pour être exploitables, ces informations doivent être traitées par un filtre qui analyse les lignes d'entrées afin de recueillir seulement les informations utiles et spécifiques au graphe désiré.

XGraph est un logiciel annexe libre. Il est capable de créer des graphiques à partir des résultats d'une simulation. En effet ce logiciel est capable d'analyser des fichiers de trace filtrés provenant de *Network Simulator 2*, notamment les informations liées aux réseaux sans fil, et peut être utilisé sous Windows, Linux ou UNIX. A partir des données traitées *XGraph* peut dessiner des graphiques colorés en deux dimensions ou même en trois dimensions et supporte des scripts pour les prétraitements. Par exemple, *XGraph* peut afficher les délais, le temps de traitement, le *« Round Trip Time »*, le nombre de nœuds intermédiaires, les débits et des informations statistiques, et ce tant pour chacun des nœuds que pour le réseau analysé tout entier.

III. Simulations et Analyses

1. Environnement de simulation

Les simulations sont réalisées en utilisant la version 2.27 du *Network Simulator 2* « *NSallinone-2.27* » sous Linux [39]. Cette variante était la version stable et courante du moment (sachant que nous avons commencé à travailler avec le simulateur en 2006). D'autres variantes ont été élaborées depuis, et actuellement nous parlons de la nouvelle version *NS-2.34*.

Le modèle de simulation est un réseau ad hoc, composé de 2 stations mobiles. Les nœuds utilisent les débits définis dans le standard IEEE 802.11b suivant la qualité du lien. C'est à dire que lorsqu'il y a dégradation de la qualité du signal, le débit diminuera de 11 Mb/s à 9, 5.5, 2, ou même 1Mbps. Les débits physiques ne dépassent pas 11Mbps, et nous avons utilisé cinq valeurs différentes agréées par la norme 802.11b en mode ad hoc {1, 2, 6, 9, 11} Mbit/s. Un trafic ***UDP*** (*User Datagram Protocol*) est considéré entre ces nœuds de type ***CBR*** (*Constant Bit Rate*). Les paquets transmis dans le réseau admettent une taille fixe égale à 1000 octets.

La couche *MAC* mise en œuvre est basée sur un modèle simulant le standard 802.11. Une couche de liaison est utilisée pour simuler la latence d'accès au support (28µs). Finalement, le mécanisme de protection *RTS/CTS* à été désactivé vu que tous les paquets échangés ont une taille inférieure au seuil au delà duquel le mécanisme est déclenché. De plus, le mode *ad-hoc* est adopté dans le scénario simulé (seulement deux stations mobiles).

Afin d'implémenter ces techniques de contrôle de débit nous devons modifier et ensuite recompiler le noyau de *NS* pour chaque algorithme. Les modifications requises sont essentiellement apportées au fichier « *Mac_802.11.cc* » par l'insertion des codes appropriés puisqu'il définit et comprend la spécification du mécanisme du choix du débit physique à utiliser.

Le script de simulation utilisé génère un trafic *CBR* de type *UDP* à $t = 10s$ de durée $80s$, et donc, se termine à $t = 90s$ en utilisant un modèle d'erreur uniforme. Le modèle de routage adopté est de type ***AODV*** (*Ad-hoc On demand Distance Vector*). Le *Tableau* 4.1 résume les caractéristiques spécifiées et définies dans le script de simulation.

Attribut	Valeur
Modèle de propagation radio	Two Ray Ground
Portée de transmission	250 mètres
Nombre de nœuds mobiles	2
Débit initial des nœuds	1Mb/s
Protocole de routage	AODV
Slot Time	16 µs
SIFS Time	8 µs
DIFS Time	40 µs
Taille paquet	1000 Bytes
Trafic	CBR / UDP
Temps de simulation	80s (de t=10s)
Surface de simulation	745x745mètres

Tableau 4.1 : Paramètres du modèle de simulation

2. Résultats et Interprétations

Dans cette section nous présentons une série de simulations dans le but d'évaluer les performances de l'algorithme proposé (*MAARF*) face aux solutions existantes.

A. Discussion des paramètres de l'algorithme MAARF

Avant de débuter les comparaisons par rapport aux autres techniques d'adaptation du lien dans les réseaux sans fil 802.11, nous effectuons une simulation qui porte seulement sur l'algorithme *MAARF* et utilisant une taille fixe de paquets. Dans un premier scénario (illustré dans la *Figure* 4.1 ci-dessous) nous appliquons plusieurs valeurs pour ses différentes variables. Le but de cette première simulation est d'étudier les paramètres décisionnels h et g du nouvel algorithme. Les courbes présentées dans la *Figure* 4.1 indiquent le débit réel mesuré en fonction du temps. Nous rappelons que h est le nombre d'occurrences consécutives où la valeur du RTT^* observée après chaque transmission s'améliore, alors que g est le nombre de fois consécutives où la valeur du RTT^* se dégrade continuellement. Nous découvrons, conformément aux valeurs théoriques déjà trouvées, qu'une meilleure adaptation de débit est atteinte lorsque :

$$h_{min} = 4 \; et \; g_{min} = 2.$$

Ce choix est justifié puisque ces paramètres (h et g) ne doivent pas avoir des valeurs très importantes pour qu'ils puissent jouer un rôle et apporter des enrichissements du choix lors de la sélection du débit physique instantané. Similairement, ces paramètres ne doivent pas être trop bas pour garder un équilibre de décision avec la première variable déjà déployée (nombre des

acquittements positifs). Nous notons aussi qu'en choisissant des valeurs trop élevées pour les paramètres *h* et *g* (respectivement 16 et 8), le débit réel enregistré devient trop bas et rejoint en valeur les débits déjà obtenus avec les autres algorithmes étudiés (*ARF* et *AARF*). Ceci s'explique par le fait que l'ancien paramètre *n* devient alors dominant pour le choix du débit physique. Dans la suite des simulations de l'algorithme *MAARF* nous choisissons 4 et 2 comme valeurs des paramètres décisionnelles *h* et *g*.

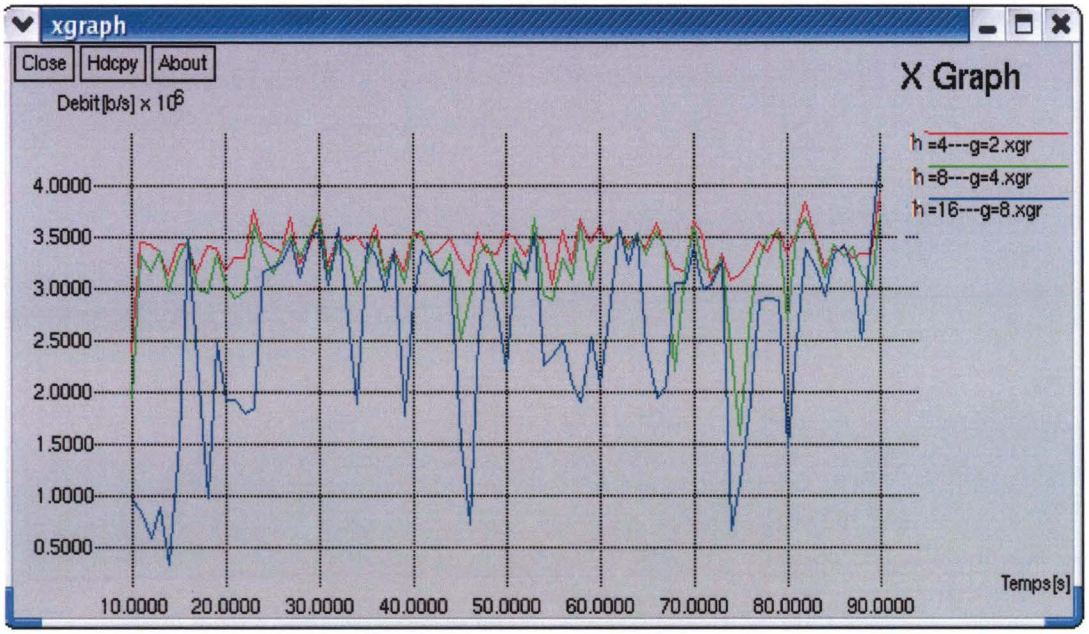

Figure 4.1 : Optimisation des paramètres g et h du MAARF

B. Débit théorique du lien physique

Les tests qui suivent décrivent l'évolution du débit physique en fonction du nombre de paquets transmis dans le réseau. Nous rappelons que l'ensemble des débits physiques alloué par la norme 802.11b est {1, 2, 6, 9, 11}. Les courbes ci-dessous, reflètent le fonctionnement exact des trois algorithmes implémentés dans *NS-2* avec différents taux d'erreurs. Elles présentent le choix instantané des débits physiques en fonction des acquittements retournés. La *Figure* 4.2 illustre une simulation d'un canal de transmission idéal (ou parfait) dont le taux d'erreur est de 0%. Nous avons tracé le comportement ainsi que le choix du débit physique pendant l'envoi des 100 premiers paquets. Nous remarquons distinctement que la rapidité de détection d'une amélioration de l'état du canal par *MAARF* est bien plus rapide que les deux autres algorithmes : dés le paquet N° 18 *MAARF* transmet avec un débit maximal (11Mbps), tandis que *ARF* et *AARF*

atteignent cette valeur seulement à partir du paquet N° 48 ce qui engendre un gain en temps considérable pour atteindre le débit adéquat (débit maximal dans ce premier cas). Ce gain sera traduit ensuite par une amélioration générale du débit réel observé. Nous notons que les courbes d'*ARF* et d'*AARF* sont confondues puisque les deux algorithmes adoptent le même débit physique pour chaque paquet envoyé. En effet, en absence d'erreurs les deux techniques emploient des décisions similaires pour les remontées des débits (chaque dix émissions réussies).

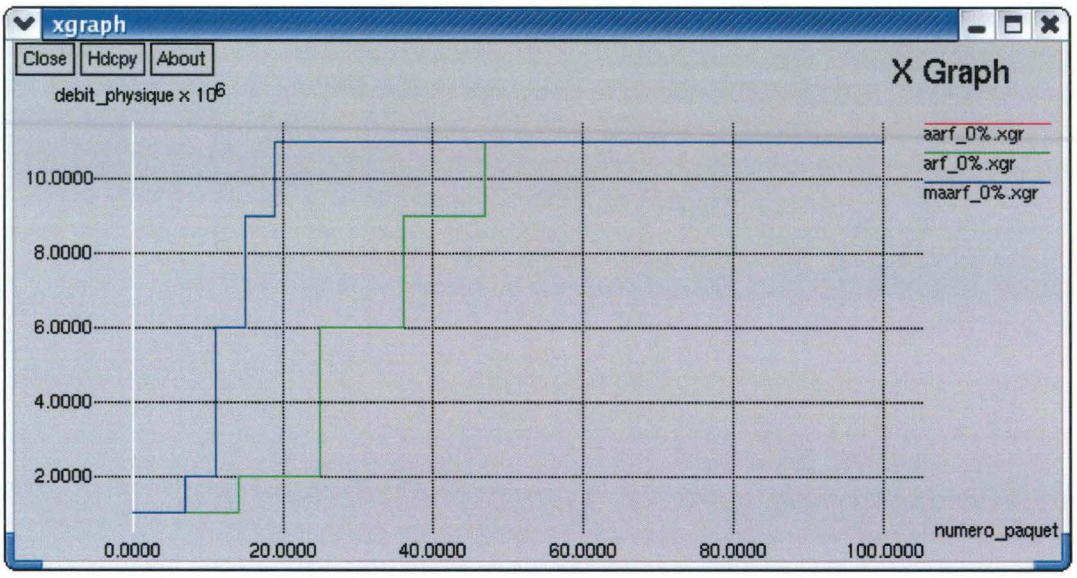

Figure 4.2 : Débit théorique en absence d'erreur

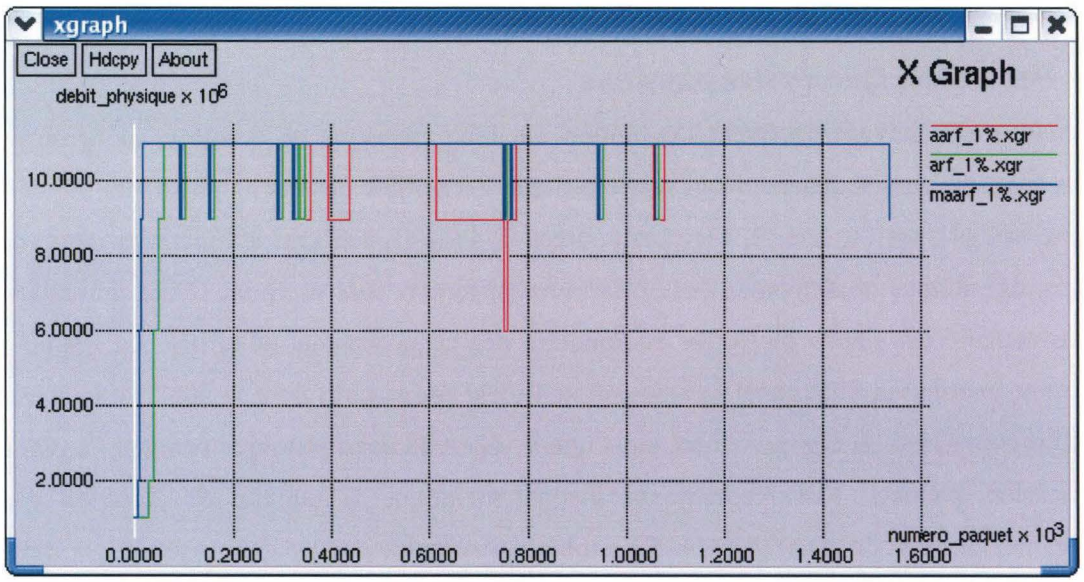

Figure 4.3 : Débit théorique pour une transmission avec un taux d'erreur = 1%

Dans le cas d'un canal peu bruité (voir *Figure* 4.3), la dynamique de montée en débit lors d'une transmission d'une grande plage de données (1400 paquets) par *MAARF* et *ARF* est plus importante que celle de l'*AARF*. Ceci s'explique par le fait que *AARF* rencontre des difficultés pour passer à un débit plus important vu qu'il implémente un algorithme *backoff* pour lutter contre les remontées inutiles. Si une erreur est survenue *AARF* met donc du temps pour corriger sa décision. Ainsi pour le paquet numéro 700, le débit de transmission utilisé à cet instant par *MAARF* et *ARF* est de 11Mbps, par contre il est seulement de 6Mbps dans *AARF*.

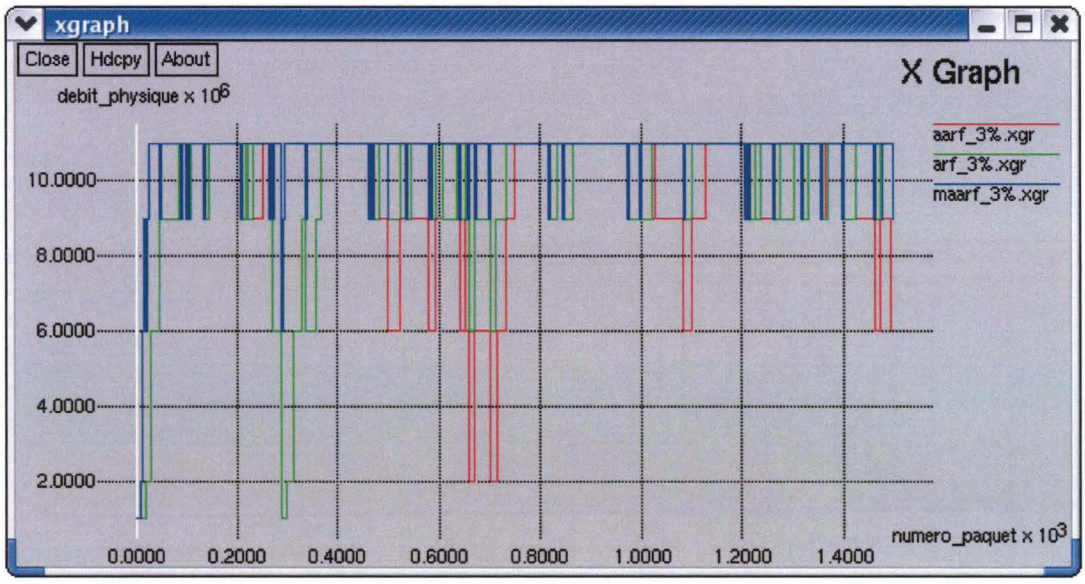

Figure 4.4 : Débit théorique pour une transmission avec un taux d'erreur = 3%

Les *Figures* 4.4, 4.5 et 4.6 montrent la différence élevée entre les trois mécanismes lorsque le taux d'erreur canal atteint respectivement 3, 5 et 7%. Le nouvel algorithme adapte des débits plus importants, tandis que les valeurs des débits par l'application d'*AARF* et d'*ARF* sont très basses.

Exemples :

- Pour un taux de perte introduit égal à 3% (voir *Figure* 4.4), la différence en débit physique employé commence à être perçue entre les trois mécanismes d'adaptation du lien : pour la transmission des paquets numérotés entre 600 et 800, l'*AARF* utilise les débits 2 et 6Mbps, *ARF* adopte fréquemment les débits 6 et 9Mbps, tandis que le *MAARF* s'appuie sur des débits plus importants (9 et 11Mbps).

- Dans la *Figure* 4.5 : le paquet N° 1200 est transmis avec un débit de 9Mbps dans *MAARF* tandis que le débit d'émission adopté par *ARF* est égal à 6 Mbps et celui adopté par *AARF* est égal à 2Mbps. Le débit physique moyen de sortie pour *MAARF* est ainsi amélioré d'environ 33% par rapport à *ARF*.

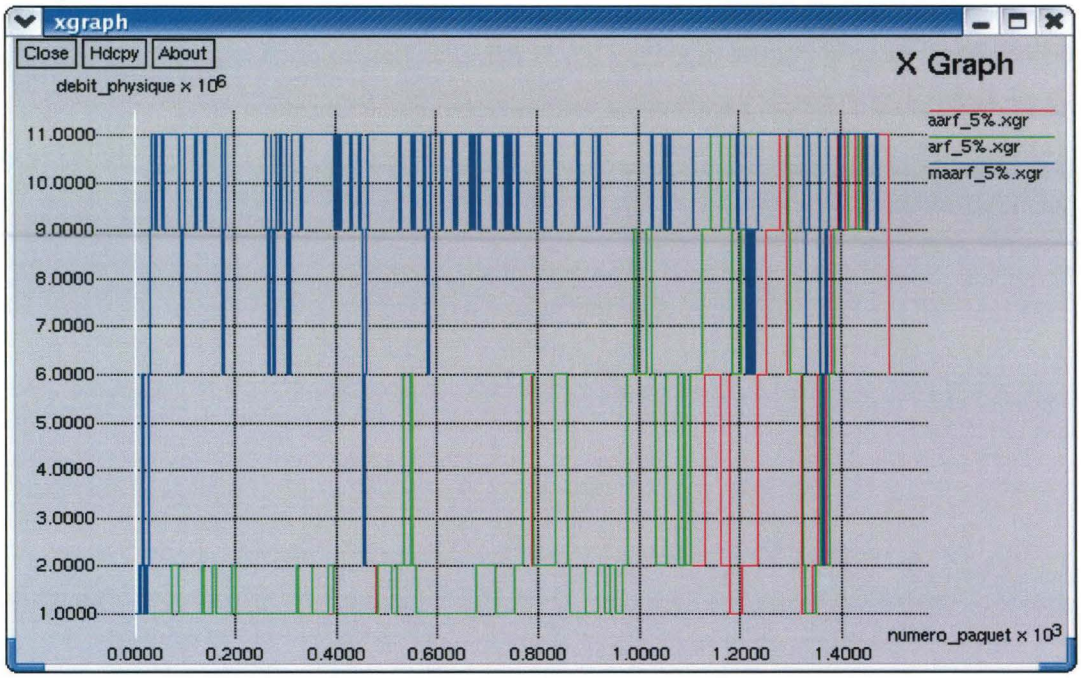

Figure 4.5 : Débit théorique pour une transmission avec un taux d'erreur = 5%

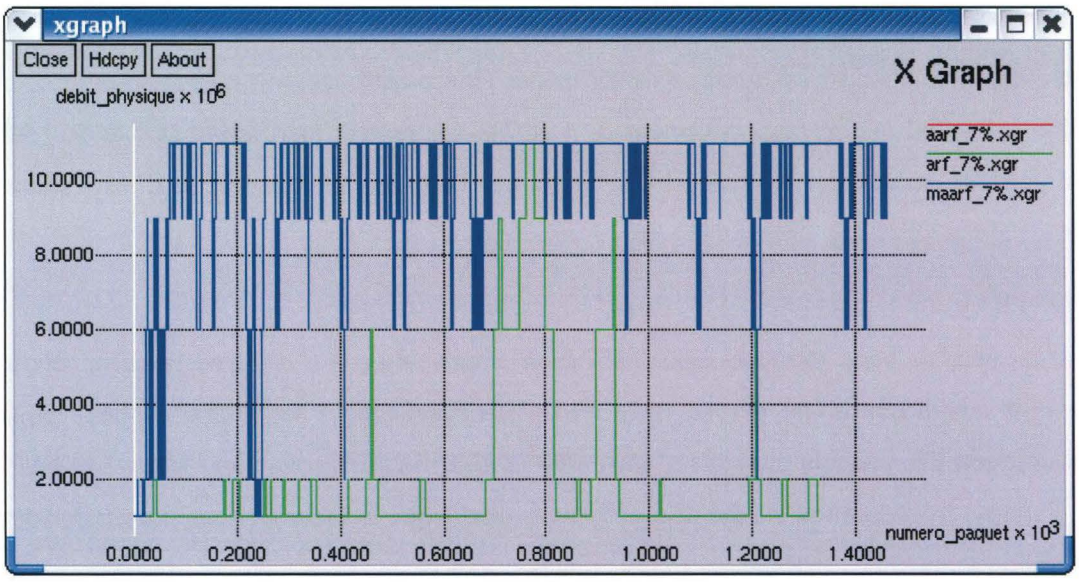

Figure 4.6 : Débit théorique pour une transmission avec un taux d'erreur = 7%

- Lorsque le taux de perte atteint 7% (voir *Figure* 4.6), le gain de la nouvelle technique est senti dès les 200 premiers paquets. Pendant cette période, le débit adopté par cette dernière oscille entre 6 et 11Mbps, tandis que les transmissions des deux autres mécanismes se déroulent avec un débit ne dépassant pas les 2Mbps.

C. Débit réel observé sur le canal

Comme deuxième série de test, nous effectuons des simulations utilisant une taille fixe de paquets et des nœuds immobiles pour différents taux de perte dans le canal. Subséquemment, dans un deuxième cas, nous conduisons d'autres simulations où les nœuds sont munis d'un mouvement uniforme à vitesse constante. Les courbes qui suivent représentent les débits réels de chacun des algorithmes récoltés à partir des fichiers *'trace'* du simulateur *NS*-2.

i. Cas de stations immobiles

Dans un premier scénario nous effectuons une transmission de données sans erreurs entre les stations. Comme illustré dans la *Figure* 4.7, le comportement des trois algorithmes est similaire en l'absence d'erreurs. La seule différence du *MAARF* par rapport aux autres mécanismes est sa vitesse d'adaptation à l'état instantané du canal. En effet, dès la première seconde de transmission, le débit moyen du lien est beaucoup plus important (à t = 10s, D = 3.78Mbs) contrairement à *ARF* et *AARF* (à t = 10s, D = 2.8Mbs). Les courbes montrent donc que la dynamique du nouvel algorithme, pour s'adapter à un canal sans erreur (engendrant de meilleures conditions) est plus rapide que les autres techniques usuelles.

Dans un deuxième scénario le taux d'erreur est fixé à 1% (voir *Figure* 4.8). L'algorithme *MAARF* atteint évidement un débit réel élevé (3.5Mbps) plus rapidement (au cours de la première seconde) que celui d'*ARF* et d'*AARF*. Les trois courbes sont similaires en régime permanent. Des limitations de l'*AARF* ont été constatées et sont dues à un retard supplémentaire introduit par l'algorithme du *Backoff* en majorant le nombre d'acquittements positifs nécessaires pour chaque remontée.

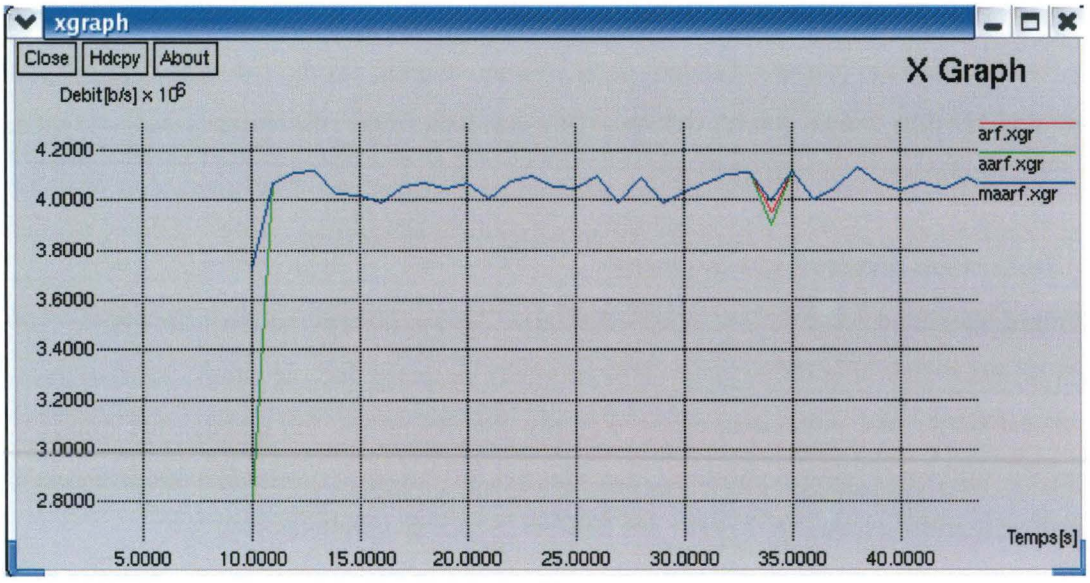

Figure 4.7 : Débit réel pour une transmission sans erreur

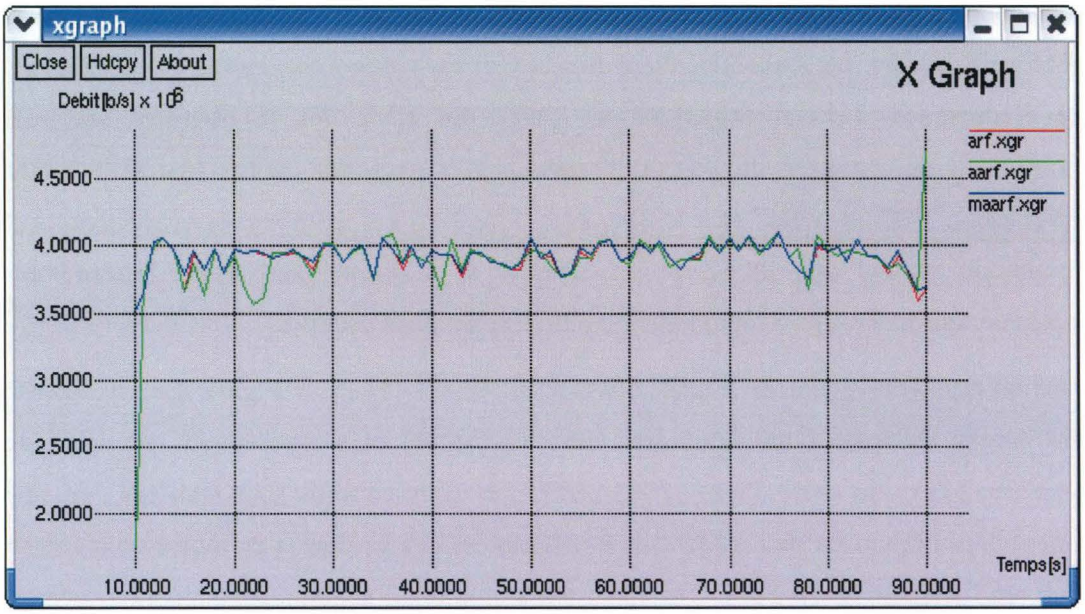

Figure 4.8 : Débit réel pour une transmission avec un taux d'erreur = 1%

Les *Figures* 4.9 et 4.10 illustrent les résultats de simulation des troisième et quatrième scénarii, où les taux d'erreurs respectifs sont de l'ordre de 3% et 5%. Ces résultats illustrent les très bonnes performances de *MAARF*. Nous constatons aussi que *MAARF* atteint toujours un débit réel moyen de 3.5Mbps et essaie de le conserver jusqu'à la fin de la transmission. En effet,

le débit de sortie mesuré varie entre 3 et 3.8Mbps, contrairement aux autres algorithmes qui effectuent des remontées et des descentes inutiles et non adaptées aux conditions instantanées du canal. Ainsi, le débit réel enregistré par ces algorithmes pour un taux de perte égal à 5%, varie essentiellement entre 0.5 et 3Mbps (trop loin de la plage des débits réels assurés par *MAARF*).

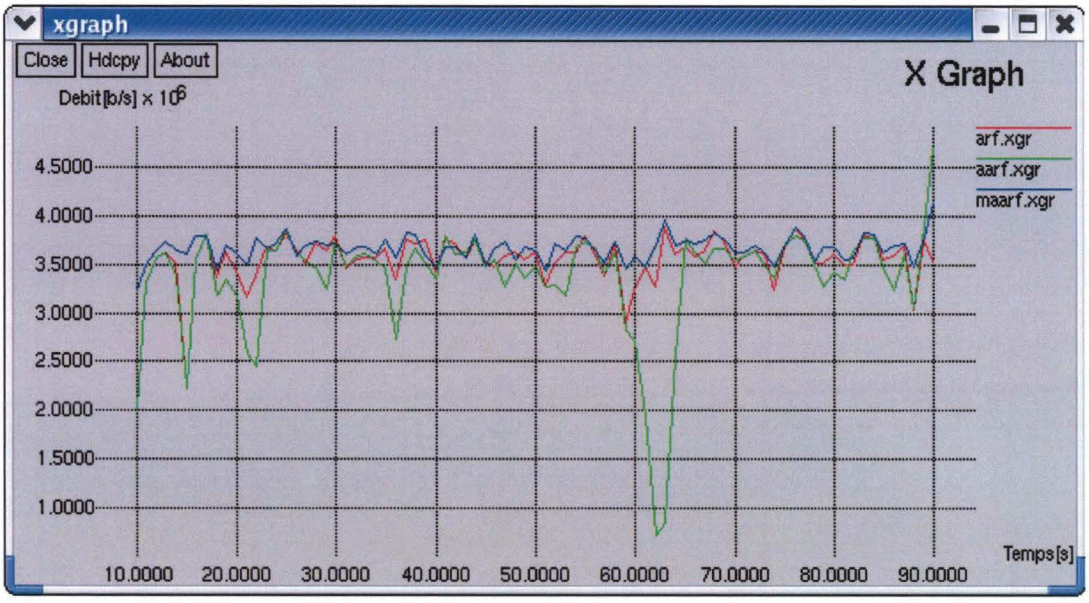

Figure 4.9 : Débit réel pour une transmission avec un taux d'erreur = 3%

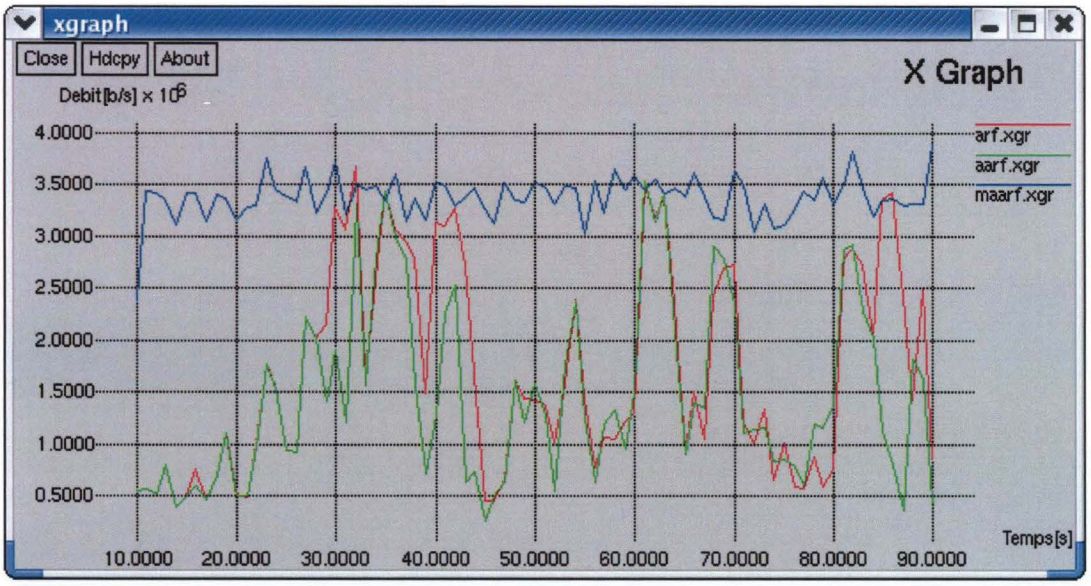

Figure 4.10 : Débit réel pour une transmission avec un taux d'erreur = 5%

Dans les deux scénarii ci-dessous, le taux d'erreur augmente et atteint 7% et 10% comme illustré respectivement dans les *Figures* 4.11 et 4.12. L'algorithme *MAARF* réagit mieux aux erreurs de canal et réussit à adopter le débit instantané adéquat (≈3Mbps), alors que *ARF* et *AARF* signalent un comportement presque stable avec une valeur de débit très basse (<1Mbps).

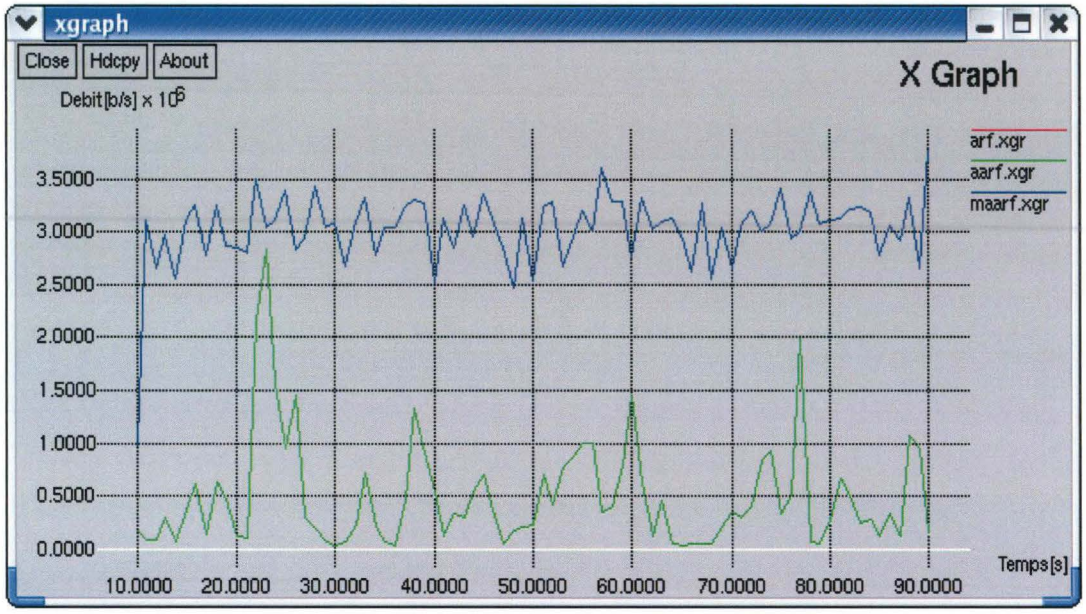

Figure 4.11 : Débit réel pour une transmission avec un taux d'erreur = 7%

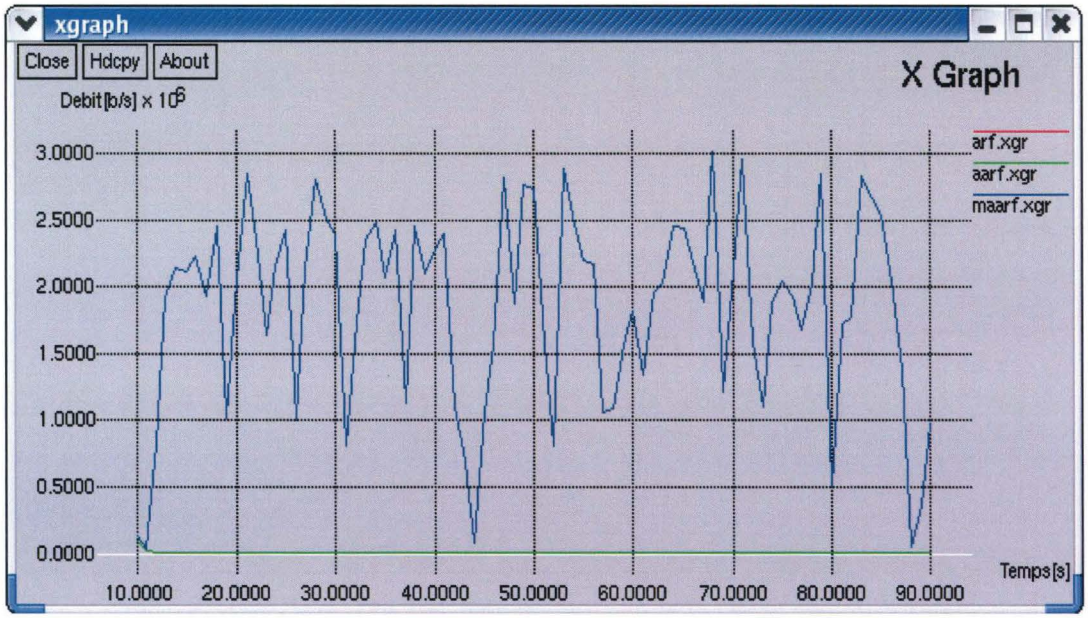

Figure 4.12 : Débit réel pour une transmission avec un taux d'erreur = 10%

Aussi, pour ces deux derniers tests réalisés, nous remarquons que les courbes de débit réel des deux mécanismes *ARF* et *AARF* sont confondues. Ceci est la conséquence directe du fait qu'ils s'appuient seulement sur la nature des acquittements retournés pour guider leur choix du débit physique à déployer. En conséquence, ces résultats traduisent l'insuffisance des décisions prises uniquement sur l'exploitation du nombre d'acquittements positifs des paquets transmis.

ii. Cas de stations mobiles

Dans cette partie nous menons des simulations dont la configuration est basée sur des nœuds mobiles du réseau sans fil. En effet, en se déplaçant sur une surface de 745x745 mètres carrés avec une vitesse réduite, la qualité du lien varie en fonction du temps et de la mobilité des stations. La portée de la liaison peut être maintenue jusqu'à une distance séparant les mobiles de 300 mètres. Au-delà de cette valeur la communication est perdue.

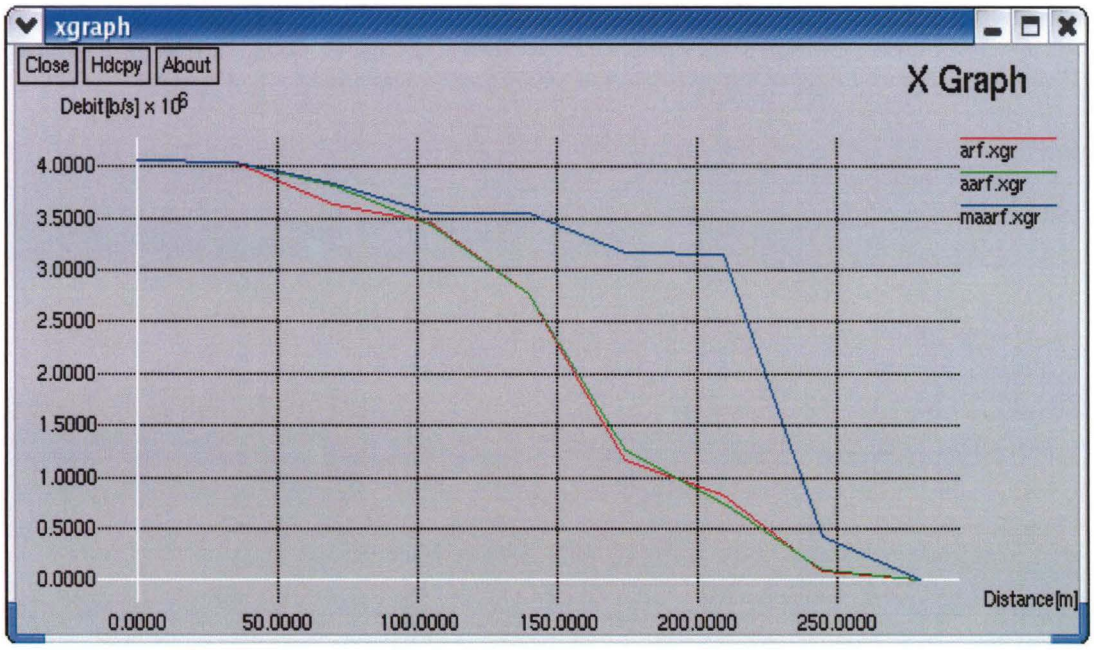

Figure 4.13 : Débit moyen en fonction de la distance

Nous présentons dans la *Figure* 4.13 le débit réel enregistré en fonction de la distance entre les stations. Nous notons que le taux d'erreur augmente linéairement en fonction de la distance. Au delà des 200 mètres, les débits moyens des trois algorithmes tendent à s'annuler, mais le *MAARF* maintient un débit élevé durant une période de temps plus grande et une distance assez étendue (jusqu'à $d = 220$ mètres). Les débits générés par les deux autres algorithmes décroissent vers des

valeurs beaucoup plus basses bien avant (environ 150 mètres). Aussi ces deux techniques actuelles convergent vers les mêmes valeurs dès que la distance entre les stations dépasse les 100 mètres. Ainsi par exemple, pour une distance d = 200 mètres, le débit moyen de *MAARF* est égal à 3.2Mbps alors que le débit instantané de *AARF* et *ARF* est seulement égal à 0.9Mbps.

D. Débit moyen observé

Dans cette section, nous menons plusieurs simulations différentes et calculons par la suite les débits moyens à partir des débits réels de sortie enregistrés sur le canal. Ces simulations sont réalisées, premièrement en fonction de la taille des paquets, ensuite en fonction des taux de perte. Dans une première expérience nous transmettons des paquets de données dont la taille varie entre 500 et 2000 octets. Le taux de perte est fixé à 5%. Le graphe de la *Figure* 4.14 montre que l'algorithme *MAARF* obtient de très bonnes performances en terme de débit moyen de sortie comparé à celles obtenues par *ARF* et *AARF* pour différentes tailles de paquets.

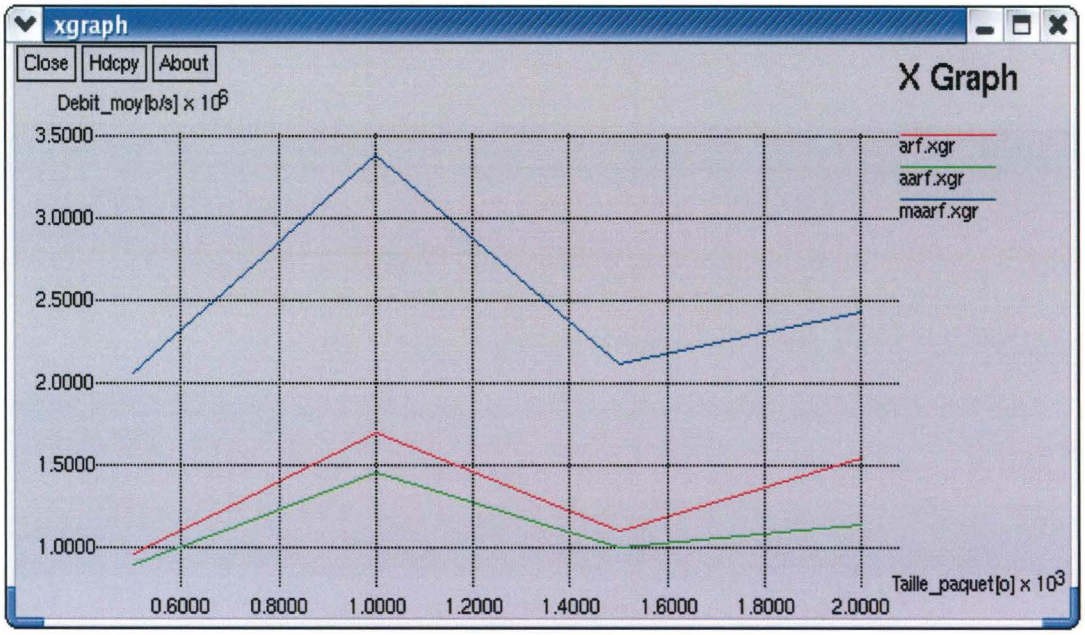

Figure 4.14 : Débit moyen en fonction de la taille des paquets

Nous constatons aussi que le meilleur débit moyen enregistré pour les trois algorithmes est celui pour une taille de paquet égale à 1000 octets (avec un débit moyen dans *MAARF* de l'ordre de 3.4Mbps, de 1.4Mbps pour *AARF* et de 1.7Mbps pour *ARF*). Ceci s'explique par le fait que la taille du paquet appropriée, au niveau de la sous-couche *MAC* de l'IEEE 802.11 est de 1200

octets (après l'ajout des entêtes requis). Ce fait limite les performances globales enregistrées pour les systèmes de transmission lors de l'utilisation des paquets de taille supérieure à 1000 octets, puisque le mécanisme de la fragmentation et de réassemblage est enclenché et s'avère très coûteux en terme de temps de traitement (chaque fragment doit être acquitté séparément).

Dans une seconde expérience nous montrons les débits moyens de sortie, schématisés dans la *Figure* 4.15, des trois algorithmes d'adaptation de lien en fonction du taux d'erreur introduit. La taille des paquets est fixée à 1000 octets.

Nous observons des améliorations très nettes apportées au débit moyen général lors d'une transmission de données : de l'ordre de 9% dans le cas d'un canal peu bruité. Cette valeur devient de plus en plus importante et considérable (elle atteint 100% et plus) lorsque la qualité du canal devient médiocre. Nous observons également que la nouvelle technique préventive essaie au mieux de conserver un meilleur débit de transmission et de l'adapter même aux conditions infectes du canal. Cette capacité n'est pas présente dans les algorithmes actuels qui sont des algorithmes réactifs à des événements déjà passés lors d'une transmission de données (après une perte de paquet, une retransmission, etc.).

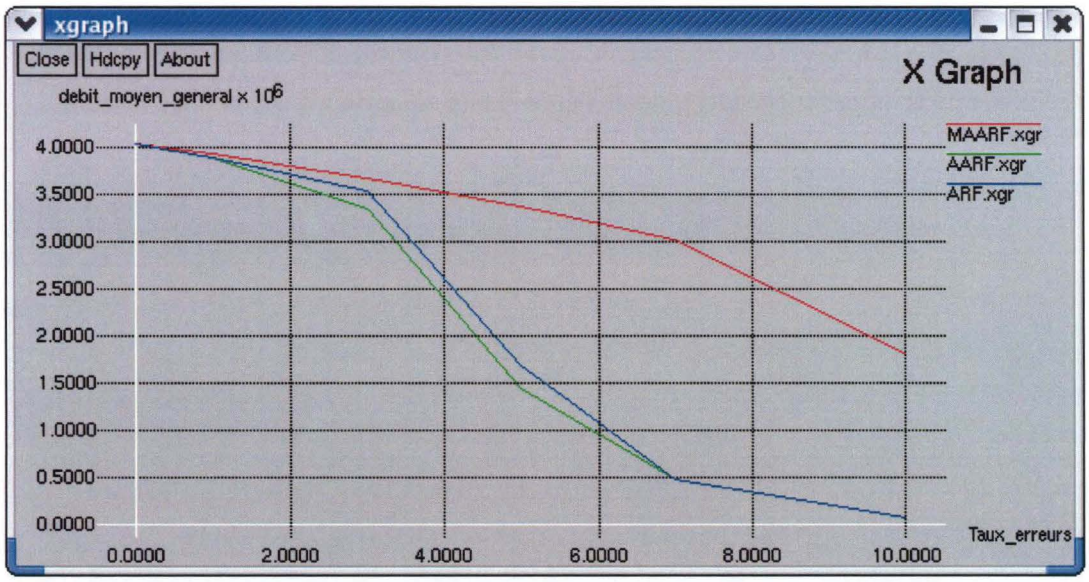

Figure 4.15 : Débit moyen en fonction du taux d'erreur

E. Taux de perte des paquets

Les courbes reportées dans la *Figure* 4.16 résument les résultats fournis par toutes les courbes précédentes. Elles schématisent le taux de perte des paquets obtenu (nombre de paquets perdus par rapport au nombre total des paquets transmis), en fonction du taux d'erreur canal introduit. Ainsi, à partir de cette figure, nous affirmons que *MAARF* apporte de meilleurs résultats : lorsque le taux d'erreur du canal croît et atteint des valeurs élevées (égal à 10%), les deux algorithmes *ARF* et *AARF* perdent une importante quantité de données (24% des paquets émis), tandis que *MAARF* garde quasiment la totalité de ces données intacte avec un taux de pertes égal à 0.2%. Ce résultat reflète directement l'efficacité de la nouvelle technique face au choix du meilleur débit adéquat aux conditions instantanées du canal de communication, ainsi que sa rapide réaction contre les changements brusques de ces conditions. La différence entre les taux de perte canal, lors de l'implémentation des algorithmes actuels, est remarquable et peut atteindre une valeur 12 fois plus grande que celle obtenue par *MAARF*. Les résultats associés aux taux de perte obtenus par les deux techniques *AAR* et *AARF* sont identiques. Ceci s'explique par les mauvaises décisions adoptées par les deux algorithmes bien qu'ils emploient une sélection différente des débits physiques instantanés.

En conclusion, *MAARF* minimise le taux de perte des données transmises en employant des débits plus adéquats, et par conséquent, décrémente le taux de retransmission dans le canal.

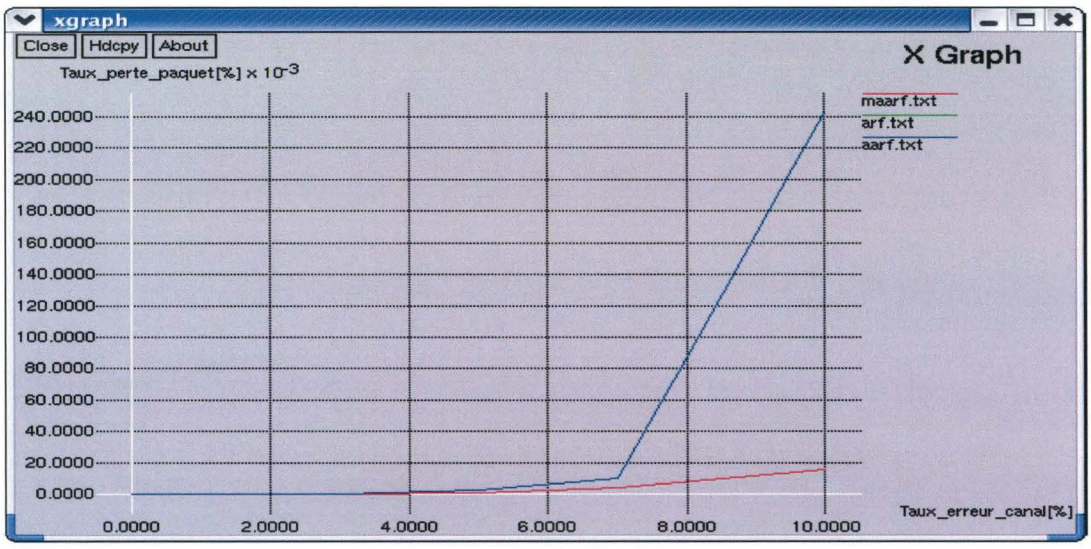

Figure 4.16 : Taux de perte des paquets en fonction du taux d'erreur canal

F. Résumé des simulations menées

Nous avons pu évaluer les performances de la nouvelle technique de contrôle de débit dénommée *MAARF* et montrer son efficacité par rapport aux deux algorithmes *ARF* et *AARF* les plus déployés actuellement. Ces simulations ont été conduites à travers un ensemble de scripts de scénarii procédés sur l'outil de test des réseaux *NS-2* (*Network Simulator version 2.27*).

Les résultats enregistrés permettent de confirmer la robustesse ainsi que l'adaptabilité du nouveau mécanisme aussi bien aux changements lents qu'aux variations rapides de la qualité du canal de transmission.

IV. Conclusion et synthèse

L'intérêt de cette première partie portait, tout d'abord, sur l'étude des mécanismes de contrôle de débit dans les réseaux WiFi IEEE 802.11. Nous avons ensuite, proposé et réalisé un nouvel algorithme baptisé *MAARF* (*Modified Auto Rate Fallback*). Dans cet algorithme, une nouvelle variable décisionnelle appelée *RTT* (*Round Trip Time*) est ajoutée coopérant avec le paramètre de base (nature et nombre des acquittements retournés). L'insertion de ce paramètre *RTT* a pour objectif d'effectuer une bonne estimation du canal (observation de l'état de canal à chaque trame transmise), et de choisir en conséquence le débit adéquat.

En se basant sur les résultats de simulation sous la plateforme *NS-2*, nous avons observé des améliorations nettes de débit à travers ce nouvel algorithme. Au niveau du débit physique le mécanisme proposé *MAARF* offre des valeurs plus importantes de l'ordre de 17% jusqu'à 100% en les comparant avec celles des algorithmes classiques. Il offre aussi des débits moyens de réception allant du simple au double dans le cas d'un canal très bruité.

Débit moyen en 10^6 Mb/s				*Amélioration en %*
Taux d'erreur	*ARF*	*AARF*	*MAARF*	*MAARF / AARF*
0%	*4.0335*	*4.0339*	*4.0461*	*0.3%*
1%	*3.8930*	*3.8877*	*3.9225*	*0.8%*
3%	*3.5419*	*3.3612*	*3.6719*	*9%*
5%	*1.6992*	*1.4564*	*3.3826*	*>80%*
7%	*0.4914*	*0.4914*	*3.0195*	*>80%*
10%	*0.0774*	*0.7740*	*1.8273*	*>80%*

Tableau 4.2 : Tableau récapitulatif des améliorations en débits moyens du MAARF

Le *Tableau* 4.2 présente un résumé récapitulatif des améliorations sur les débits moyens du *MAARF* par rapport aux algorithmes existants (*ARF et AARF*) pour différents taux d'erreur. Les améliorations apportées réduisent considérablement le taux de perte lorsque le taux d'erreur atteint des valeurs importantes (12 fois lorsque le taux d'erreur est égal à 10%).

En résumé, cette nouvelle approche permet de répondre aux problèmes usuels des algorithmes de contrôle de débits actuels et remplit les objectifs ainsi que les performances souhaités (à savoir satisfaire la transmission d'applications multimédia à *QoS* exigée en terme de débit réel).

Cette première contribution, ainsi que les améliorations apportées au schéma d'adaptation du lien dans les réseaux 802.11 sans fil, ont été publiées dans une conférence internationale « *The 2nd IEEE International Conference on Technologies, Mobility and Security* » (*NTMS'08*) [74].

CHAPITRE

5 Révision du schéma d'accès EDCA adopté par 802.11e

I. Contexte et Objectifs

Dans le cadre des *WLAN* de type 802.11 qui envahissent de nos jours tous les espaces que l'on côtoie, plusieurs solutions appuyant le support de la qualité de service ont été développées, en particulier celles qui concernent directement la différenciation de service (les plus intéressantes ont été signalées dans le chapitre 2). La plus pertinente est le nouvel algorithme ***EDCA*** (*Enhanced Distributed Channel Access*) implémenté dans la nouvelle norme 802.11e du standard de l'IEEE. Dans ce chapitre nous présentons un nouvel algorithme basé sur l'*EDCA* et améliorant le schéma de gestion du trafic *QoS* en introduisant une nouvelle classification entre les nœuds mobiles. Cette révision inclut le support de deux types de priorités lors de la transmission des paquets : la première s'appuie sur la différenciation entre les types de trafic déjà utilisés par l'*EDCA*. La seconde est une nouvelle classification qui concerne l'ordonnancement des stations actives du réseau. Ainsi, cette technique autorise la transmission des paquets prioritaires en se basant sur l'état de la station et sans être interrompue par l'émission de paquets de classes moins importantes. L'état d'une station mobile est évalué de façon dynamique à travers un récent historique des communications accomplies par cette station.

Dans le chapitre 2 de ce manuscrit nous avons présenté un état de l'art généralisé sur les travaux pertinents déjà réalisés sur la différenciation de services dans les réseaux IEEE802.11 [26, 27, 28, 29, 30, 31]. Dans le présent chapitre, nous présentons la méthode d'accès *EDCA* la plus récente sur laquelle notre contribution se basera. Ensuite, nous exposons la nouvelle

technique d'accès conçue pour améliorer l'actuel schéma de transmission, en spécifiant ses paramètres de décision et leurs rôles. Enfin, une évaluation des performances ainsi qu'une comparaison avec la technique courante est réalisée afin de prouver les apports de la nouvelle révision.

II. Enhancement Distributed Channel Access (EDCA)

1. Définition du EDCA

Nous présentons la nouvelle version d'*EDCA* qui est basée sur le modèle *EDCF* (*Enhanced Distributed Coordination Function*) antérieur. Dans [42], les auteurs présentent plusieurs changements de la nomenclature:

- ***EDCF*** devient ***EDCA*** (*Enhanced Distributed Channel Access*).
- Huit catégories de trafic ***TCs*** (*Traffic Category*) du modèle *EDCF* deviennent quatre catégories d'accès ***ACs*** (*Access Category*).
- ***HCF*** (*Hybrid Coordination Function*) est converti en ***HCCA*** (*HCF Controlled Channel Access*).

Dans [27], huit priorités du trafic sont employées (avec leurs propres files d'attente) qui sont indiquées dans IEEE 802.1D [43]. Cependant, dans [42, 28] quatre files d'attente sont utilisées à cause de la transformation des huit priorités de trafic à seulement quatre catégories d'accès *ACs*. La procédure *EDCA* utilise uniquement l'algorithme du *backoff* exponentiel « classique » ***BEB*** (*Binary Exponential Backoff*) et n'a donc plus besoin d'un ***PF*** (*Persistance Factor*) utilisé avec *EDCF* dans [26, 44]. Ainsi, comme dans *EDCF*, le compteur de *backoff* est arrêté si le canal est occupé par la transmission d'un autre paquet. Il est décrémenté si le canal devient libre pour au moins un ***AIFS*** (*Arbitration Inter-Frame Space*) temps proportionnel à la classe de priorité du paquet actuel. Afin de résoudre le problème des collisions répétées dans le médium et causées par la transmission simultanée de plusieurs paquets au même instant, l'incrémentation des fenêtres de contention « *Contention Window* » CW_{min} est activée et doit être choisie comme l'indique l'*Equation* 1 suivante (les valeurs de CW_{min} et CW_{max} varient d'un type de trafic *AC* à un autre).

$$eq.1 \qquad newCW_{min}[AC] = (2 * oldCW_{min}[AC]) + 1$$

Comme illustré dans la *Figure* 5.1, différents *ACs* sont installés sur une même station avec leurs propres ensembles de paramètres d'accès (*AIFS* et *CW*) et leurs propres valeurs du mécanisme *backoff* (CW_{min} et CW_{max}). Les catégories d'accès (*ACs*) sont classifiées suivant quatre types :

- *AC_VO* : pour les flux *Voix* (priorité 0)
- *AC_VI* : pour ceux qui transmettent de la *Vidéo* (priorité 1)
- *AC_BE* : pour ceux responsables du trafic *Best Effort* (priorité 2)
- *AC_BK* : pour les trafics *Background* (priorité 3)

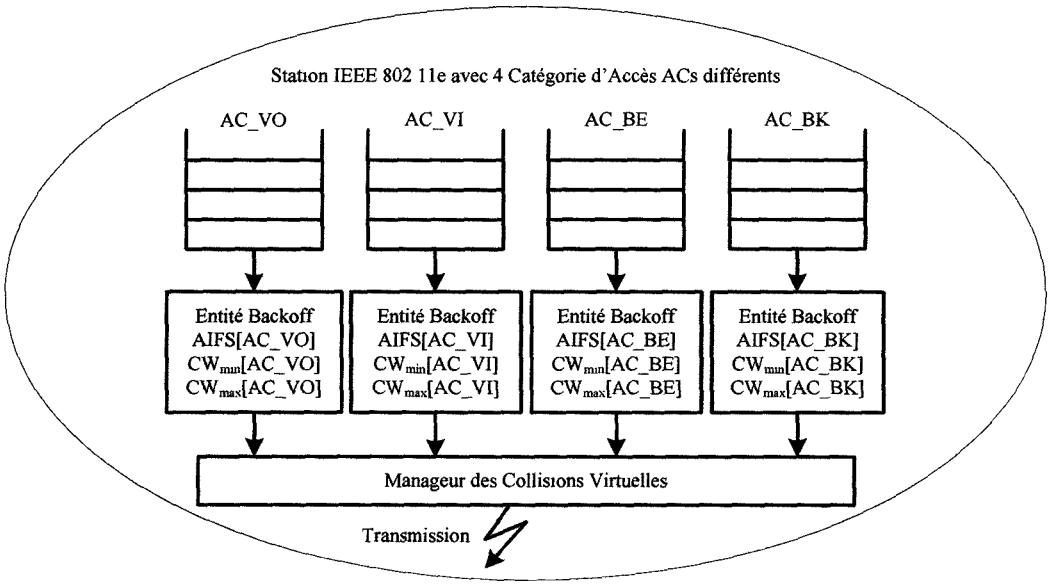

Figure 5.1 : Structure de l'EDCA adopté par le protocole MAC 802.11e

La méthode d'accès *EDCA* a installé quatre files d'attente par station. Ceci peut nous mener à une collision si nous permettons à deux *ACs* d'envoyer en même temps. Cette collision est résolue par un programmateur virtuel ***VCH*** (*Virtual Collision Handler*) qui accorde l'accès pour l'*AC* ayant la priorité la plus élevée. La différentiation de service des différents paquets est assurée par une classification inversement proportionnelle à leurs priorités *i* des valeurs du paramètre d'accès au médium *AIFS*[*i*]. Cette classification est exprimée dans l'*Equation 2* qui suit.

$$eq.2 \qquad AIFS[0] < AIFS[1] < AIFS[2] < AIFS[3]$$

Rappelons que $PIFS < AIFS[i]$ pour $i = \{0, 1, 2, 3\}$. La même règle donnée dans l'Equation 2 est employée pour l'ordre de classification des valeurs de $CW_{min}[i]$ et de $CW_{max}[i]$ respectivement pour chaque *AC*.

En résumé, l'*EDCA* qui contrôle l'accès au canal de transmission avec différenciation entre les flux de données dans le IEEE802.11e, favorise la transmission des trames d'informations les plus prioritaires en les classifiant en plusieurs catégories d'accès *ACs* liées à leurs priorités respectives.

Finalement, la différenciation des flux (voir *Figure* 5.2) est assurée par le tunage des trois paramètres généraux de la couche d'accès *MAC* du 802.11 pour chaque priorité *i* suivants :

- ✓ $CW[i]$ (*Contention Windows*) composé de CW_{min} et CW_{max}.
- ✓ $AIFS[i]$ (*Arbitrary Inter Frame Space*), le temps qui sépare la transmission de deux trames consécutives dans le canal.
- ✓ $BI[i]$ (*Backoff Interval*), responsable de la variation du délai d'attente afin de prévenir et d'éviter les collisions qui peuvent se produire au cours de la transmission par d'autres trames (transmises par d'autres stations simultanément).

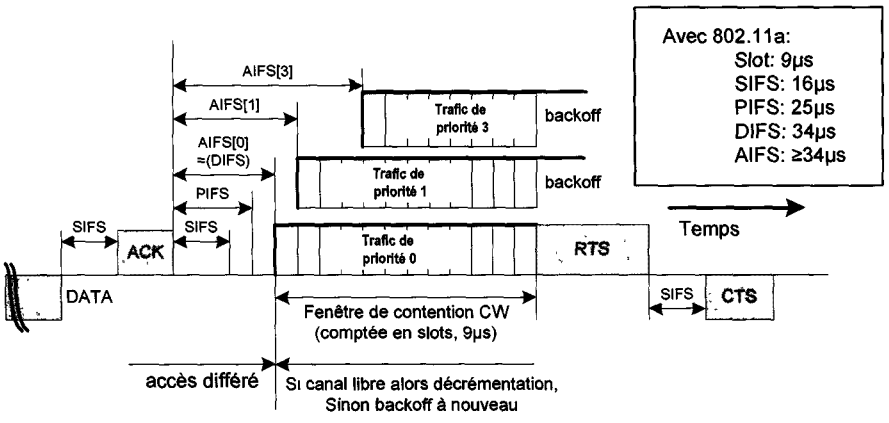

Figure 5.2 : Relation entre les temps Inter-Frame IFS dans 802.11e

2. Limites de l'EDCA

Afin de maintenir une certaine *QoS* exigée par les applications multimédias dans le protocole *MAC* du IEEE 802.11e, l'approche typique est de manipuler les trois paramètres d'accès *BI*, *CW*,

et *IFS*, en classifiant les valeurs des $AIFS[i]$, $CW_{min}[i]$ et $CW_{max}[i]$ inversement proportionnelles à leurs priorités (comme déjà illustré dans l'*Equation* 2) [45, 46].

Toutefois, nous remarquons que le mécanisme *EDCA* assure effectivement la différenciation de service pour les différents *ACs* des flux de la même station, mais pas pour les différentes stations du même réseau (accès parallèle au canal par plusieurs flux de la même classe de priorité par plusieurs stations du réseau). En d'autres termes, il n'y a aucune distinction entre les *ACs* de même priorité et issus de deux ou plusieurs stations au même instant. Ainsi l'*EDCA* élimine les collisions dues à la concurrence produite par la transmission des paquets de classes différentes, et de ce fait il favorise la transmission des trafics dédiés pour des applications multimédias et à *QoS* exigée. Cependant, ce mécanisme n'a pas éliminé les collisions dues à la concurrence produite par la transmission de plusieurs paquets appartenant à la même classe de service issus de deux ou plusieurs stations différentes. Il a plutôt encouragé cette concurrence en produisant les mêmes valeurs des paramètres liés à l'accès au canal de transmission de tous les paquets du même *AC*. Nous découvrons également que le standard l'IEEE 802.11e produit un modèle de service proportionnel désiré inefficace. En effet, lorsque les classes de priorité haute sont inactives lors d'une transmission, les classes de priorité faible ont toujours besoin d'attendre un délai élevé *AIFS* (correspondant à leur priorité) avant que l'intervalle *Backoff* (également très grand pour ces types de trafic) commence à décroître jusqu'à zéro. De plus, cet intervalle de *Backoff* est très important en valeur par rapport à celui des classes de haute priorité.

Nous rappelons que le fonctionnement général du *CSMA/CA* (*Carrier Since Multiple Access with Collision Avoidance*), présenté dans le premier chapitre, est d'attendre un délai aléatoire appelé *Backoff* (calculé en fonction de *CW* et du nombre de retardements déjà effectués au sein du réseau WiFi) après l'expiration du premier délai fixe *DIFS* qui sépare l'envoi de deux trames (dans notre cas le *DIFS* est remplacé par plusieurs *AIFS* qui dépendent des classes de priorité et ordonnés comme expliqué précédemment). En conséquence, le débit global se dégrade. De plus, les paramètres responsables de cette inefficacité ont besoin d'être revus en s'accordant à la charge totale du trafic (et même à la charge instantanée si nous pouvons la mesurer). Ceci introduit aussi de l'incertitude dans les performances de l'IEEE 802.11e. Nous proposons dans la section suivante une nouvelle approche permettant de remédier à ces problèmes.

III. Nouvelle approche de classification inter-stations

Le standard 802.11e, avec son algorithme *EDCA*, a bien fait ses preuves pour la différenciation de service sur les réseaux WiFi et a bien remédié aux problèmes de collisions virtuelles (collisions entre deux *ACs* différents et appartenant à la même station). Cependant il n'a pas remédié à la concurrence inter-stations lorsque deux *AC*[*i*] de même priorité et appartenant à deux stations différentes souhaitent émettre simultanément. Similairement, le standard 802.11e pénalise les trafics de faible priorité lorsque la station n'émet pas de trames de haute priorité. Dans ce qui suit, nous proposons une nouvelle technique remédiant aux inconvénients de l'*EDCA* tout en gardant les avantages du 802.11e.

Dans le standard 802.11e les catégories d'accès *AC*[*i*] d'une même station sont différenciées en faisant varier leur temps d'attente pour accéder au canal : le trafic le plus prioritaire attendra moins que celui qui est moins prioritaire pour conquérir le canal. Cependant deux catégories d'accès de même niveau de priorité appartenant à deux stations différentes peuvent se concurrencer sur l'accès au canal, surtout si leurs paramètres sont identiques (même *AIFS*[*i*], *CW*[*i*], etc.).

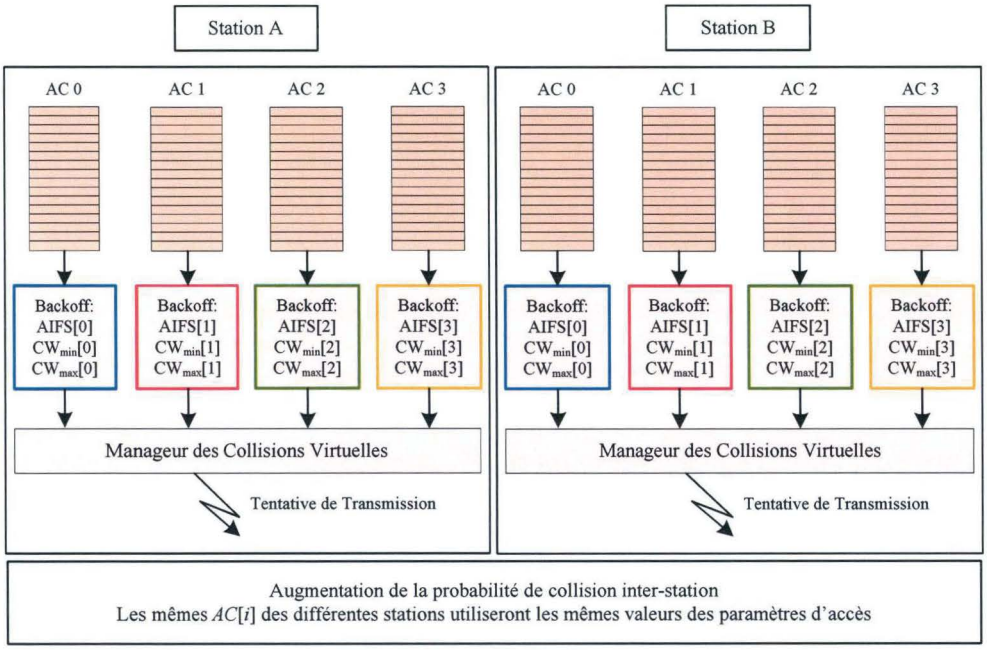

Figure 5.3 : Collisions entre stations par l'EDCA classique du 802.11e

Nous avons ajouté dans un premier temps une nouvelle classification (priorité) entre les différentes stations du réseau pour essayer de surmonter ce problème, mais sans influer (ou modifier) la méthode d'accès au canal (algorithme *CSMA/CA*). La *Figure* 5.3 illustre mieux le problème posé. Les trafics issus de deux *ACs* de même type de trafic utilisent les mêmes valeurs des paramètres d'accès, et par conséquent, la probabilité d'avoir des collisions est beaucoup plus importante.

1. Critères de la nouvelle classification

Nous introduisons un autre ordonnancement entre les stations actives sur le réseau. La question qui se pose maintenant est sur quel critère allons nous départager les stations ? C'est à dire quelle sera la référence de sélection pour juger qu'une station est prioritaire par rapport à une autre ? Il est aussi nécessaire de prendre en considération que cet ordonnancement doit gérer des flux appartenant à la même classe de service et transmis simultanément par plusieurs stations. Pour départager deux $AC[i]$ appartenant à deux stations différentes, nous avons opté pour s'appuyer sur l'historique de la transmission pour chaque station et le considérer comme facteur de décision.

Notre choix s'est porté sur l'historique des trafics transmis sur le réseau afin de mieux caractériser les stations en termes de qualité des paquets déjà communiqués. Pour être plus explicite, une station qui émet beaucoup de trames de type multimédias (*Voix* ou *Vidéo*) sera prioritaire pour accéder au canal par rapport à une autre station qui en émet peu ou adopte des flux non prioritaires. Nous introduisons ainsi la notion d'importance « *Multimédia* » d'une station par rapport aux autres nœuds du réseau actuel. Ce nouvel ordonnancement des nœuds mobiles permettra de favoriser, pour un même type d'*AC*, l'accès privilégié d'une station marquée par plusieurs transmissions multimédias plutôt que celles qui sont moins sollicitées à cause de leurs transmissions de paquets de classes moins prioritaires accomplies.

De même, le mécanisme proposé doit gérer le caractère dynamique des stations. En effet une station à caractère plutôt multimédia peut changer de propriété et accepter au cours des prochaines transmissions un trafic de classe inférieure. Elle doit donc corriger son facteur d'importance dans le réseau actuel conformément au trafic accepté récemment, et vice et versa. En conséquence, la nouvelle classification des stations dépendra de l'historique des

communications et ne doit pas être statique comme celle déjà utilisée pour les différents types de trafics *ACs* dans l'algorithme *EDCA*.

La *Figure* 5.4 suivante illustre la réduction des collisions entre deux trames issues de deux *ACs* adjacentes et appartenant à deux stations différentes.

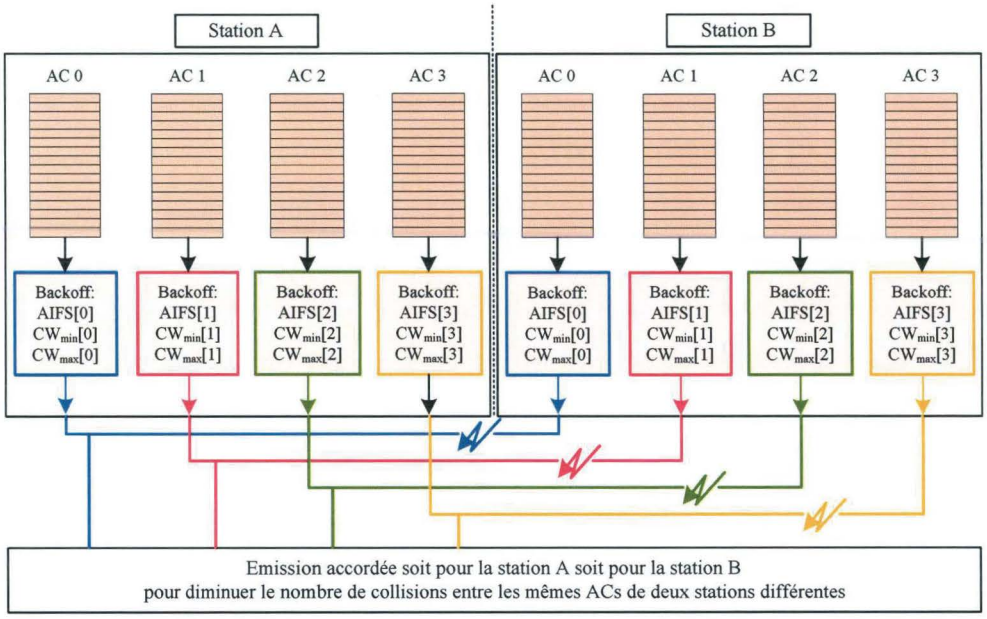

Figure 5.4 : Les collisions à résoudre par la nouvelle classification

2. Mécanisme de la nouvelle classification

Nous allons calculer pour chaque station active du réseau le nombre de trames transmises pour chaque type de flux. Ce compteur noté S_i correspond au nombre de trames déjà transmises $AC[i]$ appartenant à la classe de service i. En d'autres termes, ce nouveau paramètre S_i exprime la priorité de cette station dans le réseau. Ainsi, il influe sur le choix des valeurs des paramètres responsables de l'accès au médium. En effet, l'$AIFS[i]$ d'un type de trafic donné $AC[i]$ n'aura plus une valeur fixe mais change d'une station à une autre en fonction du nouveau compteur introduit S_i sans débordement de l'intervalle alloué. Les $AIFS[i]$ du flux de priorité i, pour n'importe quelle station du réseau, sont toujours limités en valeur entre $AIFS[i\text{-}1]$ et $AIFS[i+1]$, et ceci pour conserver la hiérarchie déjà établie entre les *ACs*. Alors, les stations actives auront une priorité j exprimant leur importance au niveau des données transmises pour la même classe de service i.

En conséquence, les flux de la même catégorie i n'admettront plus la même priorité d'accès au canal. De plus, ils ne seront plus transmis avec la même équité qu'auparavant par les différentes stations (il n'y aura plus d'accès parallèle par les stations du réseau des même flux de priorité i, et par conséquent, moins de collisions). Nous présentons dans ce qui suit notre nouvelle classification des flux de données qui seront maintenant dépendants de la priorité i du trafic ainsi que de la priorité j de la station dans le réseau.

Comme nous l'avons déjà mentionné, l'ordre des *AIFS* ne restera plus sous sa forme statique mais sera plutôt expressif dans la gestion du trafic puisque les stations de classes plus hautes auront une chance plus importante d'accéder au canal pour le même type de flux, sans négliger ou supprimer la classification déjà existante entre les différentes classes de service. Ce procédé est illustré par le nouvel ordonnancement des $AIFS[i]$ de l'*Equation 3.*

$$eq.3 \qquad AIFS[i-1][n-1] < AIFS[i][0] < \cdots < AIFS[i][j] < \cdots < AIFS[i][n-1] < AIFS[i+1][0]$$

pour $i \in \{0,\ldots,3\}$ et $j \in \{0,\ldots, n\text{-}1\}$ avec n est le nombre des priorités d'une station active dans le réseau.

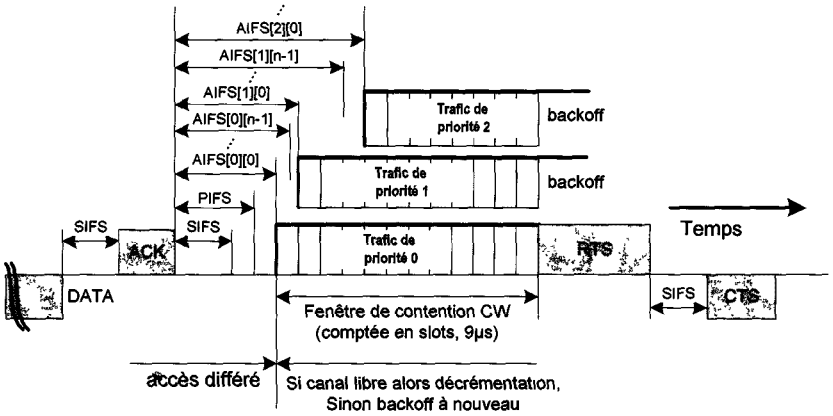

Figure 5.5 : Schéma d'accès avec la nouvelle priorité inter-station

Nous percevons, comme le montre la *Figure* 5.5 qui suit, que nous avons conservé le principe de la classification stricte en apportant les modifications seulement sur le paramètre $AIFS[i]$ et non pas sur $CW[i]$ afin d'ordonnancer les flux transmis sur le réseau selon leurs classes appropriées et leurs priorités inter-station relatives. La valeur $CW[i]$ est utilisée par toutes les stations souhaitant transmettre des paquets de type $AC[i]$ et ceci quelque soit sa priorité j

actuelle dans le réseau. Cette dernière priorité est utile uniquement lors du choix du *AIFS*[*i*][*j*] adéquat pour le prochain accès au canal.

3. Implémentation de la nouvelle classification

Pour l'élaboration et l'implémentation de ce nouvel algorithme, et afin de le valider et de le comparer avec l'*EDCA* basique par le biais des simulations réalisées sous *NS-2*, nous devons d'abord spécifier la méthode de détermination des priorités correspondantes aux stations actives dans le réseau à partir des compteurs implémentés dans chacune de ces dernières. Aussi, nous devons concevoir un dispositif pour que la nouvelle priorité soit dynamique et dépendante de l'historique récent des transmissions. Ce dispositif admet comme objectif principal d'extraire les traces de la *QoS* dans les récentes communications réalisées dans le réseau.

Un point important à noter est que n'importe quelle station dans le réseau n'a aucune idée sur les valeurs des compteurs des autres stations présentes (puisque les compteurs sont locaux). Ainsi, la prise de décision et le calcul de la priorité doivent être faits pour chaque station, uniquement en fonction des informations locales. Cette contrainte est liée à l'architecture même du réseau 802.11 (nous n'avons aucune information concrète et instantanée correspondante aux autres stations appartenant au réseau et accédant au canal). De ce fait, notre modèle sera conforme au standard IEEE802.11e et ne nécessitera aucune modification du protocole *MAC* associé. En conséquence, nous devons concevoir un modèle de création de classes inter-station et d'attribution de la priorité pour une station active dans le réseau en absence totale d'information extérieure à cette dernière.

Nous allons détailler la manière dont nous affectons les priorités *j* pour les stations du réseau en utilisant les valeurs des compteurs S_i correspondant à chaque type de trafic *AC*[*i*]. Nous débutons par éclaircir quelques principes. Sachant que les priorités des stations doivent être dynamiques et dépendantes à l'activité enregistrée dans le réseau, les nouvelles classes de stations résultent de l'état des compteurs locaux. Nous commutons d'une classe de priorité à une autre si et seulement si les compteurs liés aux différents types de trafic nous conduisent à prendre cette décision. Nous implémentons un algorithme qui effectue des transitions entre les priorités dans les deux sens (vers une priorité plus haute ou plus basse). Ainsi, en cours de transmission dans le réseau, nous éviterons le cas où toutes les stations migrent vers la classe la plus haute ou

la classe la moins prioritaire. Dans ce dernier cas, la priorité inter-station ajoutée ne servirait plus à rien puisque toutes les stations utiliseraient les mêmes valeurs des paramètres d'accès.

La motivation de l'élaboration du nouvel algorithme proposé est qu'une station soit totalement libre de transiter dans les deux sens entre les classes et complètement dynamique en fonction des valeurs des compteurs S_i. Nous interprétons ainsi qu'une station peut changer de priorité en cours de transmission vers une classe plus haute ou plus basse. Ce changement dépend directement de l'historique des transmissions de cette station sur le réseau qui sont traduites par les valeurs des compteurs implémentés. De ce fait, dans la solution proposée, nous avons ajouté un nouveau facteur de prédiction ordonnant l'accès au canal pour un même type de trafic $AC[i]$ sans modifier le mode *CSMA/CA* déjà utilisé par le standard IEEE802.11.

Nous achevons cette section par la définition des règles générales de la nouvelle version révisée de l'*EDCA* :

- Soit k le facteur de décision qui exprime la valeur retournée par un des compteurs S_i. Lorsque cette valeur est atteinte nous effectuons la permutation entre les priorités associée au type de trafic $AC[i]$.
- Si l'un des compteurs des classes de services à priorité haute (notés respectivement par S_0 et S_1) atteint la valeur k, alors la station concernée a admis, dans un proche passé, un trafic multimédia persistant. D'où la pertinence de cette station par rapport aux autres nœuds actifs sur le réseau. Et de ce fait, cette dernière migre à une valeur de priorité plus haute (c.-à-d. de j vers j-1).
- Si l'un des compteurs des classes de services à priorité basse (notés respectivement par S_2 et S_3) atteint la valeur k, alors la station concernée a admis un trafic (*Best Effort* ou *Background*) ne possédant aucune contrainte multimédia. D'où la non importance de ce nœud par rapport aux autres stations. Ainsi, cette station permute de classe de priorité en adoptant un facteur *QoS* plus bas (passe de j vers j+1). Par ce procédé, nous diminuons la charge totale du réseau (ou le taux d'utilisation du canal) causée par des trafics non pertinents.
- Seul le compteur S_i garantissant le comptage des paquets émis sur le réseau appartenant au trafic $AC[i]$, et responsable de l'ultime passage entre deux classes de priorité inter-station (celui qui atteint la valeur k), est ensuite remis à zéro.
- Toutes les stations débutent leurs transmissions sur le canal avec la même priorité qui est la plus basse (par exemple, si nous introduisons 8 classes de stations différentes alors toutes les

stations seront initialisées à la priorité la plus faible j = 7). Donc, elles utiliseront toutes $AIFS[i][7]$ comme valeur du paramètre d'accès lors de l'émission de leurs premières trames.

- Les compteurs S_i situés dans chaque mobile et liés à l'occurrence des paquets déjà transmis pour chaque type de trafic $AC[i]$ sont initialisés à zéro et changeront de valeurs dès les premiers paquets transmis sur le réseau.

Nous constatons que la valeur du facteur k ne doit pas être petite pour que le système ne soit pas prédictif, et soit même perturbé par les changements périodiques des valeurs des paramètres de la couche d'accès (la valeur d'$AIFS[i]$). Elle ne doit pas être trop grande non plus afin que le système puisse suivre l'évolution des transmissions et soit capable de favoriser les trames émises sur le réseau par des stations à caractéristiques plutôt multimédias (qui transmettent souvent des flux conditionnés que ce soit en débit ou en temps). Par la simulation de notre technique sur plusieurs configurations et scénarii possibles, nous allons pouvoir trouver une valeur adéquate du paramètre de décision k. Nous discutons aussi de la valeur du nombre n des priorités inter-station introduites. Nous pouvons même discuter le choix du facteur de décision utilisé pour une remontée et pour une descente en classe de priorité. Nous pouvons par exemple choisir une valeur k pour une permutation vers des priorités plus hautes et $2*k$ en ce qui concerne la décrémentation de priorité puisque les trafics $AC[2]$ et $AC[3]$ sont beaucoup plus présents dans le canal de transmission. Nous illustrons dans la *Figure* 5.6 le fonctionnement ainsi que les détails liés à la prise de décision du nouvel algorithme pour chaque station active dans le réseau.

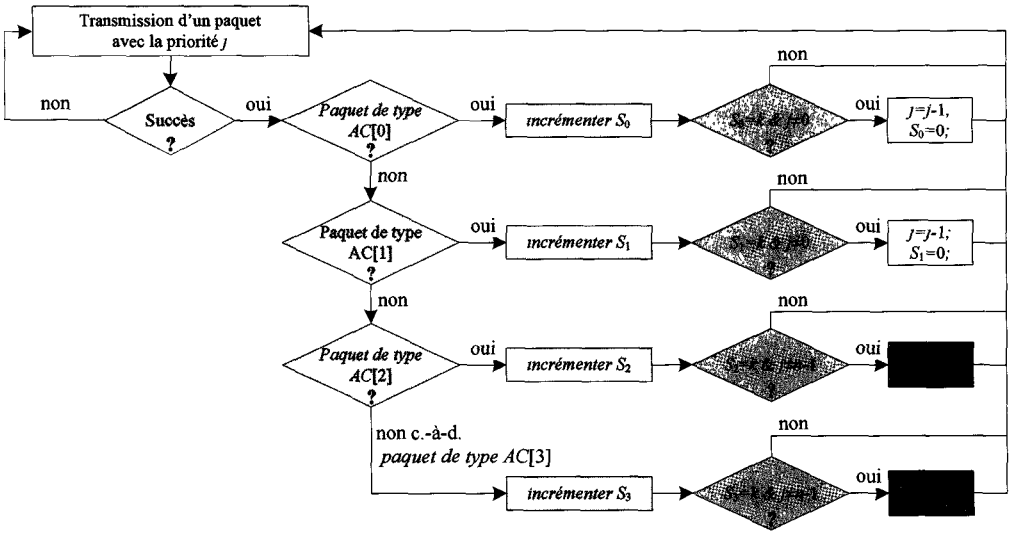

Figure 5.6 : Choix de la priorité inter-station dans l'EDCA révisé

4. Priorité tolérante pour les trafics de classes inférieures

En ce qui concerne le deuxième problème soulevé, nous rappelons qu'il est lié à l'inefficacité du modèle *EDCA* pour les flux appartenant à des classes de priorités faibles en cas d'inactivité des trafics de hautes priorités sur le canal. Ce second obstacle, qui provoque des délais d'attentes considérables en l'absence des trafics de haute priorité, est moins complexe à résoudre que le premier. Nous avons procédé, comme les auteurs de [44], à l'élimination du service à priorité stricte. En effet, les classes de trafic $AC[0]$ et $AC[1]$ produisent un service de priorité stricte ($AC[0]<AC[1]$) tant que les classes de trafic de priorité faible $AC[2]$ et $AC[3]$ produisent un meilleur service dépendant, mais toujours lié à leurs classes de priorité. Lorsque nous désignons le service de priorité stricte, nous indiquons que les valeurs des paramètres responsables de l'accès au médium différent totalement d'un type de trafic $AC[i]$ à un autre. Afin de fournir un service de priorité tolérant entre les trafics de priorité faibles, nous pouvons modifier les valeurs du paramètre $CW[i]$, comme illustré dans le *Tableau* 5.1 suivant, tout en gardant celles correspondant aux $AIFS[i]$ des trafics comme dans l'*Equation* 3.

	$AC[0]$	$AC[1]$	$AC[2]$	$AC[3]$
CW_{min}	3	7	15	15
CW_{max}	7	15	1023	1023

Tableau 5.1 : Valeurs du paramètre $CW[i]$ du modèle EDCA révisé

Nous avons choisi d'ajuster les valeurs du paramètre $CW[i]$ pour minimiser les délais d'attente en l'absence des paquets de priorité haute, et donc de maximiser le taux d'utilisation du canal (charge totale du réseau). Nous ne modifierons aucunes des valeurs des $AIFS[i]$ afin de toujours respecter la classification entres les types de trafic. De plus, notre choix était porté sur les $CW[i]$ et non pas sur les $AIFS[i]$ puisque le deuxième paramètre est un délai à durée fixe et relativement limité par rapport à *BI* (intervalle de *Backoff* qui croît exponentiellement). Donc, nous diminuons fortement l'inefficacité causée par la priorité stricte tout en gardant une différenciation stricte entre les flux à priorité basse. Par ailleurs, nous ne modifions ni le standard, ni l'algorithme du 802.11e déjà déployé. Nous modifions seulement les valeurs utilisées auparavant de $CW[2]$ et $CW[3]$ (min et max), tout en restant conforme au modèle *EDCA* déployé par le standard IEEE802.11e.

IV. Simulations et analyses des résultats

Les protocoles réseaux sont généralement séparés en plusieurs couches. Sous le simulateur de réseaux *NS-2*, chaque couche est conçue indépendamment des autres et les interfaces entre ces couches sont spécifiées de façon très rigoureuse. Les couches sont suffisamment flexibles pour qu'il soit possible de les modifier sans changer l'intégralité du système, ce qui représente en fait le grand avantage d'utiliser ce simulateur.

Nous avons mis en application en premier lieu le modèle *EDCA* classique du 802.11e, en appliquant un *patch* à la sous couche *MAC* du simulateur et nous avons effectué les tests sur ce modèle en exécutant des scripts que nous allons détailler par la suite. Ensuite nous avons implémenté le nouvel algorithme proposé et nous lui avons appliqué les mêmes scripts de simulation afin de le comparer à l'*EDCA* classique.

Le mode adopté est *BSS* (*Basic Service Set*) avec 15 stations mobiles. Ces nœuds utilisent les débits physiques définis dans le standard IEEE 802.11a suivant la qualité du lien (variant entre 6Mb/s et 54Mb/s). Plusieurs trafics *UDP* (*User Datagram Protocol*) - *CBR* (*Constant Bit Rate*) issus de différentes catégories d'accès *AC* sont considérés entre ces nœuds. Les paquets transmis dans le réseau admettent une taille fixe égale à 1000 octets. Le script de simulation est généré pour une durée de $10s$ en utilisant un modèle d'erreur uniforme et en s'étalant sur une surface de $300x300m^2$. Le modèle de propagation choisi est le « *Two Ray Ground* ». L'unité de temps *slot-time* est fixée à $6\mu s$ avec une valeur de *SIFS* égale à $8\mu s$. La valeur minimale du paramètre d'accès *AIFS*[0][0] correspondant au trafic *AC* le plus prioritaire dans le canal est égale à $20\mu s$.

1. Les scénarii adoptés

Les scénarii de simulation sont basés sur le mode infrastructure du réseau WiFi *BSS* (*Basic Service Set*) qui se compose d'une station de base *AP* (*Access Point*) et plusieurs nœuds sans fil. Des trains de paquets bidirectionnels de différents types de trafic (Voix, Vidéo, *Best Effort* et *Background*) sont transmis entre les stations mobiles. La configuration des paquets en transit sur le canal pour cinq stations, choisies aléatoirement parmi celles qui sont actives, est illustrée dans le *Tableau* 5.2. Le nombre total de stations est fixé à 15 afin d'être très proche du cas réel des réseaux WiFi. Le nombre de paquets destinés pour l'analyse des performances est supérieur à 500, ce qui est jugé suffisant pour arbitrer les deux algorithmes simulés. Ultérieurement, nous

adopterons des nombres plus importants de paquets pour mieux montrer la différence en taux de collision entre les deux techniques.

	Station 1	*Station* 2	*Station* 3	*Station* 4	*Station* 5
Voix	20	30	60	10	-
Vidéo	10	20	-	10	40
Best Effort	20	-	30	-	20
Background	40	100	20	90	-

Tableau 5.2 : Trafics adoptés par les stations lors des simulations

Nous choisissons de simuler différents scénarii du trafic des nœuds pour comparer le nouveau procédé avec le modèle standard d'*EDCA* comme illustré dans [47]. Les résultats liés à l'ordonnancement de l'accès des stations au canal de transmission pour les trafics de type *Voix*, *Vidéo* et *Best Effort* sont illustrés dans les *Figures* 5.7(a), 5.7(b) et 5.7(c). Les courbes reportées dans ces figures présentent le choix de la station accédant au canal en fonction du numéro du paquet. Nos résultats sont comparés à ceux qui sont simulés avec le mode standard d'*EDCA*. Les valeurs des paramètres k et n de notre technique sont fixées respectivement à 10 et à 6. Le choix de ces valeurs est discuté dans une deuxième série de tests, et les résultats associés sont présentés dans les *Figures* 5.8(a), 5.8(b), 5.8(c) et 5.8(d).

2. Simulations et résultats obtenus

A. Première évaluation : l'EDCA révisé par rapport l'EDCA classique

Dans les *Figures* 5.7 et 5.8 et pour chaque modèle d'accès simulé, nous schématisons le numéro de la station accédante au canal et responsable du dernier envoi en fonction des paquets à transmettre sur le réseau. Comme montré dans la *Figure* 5.7(a), dès le paquet numéro 30 du trafic le plus prioritaire (*Voix*) le modèle élaboré ne suit plus le même schéma d'accès au canal adopté par les stations mobiles et utilisant l'*EDCA* classique. La nouvelle technique favorise la station N° 1, puis la station N° 2 et enfin la station N° 3 d'une façon ordonnée, et surtout, sans collisions dues à l'utilisation des mêmes valeurs des paramètres d'accès. Le mécanisme d'apprentissage des classes de stations démarre dès les premières transmissions. Ainsi, la classe de priorité actuelle j a changé de valeur à la fin de la transmission de cet ensemble de paquets multimédias pour les nœuds mobiles. A partir des prochaines communications (montrées dans les *Figures* 5.7(b) et 5.7(c)), l'impact de ce mécanisme est plus marquant et nettement discerné,

et la transmission des nouveaux paquets est plus dépendante de la pertinence des stations à caractère multimédia.

L'impact du nouveau mécanisme se fait bien ressentir également sur le deuxième niveau de transmission concernant les paquets de type *Vidéo*. En effet, la station N° 2 prend l'exclusivité lors de l'accès au canal pour la transmission de son trafic puisqu'elle est la plus prioritaire par rapport aux autres nœuds mobiles en l'absence du trafic *Vidéo* pour la station N° 3 qui admet la classe inter-stations la plus haute (voir *Figure* 5.7(b)). En suivant cet ordonnancement, la station qui prend le contrôle du canal par la suite est la station N° 1, puis N° 4 et enfin la station N° 5 qui est le moins prioritaire pour cette configuration instantanée.

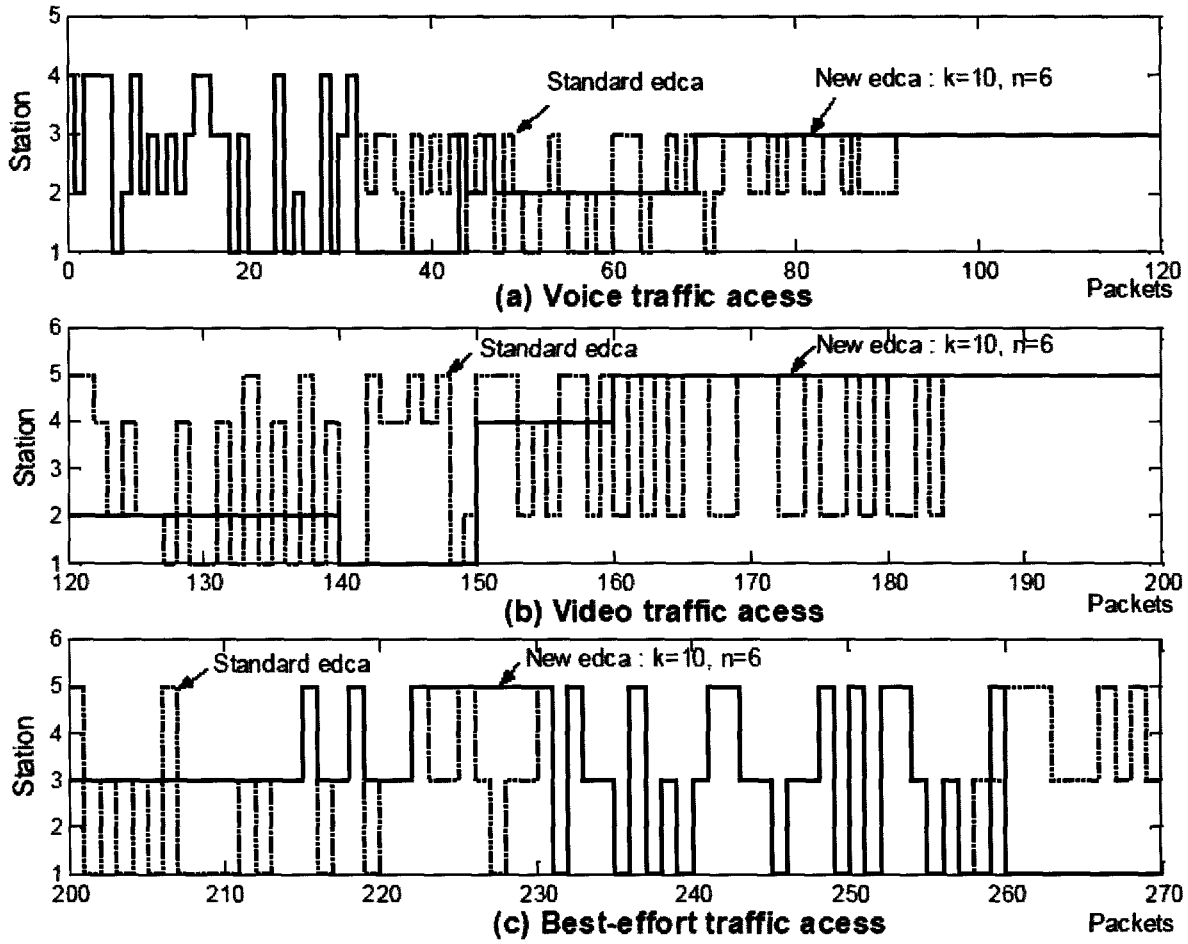

Figure 5.7 : Schémas d'accès au canal par les cinq stations examinées du réseau 802.11e

Simultanément, en l'absence de trafics de haute priorité, la station N° 3 débute la transmission de ses paquets *Best Effort* puisqu'elle est prioritaire actuellement (voir *Figure*

5.7(c)). Après quelques transmissions, la station N° 3 devient de moins en moins prioritaire par rapport aux autres stations et perd par la suite son caractère multimédia. Dès la transmission du paquet numéro 223, la station N° 5 devient plus prioritaire et prend le contrôle du canal. Ainsi de suite, les compteurs des trafics évoluent jusqu'à atteindre la valeur k souhaitée pour ensuite prendre la décision adéquate sur la valeur des priorités inter-stations instantanées.

L'impact du modèle proposé agit directement sur le contrôle d'accès au canal par les stations les plus sollicitées. De ce fait, la concurrence des stations actives pour l'accès au médium du même type de trafic est beaucoup plus nuancée que celle employée par l'*EDCA* classique du standard IEEE 802.11e. En résumé, l'algorithme proposé ne modifie pas le protocole d'accès utilisé et engendre une importante diminution du nombre de collisions entre les flux du même type de trafic, qui sera analysée un peu plus loin.

B. Deuxième évaluation : choix des paramètres k et n de l'EDCA révisé

Dans cette sous section, nous traitons l'influence du nombre des priorités inter-stations n ainsi que le facteur k déployés par le nouveau mécanisme. Pour ce faire, nous avons simulé quatre configurations différentes de scénario. Pour la première configuration nous choisissons la valeur du nombre de classes inter-stations $n = 6$. Dans la seconde formation, nous posons seulement deux classes de priorité $n = 2$. Nous rappelons que dans ces deux scénarii simulés, nous choisissons de fixer la valeur du paramètre de décision $k = 10$. Nous présentons les résultats liés à l'accès par les trafics de type *Vidéo* et *Best Effort*, respectivement, dans les *Figures* 5.8(a) et 5.8(b). Nous observons que la nouvelle classification a introduit un ordonnancement d'accès au médium excellent et optimisé pour les stations qui n'admettent pas le même type de trafic. Également, nous observons que le nombre de priorités déployées dans le réseau WiFi est proportionnément lié au nombre de stations actives. L'avantage principal de la technique proposée est la réduction importante du nombre de collisions dans le réseau. De ce fait, et en partant du principe que les réseaux sans fil WiFi admettent au minimum 10 stations actives simultanément (en s'appuyant sur les études réalisées dans [47, 48, 49]), le choix du nombre de classes inter-station n égal à 6 sera le plus approprié et le mieux adapté aux réseaux 802.11.

Nous illustrons dans les *Figures* 5.8(c) et 5.8(d) la manipulation du facteur de décision k déployé dans le mécanisme proposé et son influence sur les trafics de type *Voix* et *Best Effort*. Nous étudions deux cas de figures en variant le paramètre k. Celle dont la valeur de k est égale à

10 et en seconde configuration *k* est égale à 20 avec un nombre de priorités inter-station *n* fixé à 6 dans les deux cas. Nous constatons, lors de ces deux nouvelles simulations réalisées, que la valeur du facteur de décision *k* étudiée doit être proportionnelle à la charge du réseau en trafic et non dépendante du nombre de nœuds mobiles. De même nous concluons que d'une part, la valeur du facteur *k* ne doit pas être trop petite pour que le système ne soit pas très prédictif, et soit même perturbé par les changements périodiques des valeurs des paramètres de la couche d'accès. D'autre part, cette valeur de *k* ne doit pas être trop grande afin que le système suive l'évolution des transmissions et soit capable de favoriser les trames émises sur le réseau par des stations à caractéristiques plutôt multimédias.

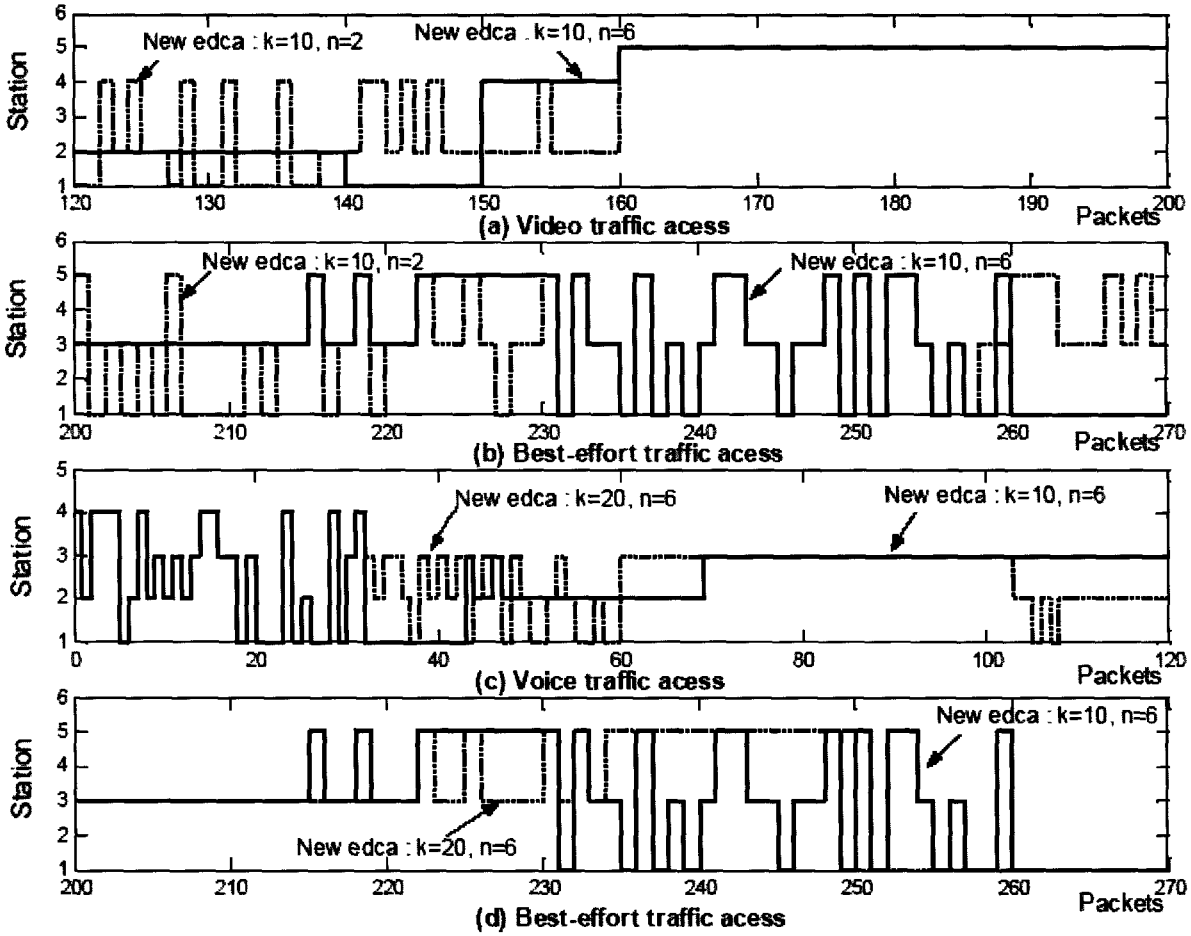

Figure 5.8 : Discussion des paramètres k et n du mécanisme proposé

Ces expérimentations nous ont permis d'explorer les performances du nouveau mécanisme proposé afin d'améliorer le support des applications multimédias dans les réseaux de type IEEE802.11e. Nous pouvons confirmer que cette révision du modèle *EDCA* a modifié le schéma

d'accès pour la transmission des paquets. En effet, à partir des *Figures* 5.7 et 5.8, nous confirmons que l'accès au canal par les stations émettant des trames du même AC n'est plus totalement aléatoire, mais plutôt basée sur un historique proche traduit par une nouvelle priorité entre les nœuds du réseau et qui minimise les collisions d'accès entre ces mobiles.

C. *Troisième évaluation : priorité tolérante aux trafics non-multimédias*

Une nouvelle configuration de simulation est bâtie pour prouver l'efficacité du nouvel algorithme en termes de priorité tolérante pour les trafics de classes de priorité inférieures.

	Station 1	*Station* 2	*Station* 3	*Station* 4	*Station* 5
Voice	-	-	-	-	-
Video	-	-	-	-	-
Best Effort	20	30	30	50	20
Background	40	100	20	90	-

Tableau 5.3 : Trafics adoptés par les stations en l'absence de trafic multimédias

Nous exploitons les résultats d'un scénario de trafic illustré dans le *Tableau* 5.3 (absence de trafics multimédias). Nous n'illustrons pas le schéma d'accès adopté par les stations puisqu'il est similaire à celui effectué par l'*EDCA* classique. En effet, en l'absence de trafic multimédia, les stations commencent avec des priorités initiales minimales ($j = n$-1) et ne changent pas au cours du temps (seuls les compteurs S_2 et S_3 sont incrémentés et ne sont responsables d'aucun changement de priorité). Donc toutes les stations utiliseront les mêmes valeurs des paramètres d'accès (comme celles utilisées par l'*EDCA* classique). Cependant nous notons après la simulation de ce dernier scénario que la charge totale du réseau de transmission est incrémentée. En effet l'utilisation du canal est devenue beaucoup plus efficace que l'*EDCA* classique. Ceci est accompli par l'utilisation de la procédure de priorité tolérante présentée dans la section précédente et adoptée par le nouveau mécanisme (concernant les valeurs de $CW[i]$).

Les résultats de simulation présentés dans le *Tableau* 5.4 montrent le taux moyen d'utilisation du canal en pourcentage. Ce taux est égal au rapport entre le temps alloué à une station donnée afin de transmettre ses paquets et le temps total de transmission. Ce taux reflète donc les pertes de temps réalisées dans le canal de transmission et il est proportionnel aux temps d'attente accomplis par les stations avant de procéder à l'émission effective de leurs paquets.

	EDCA Classique	*EDCA Révisé*
Best Effort	17,6%	25,4%
Background	12,0%	14,9%

Tableau 5.4 : Taux d'utilisation du canal de transmission

Nous remarquons ainsi une meilleure exploitation du canal de transmission en l'absence de trafic des classes supérieures. Ces résultats sont dus au choix de manipuler les valeurs des paramètres $CW[i]$ pour minimiser les délais d'attente en l'absence des paquets de priorité haute. Par le déploiement de ce nouveau procédé nous avons maximisé le taux d'utilisation du canal. Ainsi une meilleure gestion des capacités du réseau est perçue en l'absence de trafic multimédia par la minimisation des temps d'attente excessifs des trafics non prioritaires.

D. Quatrième évaluation : taux de collision enregistrés

Afin de mieux illustrer les performances de notre nouvelle technique d'accès, une dernière série de simulations est jugée nécessaire. Ces nouvelles simulations sont réalisées pour différentes valeurs du nombre de paquets admis à chaque type de trafic. Ces expériences sont menées sur l'*EDCA* classique, et sur l'*EDCA* révisé pour $(k, n) \in \{(10, 2) ; (20, 6) ; (10, 6)\}$. Nous exploitons le fichier trafic de ces simulations. A partir d'une moyenne des données extraites, nous évaluons le nombre des collisions encourues dans le réseau pendant la durée totale de la simulation pour chaque classe de services.

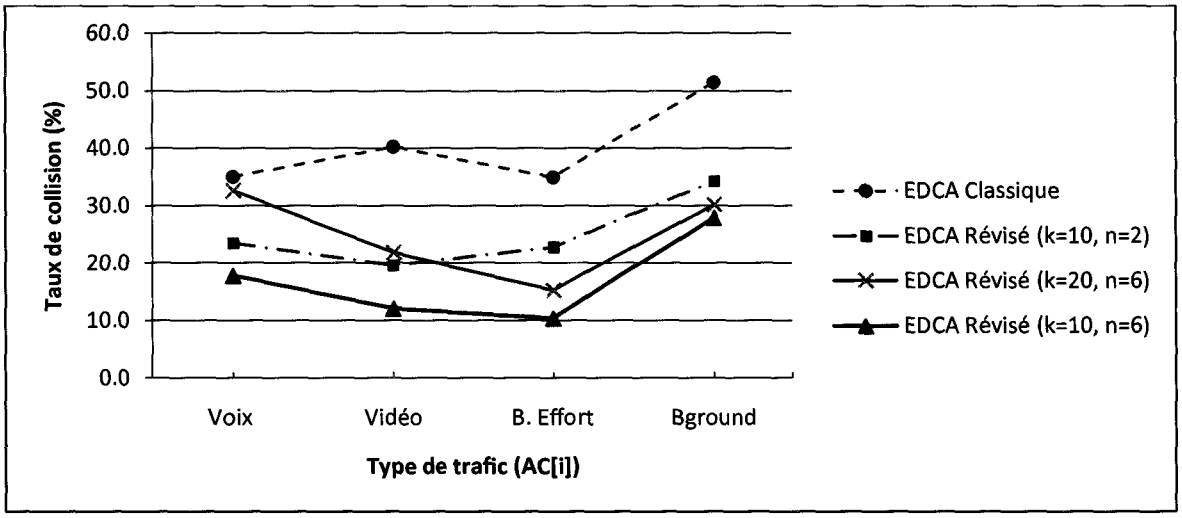

Figure 5.9 : Taux de collision enregistrés pour plusieurs configurations et par type de trafic

Les résultats reportés dans la *Figure* 5.9 confirment que le mécanisme proposé minimise considérablement le nombre de collisions. Le taux de collisions reporté dans cette figure est calculé par le rapport entre les paquets réalisant une collision de la même classe de trafic et le nombre total des paquets émis pour un type de trafic donné. Il faut noter que les résultats obtenus par les trois configurations différentes de l'*EDCA* révisé sont comparés au modèle d'origine déployé avec le standard IEEE802.11e. Les résultats montrent que l'*EDCA* révisé fixant ses paramètres *k* and *n* respectivement à 10 et 6 est beaucoup plus efficace que l'*EDCA* classique. Il est aussi meilleur que les deux autres configurations en termes de taux de collisions.

3. Résumé des simulations réalisées

En conclusion de cette section, nous avons simulé des scénarii faisant intervenir différents type de trafics (*Voix*, *Vidéo*, *Best Effort*, *Background*) en leurs appliquant les deux modèles : l'*EDCA* classique et celui de l'*EDCA* révisé proposé. Nous avons dégagé des apports bénéfiques de cette nouvelle approche à travers la confrontation de ces deux techniques d'accès. Les résultats de simulation prouvent que le mécanisme proposé surpasse le modèle de l'*EDCA* de base en termes de qualité de transmission. L'algorithme proposé améliore la priorité stricte et également le service tolérant. Comparé à d'autres solutions, le nouveau modèle d'*EDCA* est plus facile à mettre en application par rapport à d'autres systèmes étudiés. Il permet aussi une meilleure exploitation globale et plus stable du canal de transmission.

V. Conclusion et synthèse

L'algorithme *EDCA* de la norme 802.11e a évidemment apporté des améliorations pour le schéma de gestion de la *QoS* dans les réseaux WiFi en différenciant les flux de données émis par une seule station. Après avoir étudié ce dernier en profondeur et dégagé ses limites, nous avons jugé utile d'apporter quelques rectifications sur ce mode d'accès en élaborant une nouvelle différenciation de services entre les stations. Ceci a été réalisé en s'appuyant sur un historique récent du trafic adopté par les stations actives sur le réseau. Lors de l'accès au canal, cette nouvelle technique favorise les stations qui émettent régulièrement des trames multimédias par rapport à ceux qui n'en émettent pas ou peu. Par conséquent, nous avons amélioré la méthode d'accès au canal *EDCA* en favorisant les « stations à caractère multimédia », tout en gardant l'ancienne classification entre les types de flux déjà proposée par le 802.11e. Enfin, nous avons

implémenté le nouveau schéma de transmission sous la plateforme *NS*-2, et comparé ce dernier avec celui adopté par l'*EDCA* classique. Nous avons prouvé son apport et confirmé ses avantages pour le support des flux multimédias. La *Figure* 5.10 présente les taux de collision généraux (provenant des quatre classes de services) enregistrés pour la méthode *EDCA* et celle révisée, en faisant varier le nombre de paquets (de classes aléatoires). Nous observons une baisse très importante du taux de collision lorsque le nombre de paquets échangés augmente (diminution de 27.3% des collisions produites lorsque le nombre de paquets atteint 10000).

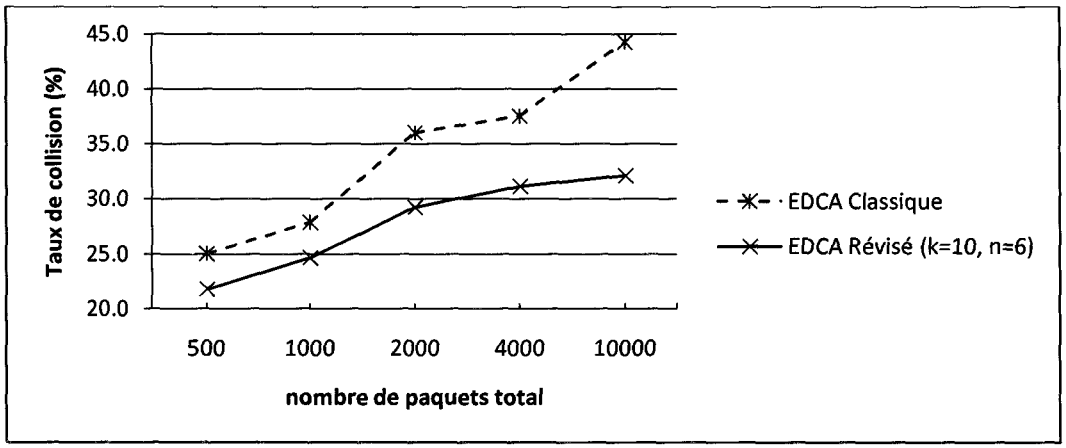

Figure 5.10 : taux de collision général en fonction du nombre de paquets

En résumé, notre approche est conforme à la norme IEEE 802.11e et les simulations réalisées montrent qu'elle améliore la méthode *EDCA* classique, ce qui la rend une bonne candidate pour fournir une meilleure gestion de la *QoS* basée sur l'ajout d'une classification inter-stations. Ces travaux de recherche ont fait l'objet de deux publications dans deux conférences internationales « *The 8th IEEE International Conference on Networks* » (*ICN'09*) [76], et « *The 9th International Conference on ITS Telecommunication* » (*ITST'09*) [79].

CHAPITRE

6 Etude du Handover 802.11

I. Objectifs

Ce chapitre est consacré à un état de l'art sur les mécanismes réalisés pour la réduction de la latence totale du Handover et visant l'optimisation de la transmission du flux multimédia dans les réseaux sans fil IEEE 802.11. Plusieurs travaux ont été effectués récemment sur ce sujet. Nous détaillons ces derniers dans le présent chapitre, et extrayons les apports optimisant les phases du Handover ainsi que les inconvénients relatifs perçus. Enfin, et avant toute illustration, nous jugeons nécessaire d'estimer la latence du mécanisme du Handover 802.11 afin de comprendre les temps passés dans ses différentes phases et pouvoir exploiter ses paramètres actuels.

II. Estimation du temps du Handover

Comme nous l'avons déjà détaillé dans le chapitre 2, le Handover 802.11 est composé d'une phase de désauthentification, d'une phase de recherche d'un nouveau point d'accès, d'une phase d'authentification et d'une phase d'association. Pendant ces différentes phases, un client ne sera pas capable de communiquer avec son *AP* actuel (celui auquel il est affilié). Ce phénomène s'explique par la scrutation de la station mobile des autres canaux pour la recherche de potentiels *APs*. En s'appuyant sur des valeurs définies par la norme, une station pourra rester inaccessible pour toute autre entité du réseau pendant près d'une seconde. Cette valeur est jugée démesurée lors des transmissions, puisque la station concernée pourra perdre plus que 500 paquets qui lui sont destinés.

Le temps de la phase de désauthentication est très faible, puisqu'il comprend uniquement l'échange d'une trame *Désauthentication* entre la station et le point d'accès, suivie immédiatement de son acquittement qui est une trame prioritaire. De plus, tant que cette dernière n'a pas été envoyée nous considérons que le Handover n'est pas encore déclenché, donc le temps qui s'écoule lors de l'accès au médium par la station ne fera pas partie du temps imputé au Handover. Nous pouvons également affirmer que cet échange rajoute peu de charge au réseau sans fil et a peu d'effet sur le trafic de la cellule abandonnée. Aussi il n'y aura pas de coupure de communication puisque la station utilise encore le canal actuel et reçoit les paquets qui lui sont destinés.

La deuxième phase du Handover (la phase de *Scan*) est la plus coûteuse en termes de temps et de trafic. Elle peut être divisée en deux sous-phases : la sous-phase du probe et la sous-phase de commutation de canal. Pendant ces deux sous-phases chaque canal possible (admettant un *AP* fonctionnel) doit être scruté et examiné, ce qui entraine une commutation entre les canaux ainsi que des temps d'attente sur chacun d'eux. La latence de la sous-phase de probe dépend du mode de scan employé (c.-à-d mode passif ou actif précédemment présentés dans le chapitre 2). La latence moyenne de cette sous-phase en adoptant un scan passif peut être représentée en fonction de l'intervalle de temps entre les *beacons* et du nombre de canaux disponibles. Spécifiquement, si l'intervalle entre les *beacons* est de $100ms$ la latence moyenne IEEE 802.11b avec 11 canaux et 802.11a avec 32 canaux est respectivement de $1100ms$ et de $3200ms$. La durée de commutation d'un canal à un autre, comme elle a été définie dans [10], est négligeable et varie entre 40 et $150\mu sec$. D'autre part, le temps encouru avec un scan actif peut être déterminé par les valeurs *MinChannelTime* et de *MaxChannelTime*. Le procédé du scan actif exige qu'un client doit balayer tous les canaux disponibles (11 canaux pour IEEE 802.11b, 32 canaux pour IEEE802.11a). Par conséquent, cette quantité peut être exprimée comme illustré dans *l'Equation* 1.

$$eq.1 \qquad N \times MinChannelTime \leq T_{probe} \leq N \times MaxChannelTime$$

où N est le nombre de canaux disponibles, *MinChannelTime* et *MaxChannelTime* représentent respectivement le temps minimal et maximal pendant lequel un client va rester sur un canal donné et attend les trames de gestions (*probe response*) des éventuels points d'accès.

La méthode la plus intuitive et intéressante pour réduire le temps de la phase de scan est de réduire le nombre de canaux à sonder. Le temps écoulé pendant cette phase peut être largement réduit en balayant seulement des canaux prédéterminés plutôt que tous les canaux disponibles. Une autre alternative pour réduire les valeurs importantes de cette phase est d'ajuster les valeurs de *MinChannelTime* et de *MaxChannelTime* afin de minimiser le temps d'attente sur chaque canal. Les travaux de recherches basés sur ces méthodes seront présentés plus loin.

La phase d'authentification permet une vérification de l'identité d'une station. Selon la sécurité utilisée, ce processus peut être plus ou moins long. Dans un système non sécurisé, seules deux trames *Authentication* sont échangées, avec leurs acquittements respectifs 802.11. En utilisant un système sécurisé, par exemple le cryptage *WEP*, quatre trames doivent être échangées (voir les détails de cette phase dans le chapitre 2). La latence de la phase d'authentification est ainsi proportionnelle au nombre de messages échangés entre le point d'accès (*AP*) et le client. Par conséquent, la durée du système *shared key authentication* est plus longue que celle de l'*open system authentication*. En outre, si un réseau IEEE 802.11 utilise les arrangements d'authentification effectués sur la norme IEEE 802.11i (par exemple IEEE 802.1x et *EAP-tls* [33]), alors le nombre de messages à échanger sera plus grand. Par exemple, les systèmes publics récemment déployés de *WLAN* (par exemple *NeSpot* en Corée [50]) utilisent l'arrangement d'authentification de la norme IEEE 802.11x. Par conséquent, la réduction du temps de la phase d'authentification est susceptible de devenir une issue bien plus intéressante dans de futures versions du 802.11.

La phase d'association ou de réassociation achevant le processus du Handover s'effectue par l'échange de deux trames (*Association Request* et *Association Response*), toutes les deux seront acquittées. Le temps de cette phase, comme celui de la phase d'authentification, se limite d'une part au temps d'accès au médium, qui dépend du trafic dans la cellule (car ces trames de gestion n'ont pas de priorité particulière), et d'autre part au temps d'émission de ces trames.

Dans [51] une étude a été réalisée sur le temps de ces deux dernières phases qui a été évalué à moins de 4*ms* en l'absence de trafic lourd dans la nouvelle cellule sélectionnée. La latence totale du Handover est évaluée par l'expression définie dans l'*Equation* 2 qui suit :

$$eq.2 \qquad T_{Handover} = N \times (T_{commutation} + T_{probe}) + T_{authentification} + T_{association}$$

où N est le nombre de canaux scrutés (ceci dépend du paramètre *ChannelList* de la norme, par défaut N=11 en 802.11b/g), le temps de probe est borné par les paramètres *MinChannelTime* et *MaxChannelTime* (ces deux paramètres peuvent varier d'un modèle de carte réseau à un autre).

En pratique, et en s'appuyant sur l'*Equation* 2, un Handover réalisé par le standard sur un réseau 802.11 pourra avoir théoriquement des valeurs allant de $114ms$ jusqu'à $940ms$ (pour N = 11). Cette valeur est très importante et non acceptable par la plupart des applications à *QoS* exigée (par exemple, les trames de voix qui sont bornées en temps de transmission doivent être reçues chaque $50ms$). Nous concluons cette section par une comparaison présentée dans le *Tableau* 6.1 récapitulant les différences entre les phases du Handover en termes de temps achevés, de charge du trafic et de perte de trames.

	Temps écoulé important	***Charge du trafic affectée***	***Perte de trames produite***
Désauthentification	*Non*	*Non*	*Non*
Scan	*Oui*	*Non*	*Oui*
Authentification	*Suivant la stratégie de sécurité*	*Peu*	*Non*
Association	*Non*	*Non*	*Non*

Tableau 6.1 : Comparaison des phases du Handover

III. Etat de l'art

1. Solutions exploitant les paramètres de la norme

Mishra, Shin et Arbaugh [52] ont démontré que parmi les différentes phases du Handover, celle qui est la plus coûteuse en temps et également en trafic est incontestablement la phase du *Scan* qui consiste en la recherche d'un nouveau point d'accès. Dans leur article [52], ces auteurs proposent une étude détaillée du processus de Handover dans laquelle ils évaluent chaque phase de ce mécanisme. Plusieurs travaux existent dans la littérature sur l'optimisation du Handover qui traitent principalement de l'amélioration de la phase du *Scan*.

ChannelList : Ce paramètre définit la liste des canaux à scruter par une station mobile lorsqu'elle recherche un point d'accès auquel s'affilier, aussi bien en scan actif qu'en scan passif. Il est possible de réduire le nombre de canaux examinés si nous avons connaissance de la configuration de l'infrastructure à laquelle nous essayons de nous connecter. Par exemple, les couvertures cellulaires 802.11b utilisent généralement trois canaux (1, 6 et 11) indépendants qui

n'interfèrent jamais. Nous pouvons par la suite réduire la recherche uniquement à ces 3 canaux au lieu des 13 possibles, ce qui minimise considérablement le temps de la phase du *Scan*.

MinChannelTime et MaxChannelTime : Dans le cas du scan actif, le paramètre *MinChannelTime* définit un temps minimum pendant lequel une station restera connectée sur un canal donné et attendra les trames de gestion (*probe response*) des éventuels points d'accès. Ce temps doit être assez long pour ne pas manquer ces trames et doit respecter la formule donnée dans l'*Equation* 3 suivante :

$$eq.3 \qquad MinChannelTime \geq DIFS + (CW \times SlotTime)$$

avec *DIFS* le temps nécessaire qu'une trame doit attendre pour l'accès au canal après que l'intervalle de *Backoff* (représenté par *CW* x *SlotTime*) commence à décroître. Le paramètre *MinChannelTime* représente le temps maximum d'envoi d'une trame. Une fois ce temps écoulé, le client devra recevoir une réponse du point d'accès. Dans le cas contraire, la station juge soit qu'il n'y a pas de point d'accès sur le canal, soit qu'il y avait d'autres types de trafic en concurrence avec la trame de gestion attendue. Dans ce cas le client attend jusqu'au temps *MaxChannelTime*.

Dans [53] les auteurs ont fixé *MinChannelTime* à $1ms$ et suite à leurs expérimentations, ils ont jugé qu'une valeur de $10ms$ pour *MaxChannelTime* était efficace. Ils ont choisi comme paramètres le nombre de stations dans une cellule, la charge en trafic et le nombre de canaux sur lesquels fonctionnent des points d'accès. Des expérimentations réalisées par simulation du standard 802.11b leur ont permis d'obtenir un temps total de la phase du *Scan*, pour 13 canaux utilisés b et sans aucune charge de trafic, égal à $160ms$. Les auteurs ont aussi démontré dans une seconde étude réalisée avec différents modèles de cartes sans fil 802.11b que le temps de scan peut varier selon les constructeurs (de $87ms$ à $288ms$) puisqu'ils utilisent différentes valeurs de ces paramètres.

2. Solutions pour réduire le temps du Scan

Puisque la phase du *Scan* est la plus coûteuse en temps par rapport au coût total d'un Handover, presque toutes les techniques proposées ont focalisé leurs contributions sur la minimisation du délai de cette phase.

A. Déploiement d'un réseau de capteur

Waharte, Ritzenthaler et Boutaba [54] proposent une solution innovante pour optimiser cette recherche basée sur l'utilisation des capteurs fonctionnant sur le réseau 802.11. Ces capteurs sont disposés dans les cellules WiFi et espacés de 50 à 150 mètres comme le montre la *Figure* 6.1.

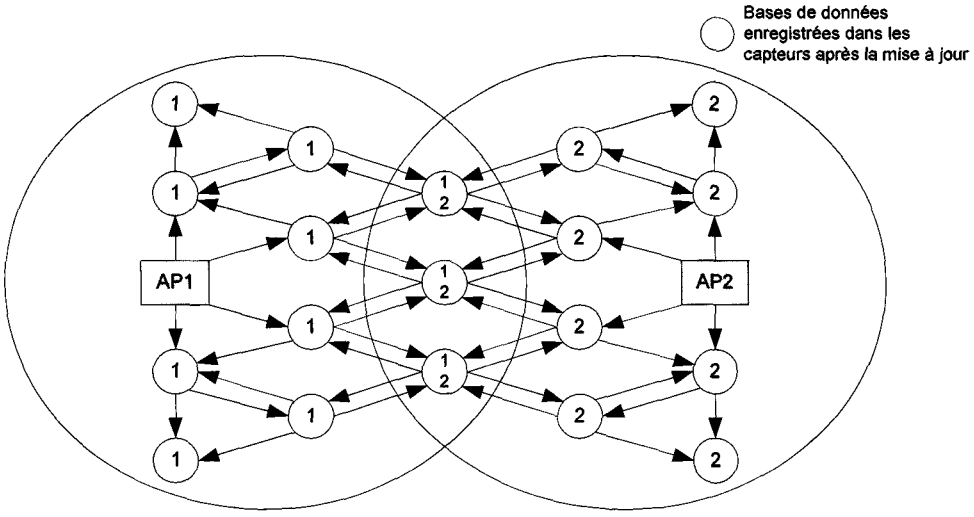

Figure 6.1 : Réseau de capteurs déployé dans le WiFi

Ces capteurs ont pour rôle d'écouter le réseau grâce aux *beacons* envoyés périodiquement par les différents points d'accès. Chaque capteur est capable d'identifier les points d'accès environnants qui lui sont accessibles. Quand un mobile du réseau doit changer de cellule, il procède à une opération de *pré-scan* qui consiste à envoyer une requête d'interrogation aux capteurs. Seuls les capteurs ayant reçu cette requête (à la portée de la station) répondent en envoyant la liste des points d'accès qu'ils ont pu identifier. Chaque capteur répond en utilisant une fenêtre de contention (*CW*) calculée grâce à la puissance du signal de la requête. De cette manière ces trames éviteront les collisions réalisées par la réponse de tous les capteurs en même temps. Aussi, les capteurs les plus proches du mobile, contenant la liste de points d'accès potentiels, répondent les premiers. Les réponses obtenues permettent au mobile d'obtenir des informations liées à ces points d'accès retournés telles que le *BSSID* (adresse *MAC* du point d'accès), le canal utilisé et la puissance du signal. Il effectue ensuite le scan classique du standard 802.11 mais en se limitant uniquement aux canaux de ces points d'accès éventuels. La *Figure* 6.2 illustre cette nouvelle technique de Handover.

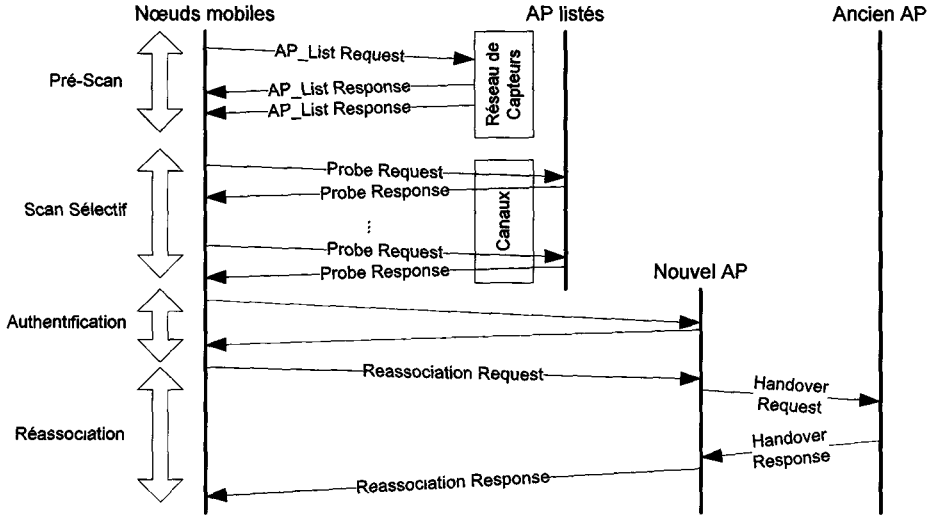

Figure 6.2 : Handover par interrogation de capteurs

Cette solution est efficace en terme de choix du prochain *AP*. Cependant, bien qu'elle ait amélioré considérablement le mécanisme standard, elle est très coûteuse. En effet, elle provoque une charge de trafic inutile du réseau due aux capteurs. De plus, cette solution est non-conforme à la norme IEEE 802.11 et nécessite des modifications radicales du standard.

B. Exécution d'un Scan Synchronisé

Ramani et Savage [55] abordent tous les problèmes liés au mécanisme du Handover en se concentrant sur un problème simple : *comment surveiller sans interruption la proximité des points d'accès voisins ?* Leur solution, appelée « *Synchronized Scan : **SyncScan*** », remplace les grands surcoûts temporels du scan actif par un processus continu qui surveille passivement la présence des points d'accès dans d'autres canaux. La rupture potentielle de la commutation avec le canal actuel est réduite au minimum en synchronisant des périodes courtes d'écoute des autres canaux avec les transmissions périodiques régulières de chaque point d'accès.

En effet, les points d'accès envoient périodiquement des paquets spéciaux, ou balises, pour s'identifier aux clients potentiels et pour synchroniser l'état avec ces derniers. Dans la norme 802.11, les points d'accès sont généralement configurés pour envoyer leurs balises chaque 100*ms*. A chaque période, la norme ne limite pas le temps pour que ces paquets soient générés, ainsi les balises de chaque *AP* sont envoyées sur leur canal respectif à des instants différents des autres balises des *APs* voisins. Les auteurs ont exploité ce degré de liberté pour synchroniser les

clients avec l'émission des balises (*beacons*) de la part des *APs* sur chaque canal. Ils font en sorte que les clients puissent balayer passivement en commutant les canaux quand une balise est sur le point d'être émise sur ces derniers. En d'autres termes, les stations ne quittent leur cellule actuelle pour l'exploration des autres canaux, qu'à des instants très précis pour des périodes très courtes et calculées à l'avance. D'où le terme scan synchronisé (ou *SyncScan*). Un client mobile peut alors utiliser cette propriété pour localiser d'une façon efficace tous les points d'accès dans son voisinage. Par une commutation régulière et ordonnée sur chaque canal, le client réduit alors au minimum le temps de déconnexion avec son propre point d'accès.

La *Figure* 6.3 illustre le fonctionnement de l'algorithme de *SyncScan* ainsi que l'état des points d'accès et du client en termes d'occupation de canal et du signal produit. En effet, le client associé à l'*AP*1 exécute *SyncScan* pour recevoir les balises provenant de l'*AP*6 fonctionnant sur le *canal* 2 et de l'*AP*11 fonctionnant sur le *canal* 3. L'envoi de deux balises est séparé par la période *d*.

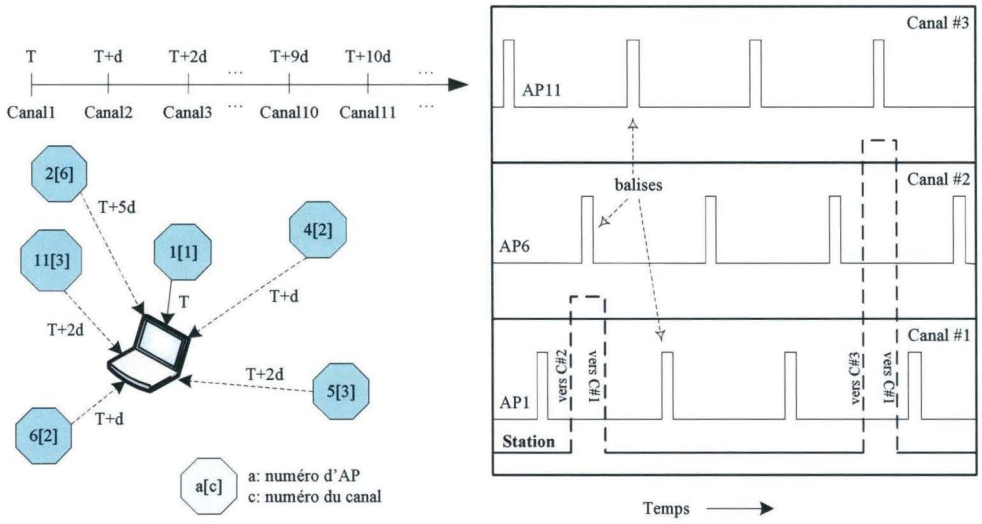

Figure 6.3 : Fonctionnement de l'algorithme SyncScan

Les points d'accès multiples fonctionnant sur le même canal essaient de produire des balises en même temps, donc ils s'interférent potentiellement entre eux. Quand les *APs* ont la même portée de signal, le standard 802.11 utilise l'algorithme d'accès au média ***CCA*** (*Clear Channel Assessment*) pour résoudre le conflit des *beacons* (balises), mais si elles sont encore séparées, les balises peuvent interférer avec un client intermédiaire. Pour éviter ce problème, le temps de génération de balise peut être aléatoirement décalé au-dessus d'une petite fenêtre (par exemple

3*ms*). Un client connecté sur un canal pendant une fenêtre entière devrait recevoir la plupart des balises sur ce canal.

En conclusion, le procédé de *SyncScan* admet un coût caché. Tandis qu'il enlève les surcoûts temporels de la phase du scan, il les remplace par des surcoûts réguliers. Précisément, lorsqu'un client écoute d'autres canaux, il ne peut ni envoyer ni écouter sur son propre *AP*. En conséquence directe, le client peut manquer des paquets qui lui sont envoyés lorsqu'il est en train d'explorer d'autres canaux. Ces erreurs sont très coûteuses en termes de perte de trames et retransmissions réalisées, ce que nous voulons éviter au maximum pour ce type de réseau.

C. Graphe de voisinage

Dans [56], les auteurs proposent un mécanisme rapide de recherche d'un nouvel *AP* dans le cadre des réseaux sans fil 802.11 de l'IEEE qui soutiennent l'*IAPP* [6] afin de diminuer la latence de Handover au dessous de 50*ms* pour pouvoir soutenir la plupart des applications temps réel. Pendant la phase de scan une station sélectionne, à partir d'un graphe de voisinage [57], un nombre limité d'*APs* potentiels et seuls les canaux correspondants seront balayés.

Les auteurs ont amélioré l'approche du graphe de voisinage en mettant la station en mode économie de puissance ***PSM*** (*Power Saving Mode*) avant le balayage des *APs* voisins. La *Figure* 6.4 explique l'architecture de leur système. Les lignes pointillées représentent le lien de voisinage entre les *APs*. Cette solution est basée sur l'utilisation d'un serveur de graphe de voisinage ***NG*** (*Neighboor Graph*) appelé serveur *RADIUS* [58] dans le réseau IEEE 802.11.

Chaque client exécute une fonction appartenant à la couche application (*application-level*), appelée client *NG*, responsable d'échanger les informations concernant le graphe de voisinage avec un serveur *NG*. Ce dernier maintient une table du graphe de voisinage, comme illustré dans le *Tableau* 6.2. Chaque entrée de la table de *NG* contient l'*AP* courant de chaque client, les *APs* voisins de l'*AP* courant, le numéro de canal de ces *APs*, le champ *loading* (facultatif) représente le nombre des stations connectées à l'*AP*, l'adresse *IP* de chaque *AP* et un champ *BSSID*.

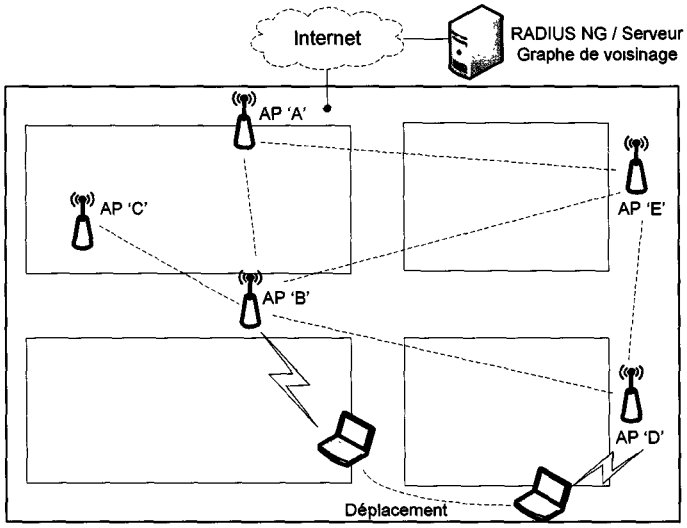

Figure 6.4 : Architecture du graphe de voisinage

AP courant	*Voisin*	*Canal*	*Loading*	*IP*	*BSSID*
AP 'A'	*AP 'B'*	6	2	192.168...	00:60:B3...
	AP 'E'	6	6	192.168...	00:60:B3...
	AP 'D'	11	7	192.168...	00:60:B3...
	...	...	...	...	...
AP 'B'	*AP 'A'*	1	3	192.168...	00:60:B3...
	AP 'C'	11	1	192.168...	00:60:B3...
	AP 'D'	11	7	192.168...	00:60:B3...
	...	...		192.168...	...
...	...	...	...	...	...

Tableau 6.2 : Table du graphe de voisinage

Quand un client effectue un Handover, il reste inaccessible pour toute autre entité du réseau et par conséquent plusieurs trames qui lui sont envoyées peuvent être perdues. Les auteurs proposent par conséquent un mécanisme de pré-repérage pour que l'*IAPP* réduise la latence de Handover. Pour éviter le problème des paquets perdus, ils proposent un arrangement d'expédition et buffering d'armature. Six nouveaux paquets *IAPP* sont conçus à cette fin. La *Figure* 6.5 illustre l'enchaînement de ces messages accompli dans le réseau. La *Figure* 6.6 présente le diagramme d'état du client *NG*. Les étapes exécutées par un client *NG* sont détaillées dans ce qui suit :

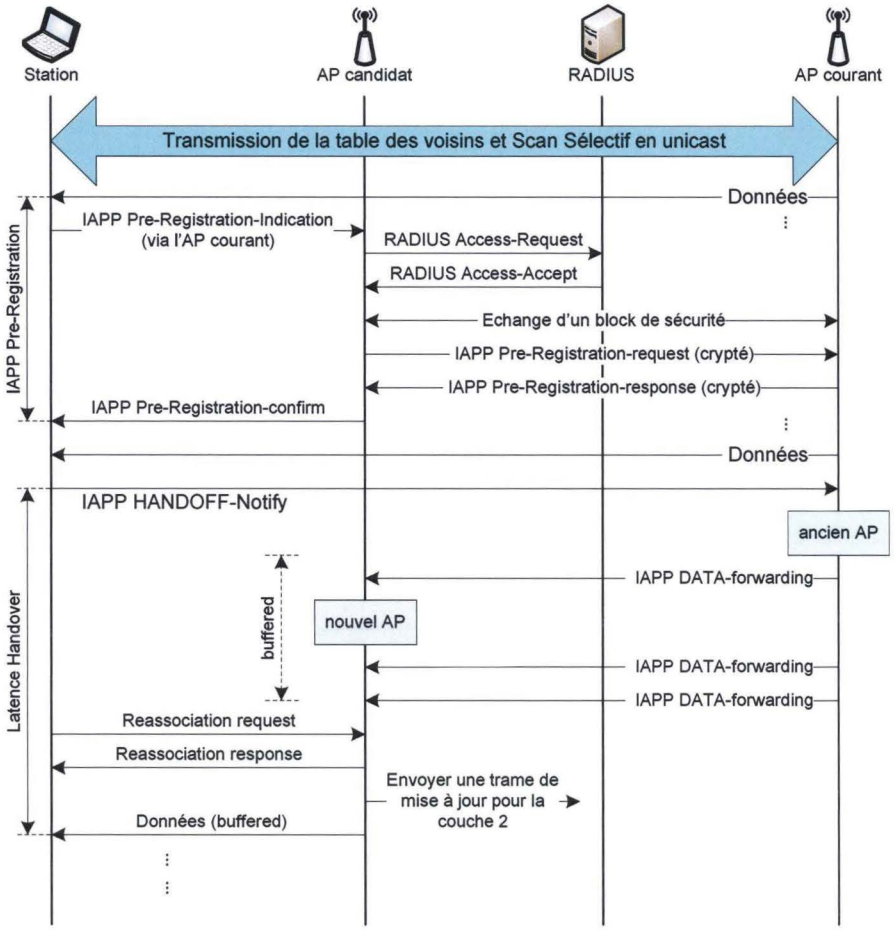

Figure 6.5 : Les paquets échangés par un client NG

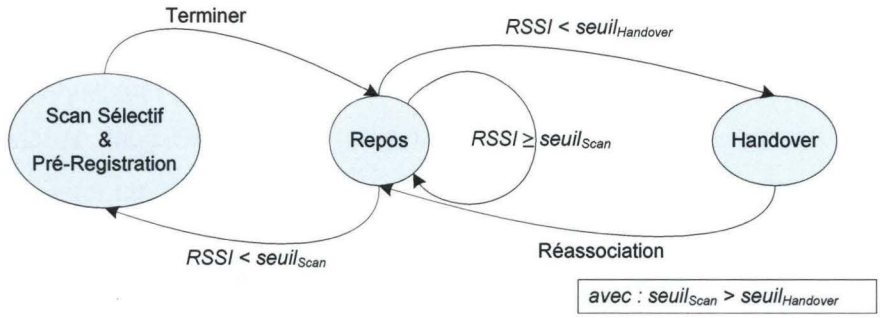

Figure 6.6 : Diagramme d'état du client NG

1) Quand un client *NG* s'associe au réseau, il se relie automatiquement au serveur *NG* pour obtenir la partie de la table *NG* relative à son *AP* courant. Le client aura ainsi la liste de ses voisins qui seront la cible du balayage sélectif dans l'étape 2. Au début, le client est dans l'état repos.

2) Périodiquement, ou après la réception des trames qui lui sont destinées, le client vérifie le *RSSI* (l'indicateur de la force du signal reçu) de son *AP* courant. Quand cet indicateur atteint un seuil appelé le *seuil du scan*, le client *NG* entame l'état de balayage sélectif et de pré-repérage. Dans cet état, le client informe son *AP* courant qu'il entre dans le mode économie de puissance (*PSM*) de sorte que l'*AP* puisse protéger les données qui lui sont destinées. Pendant cette période, le client balaie tous les *APs* voisins uniquement désignés par l'étape 1. L'attente de la réception d'une réponse de chaque *AP* est bornée par *MinChannelTime* (puisque les *APs* visités existent réellement). Si pendant cette période il n'y a ni réponse ni trafic sur le canal, alors le prochain *AP* est sondé. Il s'agit d'un balayage sélectif. Après avoir fini de balayer tous les *APs*, le client informe son *AP* courant qu'il est revenu au mode actif pour recevoir des données.

3) Dès que la station revient en mode actif, elle envoie par l'intermédiaire de son *AP* courant un paquet appelé *IAPP Pre-Registration-indication* à l'*AP* qui admet le meilleur *RSSI* enregistré dans l'étape 2. Au moment de la réception du paquet *IAPP Pre-Registration-indication*, l'*AP* candidat envoie un paquet *RADIUS Access-Request* au serveur *RADIUS* [58] pour vérifier le *BSSID* de l'*AP* courant obtenu à partir du paquet *IAPP Pre-Registration-indication* reçu. Si l'*AP* courant est déclaré, le serveur *RADIUS* répond par un paquet *RADIUS Access-Accept* contenant un bloc de sécurité pour une communication entre l'*AP* courant et l'*AP* candidat. À la réception du paquet *RADIUS Access-Accept*, l'*AP* candidat échange un bloc de sécurité avec l'*AP* courant. Après quoi, l'*AP* candidat transmet un paquet chiffré, *IAPP Pre-Registration-request*, à l'*AP* courant pour demander l'information de contexte du client. L'*AP* courant répond par un paquet *IAPP Pre-Registration-response* qui inclut l'information de contexte du client. Finalement l'*AP* candidat répond par un paquet *IAPP Pre-Registration-confirm*, ce qui termine le procédé de pré-repérage.

4) Quand le *RSSI* de l'*AP* courant, vérifié d'une façon continue, se dégrade au dessous du seuil de Handover, le client *NG* prend une décision de Handover. Le client *NG* informe alors l'*AP* courant de cette décision en lui envoyant le paquet *IAPP HANDOFF-notify*, contenant l'identification du nouvel *AP*. L'*AP* courant casse son association avec le client, et ce dernier peut s'associer directement à un autre *AP*. Avant que l'ancien *AP* ne reçoive une trame de mise à jour de la couche 2 (liaison), toutes les trames transmises au

client seront conduites à l'ancien *AP*. Ces trames sont expédiées au nouvel *AP* par les paquets *IAPP DATA-forwarding*. Lors de la réception de ces trames, le nouvel *AP* protège ces trames et les fournit au client après son affiliation avec lui. Le client *NG* revient alors à l'état repos.

5) Dès que le client est affilié à un nouvel *AP*, le client *NG* se relie automatiquement au serveur *NG* pour obtenir la partie de la table *NG* liée à son nouvel *AP*.

Cette solution a éliminé le problème de perte des paquets lors de la phase du *Scan* par rapport aux autres déjà exposées. Cependant, un coût supplémentaire et non négligeable est à prendre en compte : le nombre considérable de paquets *IAPP* ajoutés qui peut affecter le trafic actuel admis par le réseau. De plus, nous notons que tous les paquets déjà conduits vers l'ancien *AP* sont ensuite routés encore une fois vers le nouvel *AP* choisi avant que la mise à jour de la couche liaison soit réalisée, ce qui correspond à un double acheminement des mêmes trames d'information dans le réseau. Ceci augmente considérablement les chances de collision et de perte dans 802.11 qui seront plus difficiles à gérer que dans les réseaux filaires.

D. Handover rapide en évitant l'attente de probe

Récemment Chintala et Zeng [59] ont proposé deux modifications sur l'algorithme de base du standard IEEE 802.11 qui diminuent de manière significative la latence moyenne de Handover en employant la communication inter-*AP* dans la phase de *Scan*. Les surcoûts temporels encourus pendant la phase de *Scan* sont diminués en forçant les *APs* à envoyer les paquets *probe response* à l'ancien *AP*. La station peut donc éviter d'attendre *MinChannelTime* ou *MaxChannelTime* comme dans l'algorithme classique du standard IEEE 802.11. Aussi, elle évitera la perte des données qui lui sont destinées sans aucun coût supplémentaire dans le réseau correspondant.

Par cette technique, appelée ***FHAP*** (*Fast Handoff by Avoiding ProbeWait*), l'attente du probe est évitée en forçant tous les *APs* voisins fonctionnant sur un canal différent à envoyer leur *probe response* à l'ancien *AP* en utilisant le protocole *IAPP* [6] via le système de distribution reliant les *APs*. La station commute juste les canaux et envoie des paquets *probe request*. Après la phase de probe et suite à l'envoi d'une demande, la station rejoint de nouveau l'ancien *AP* et reçoit tous les *probe response*. Une fois la phase de découverte achevée, la station mobile passera directement

au processus de *Reauthentification* du nouvel *AP*. Le procédé complet de la méthode *FHAP* est décrit dans la *Figure* 6.7.

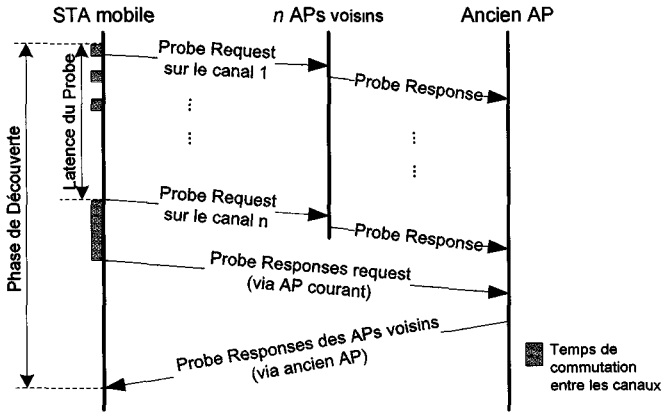

Figure 6.7 : Phase de découverte d'un nouvel AP avec FHAP

Plusieurs aspects importants doivent être pris en considération pour aboutir à un algorithme rapide et efficace comme désiré. Deux inconvénients peuvent être dégagés sur l'approche *FHAP*. Premièrement, le seuil de Handover doit être ajusté de sorte que la station puisse communiquer avec l'ancien *AP* après la phase de *probe*. Ceci implique que le seuil de Handover soit légèrement diminué. Deuxièmement, le problème de non acheminement des paquets *probe response* des éventuels *APs* à l'ancien *AP* doit être abordé.

Afin d'améliorer l'efficacité de la technique *FHAP* effectuée sur l'algorithme de base de la norme IEEE 802.11, les auteurs imposent que la station donne une priorité aux canaux à balayer au cas où la phase de découverte échouerait. La station décale tous les canaux sur lesquels elle a reçu les paquets *probe response* à la fin de la liste prioritaire. Le balayage est effectué sur cette liste seulement si la station ne peut pas trouver un meilleur *AP* sur d'autres canaux. Enfin, il ne faut pas oublier que les paquets *probe response* sont reçus via l'*AP* actuel et non pas sur leurs canaux respectifs. En résumé, les stations ne pourront plus mesurer les valeurs des *RSSI* instantanées et juger la qualité réelle du canal visité (qui n'est possible que si la réception est réalisée sur le canal relatif).

E. Handover préemptif rapide et adaptatif

Chintala et Zeng [59, 60] ont amélioré leur technique *FHAP* en proposant un nouveau dispositif appelé le scan préemptif des *APs*, ***APFH*** (*Adaptive Preemptive Fast Handoff*). La

méthode *APFH* impose que la station prédétermine un nouvel *AP* avant le déclenchement du Handover. Une fois que le seuil de Handover est atteint, la station évite la phase de découverte et déclenche directement la phase de *Reauthentification*, ce qui réduit la totalité de la latence de Handover. Cependant, les auteurs n'ont pas précisé comment la station prédétermine un nouvel *AP*. Le mécanisme de *SyncScan* [55] présente une solution à ce problème. Les ajustements apportés dans *APFH* fournissent un meilleur scan préemptif des *APs*. De plus, toute la phase de découverte est divisée en plusieurs sous-phases et un seul canal est balayé dans chaque sous-phase. La technique APFH subdivise la région de couverture d'un *AP* en trois zones : zone sûre, zone grise et zone de Handover, en fonction de la force du signal reçu (*RSSI*) (voir *Figure* 6.8).

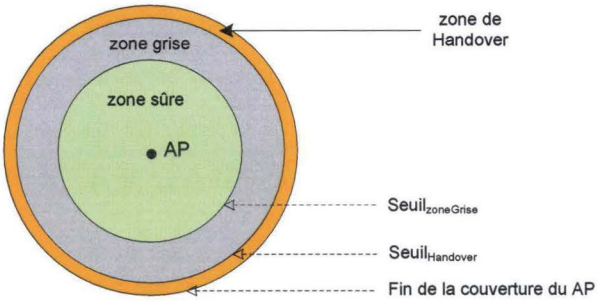

Figure 6.8 : Les zones de couverture d'un AP

Comme son nom l'indique, la zone sûre est la partie du secteur de couverture où la station n'est pas sous la menace d'un Handover. En conséquence la station ne déclenche pas la phase de découverte jusqu'à ce moment et le transfert de données s'effectue normalement. La zone grise est définie comme un secteur où la probabilité de Handover est haute. La station commence à recueillir des informations sur le nouveau meilleur *AP* une fois qu'elle entre dans cette zone grise. La vitesse maximale d'une station mobile choisie pour les simulations est de 15 m/s comme dans [61]. Les *RSSI* absolus et le changement du *RSSI* pendant des intervalles fixes sont employés pour estimer la période de temps pendant laquelle la station reste dans la zone grise.

Le premier mécanisme *FHAP* proposé par ces auteurs a minimisé le délai du probe en recevant toutes les trames *probe response* sur l'ancien *AP*. Néanmoins, cette technique ne respecte pas les contraintes multimédias. Le second mécanisme *APFH* étend toute la latence du Handover et intercale la transmission de données normale avec le balayage de canal. Cette méthode permet de respecter les contraintes des trames *VoIP* à *QoS* stricte.

3. Gestion du Handover Inter-AP

Nous avons vu jusqu'à présent une gestion de la commutation entre les cellules dans un réseau 802.11 au niveau des échanges entre la station et les points d'accès. Nous détaillons davantage dans cette section l'utilité du protocole *IAPP* (*Inter Access Point Protocol*) conçu et standardisé par la norme IEEE 802.11f [6]. Il a été intégré dans plusieurs approches déjà présentées dans la section précédente visant la réduction de la latence du Handover. Avant tout, nous abordons le problème majeur des paquets transmis dans ce type de réseau lors d'un Handover. Quand une station mobile est en train d'effectuer un transfert de données avec une autre station mobile ou une entité de l'infrastructure, les trames sont envoyées à cette station par le point d'accès auquel elle est affiliée. Quand un Handover est effectué, le point d'accès servant de relais va changer et rien n'est prévu dans la norme pour effectuer un suivi du parcours de la station. En conséquence des retards et/ou pertes de trames de données seront enregistrés et peuvent être gênants (par exemple dans le cas de la téléphonie sur WiFi). Nous détaillons dans la suite le *roaming* inter-*AP* de la version 802.11f.

Le protocole *IAPP* (*Inter Access Point Protocol*) a été défini dans une extension du standard IEEE sous la norme 802.11f [6] et révisée en juillet 2003 [62]. C'est un protocole de communication permettant l'interopérabilité entre des points d'accès appartenant au même système de distribution. Il utilise les protocoles ***TCP*** (*Transmission Control Protocol*) et ***UDP*** (*User Datagram Protocol*) sur *IP* pour échanger les paquets du protocole *IAPP* entre les points d'accès. Le protocole *IAPP* admet deux fonctions principales :

- L'association unique d'un client à un réseau ***ESS*** (*Extended Service Set*) : Il vérifie qu'un seul utilisateur est connecté au réseau sous une identité donnée et exploite le serveur *RADIUS* [58] (*Remote Authentification Dial In User Service*) pour gérer l'authentification des stations.
- L'échange proactif du contexte d'une station effectuant un Handover aux points d'accès voisins afin de réduire le temps de latence lors de la nouvelle association.

La *Figure* 6.9 illustre le rôle du serveur *RADIUS* dans le fonctionnement du protocole *IAPP*.

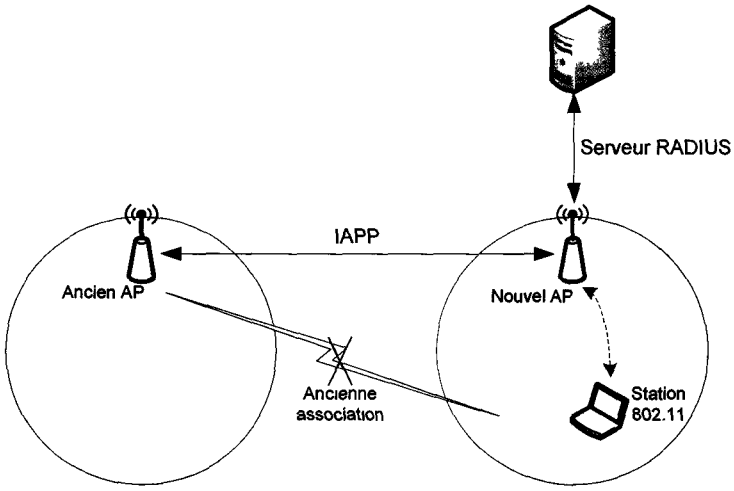

Figure 6.9 : Gestion de l'authentification des stations par le serveur RADIUS

On peut distinguer 3 familles d'évènements :

ADD : Il se produit quand un point d'accès reçoit une trame *Association Request*. Un envoi de trame contenant l'adresse *MAC* de la station est alors effectué sur l'infrastructure. Ceci permettra la mise à jour des tables des éléments réseaux de niveaux 2 (liaison) se trouvant sur l'infrastructure (par exemple les *Switchs* qui mettent à jour leurs tables d'adresses *MAC*). Ce procédé permet de commuter les trames dans la direction du nouveau point d'accès alors qu'en temps normal cette mise à jour n'est effective qu'après l'envoi de la première trame par cette station.

MOVE : Il se produit quand un point d'accès reçoit une trame *Réassociation Request*. Une station va utiliser ce type de trame plutôt que l'*Association Request* quand elle a déjà été affiliée à un autre point d'accès du même *ESS* (*Extended Service Set*). La trame *Réassociation Request* a pour avantage d'inclure un champ "Point d'accès courant". Ce champ identifie l'ancien *AP* qui sera destiné au nouvel *AP* auquel le client est en train de s'affilier. Les deux points d'accès vont alors pouvoir échanger le contexte de cette station.

CACHE : Le principe pour un point d'accès est de transmettre le contexte d'une station aux points d'accès auxquels il est possible qu'elle s'affilie quand elle quittera sa cellule, ces points d'accès sont appelés voisins. Pour cela, grâce aux différents échanges de trames, les points d'accès vont maintenir un graphe qui définit la topologie de l'infrastructure en se référant aux points d'accès voisins qui les entourent.

Le protocole *IAPP*, implémenté dans tous les points d'accès d'une infrastructure, permet d'effectuer un Handover sans rupture de communications (appelé '*Seamless Handover*'). Le but principal de ce protocole est de :

1. Assurer la mise à jour rapide des tables de pont (*bridge tables*) pour ne pas perdre le trafic qui est dirigé vers une station mobile.
2. Partager des données d'authentification concernant la station mobile pour permettre une réauthentification rapide.
3. Introduire des informations sur l'identité des autres *APs* dans le même réseau, pour être ensuite employé par des outils diagnostics.

Comme le diagramme de la *Figure* 6.10 ci-dessous l'illustre, l'échange d'*IAPP* est exécuté entre les *APs* et cela par l'intermédiaire du système de distribution.

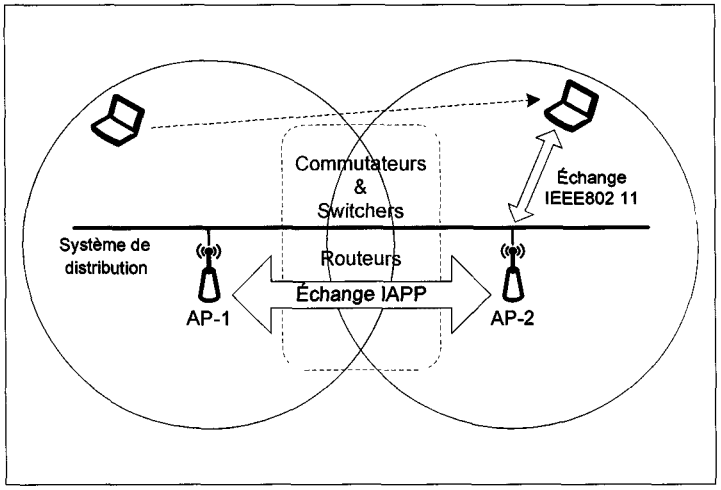

Figure 6.10 : Diagramme d'échange IAPP entre les APs

Quand une station se déplace d'un *AP* à un autre, une réassociation à un "nouvel" *AP* est exécutée pour s'assurer que la station maintient la connexion au réseau. De cette manière le "nouvel" *AP* s'aperçoit de l'arrivée de cette dernière et lui commute les trames qui lui sont destinées. L'ancien *AP* doit être informé de cet événement, de sorte qu'il ne commute plus de trafic destiné à la station en question. L'ancien *AP* peut être informé de deux manières différentes :

- ***Passive*** : La station lance le trafic qui est reçu par l'ancien *AP* sur un port différent (le port d'Ethernet par exemple), où l'ancien *AP* l'a attendu (c.-à-d. le port sans fil). En conséquence l'ancien *AP* mettra à jour ses tables de pont.

- ***Active (par protocole) :*** le nouvel *AP* informera l'ancien que la station est réassociée.

4. Optimisation du Handover au niveau IP

À partir du début des années 90, plusieurs approches ont été proposées pour répondre aux problèmes posés par la mobilité des clients dans les réseaux *IP*. L'approche « officielle » proposée à l'***IETF*** (*Internet Engineering Task Force*) n'a pas réussi à obtenir le consensus nécessaire pour un déploiement à grande échelle. D'autres propositions intervenant à différents endroits de l'infrastructure Internet ont été étudiées, chacune avec ses avantages et ses faiblesses. En conclusion, à l'heure actuelle, aucune solution ne s'est imposée pour répondre de manière satisfaisante à tous les problèmes de la mobilité.

Les extensions du protocole *IP* sont regroupées dans un protocole appelé *Mobile IP*, le même nom porté par le groupe de travail introduisant ce protocole. L'Internet a continué d'évoluer et des nouvelles problématiques et contraintes sont apparues. En réponse, de nouvelles fonctionnalités et améliorations ont été proposées et ajoutées au standard spécifié initialement dans le RFC 2002 [63]. Actuellement, les documents les plus récents qui spécifient les extensions pour la mobilité des clients sont le RFC 3344 [64] pour *IPv4* et le RFC 3775 [65] pour sa version *IPv6*.

Mobile IP est une technique qui intervient exclusivement au niveau *IP* et qui fournit la transparence vis-à-vis des couches supérieures, y compris le protocole *TCP*. Dès le début de la conception du *Mobile IP* (au moins pour sa version v4), la compatibilité avec les clients a été prise en compte. Dans cette section, nous limitons la discussion au protocole *Mobile IPv6* qui représente la solution la plus récente et la plus robuste parmi celles proposées au niveau des couches supérieures (spécifiquement *IP*) pour améliorer la gestion de la mobilité des stations. Nous jugeons inutile la présentation des autres anciennes techniques qui ont très vite été abandonnées.

A. Architecture du Mobile IPv6

Avec le *Mobile IP*, un client mobile est toujours associé à une adresse *IP* de base qui reste inchangée. Celle-ci correspond au sous-réseau d'origine du client mobile. Quand l'hôte se connecte à un sous-réseau différent, il dispose d'une adresse temporaire, propre au nouveau point d'attachement. Il continue cependant d'utiliser son adresse *IP* fixe dans la communication avec

ses correspondants. Dans le schéma d'opération présenté dans la *Figure* 6.11, les paquets destinés à l'hôte mobile sont toujours adressés à son adresse de base. Un nœud spécial dans le sous-réseau d'origine, appelé *agent mère*, intercepte les paquets et les remet à l'emplacement actuel de l'hôte mobile.

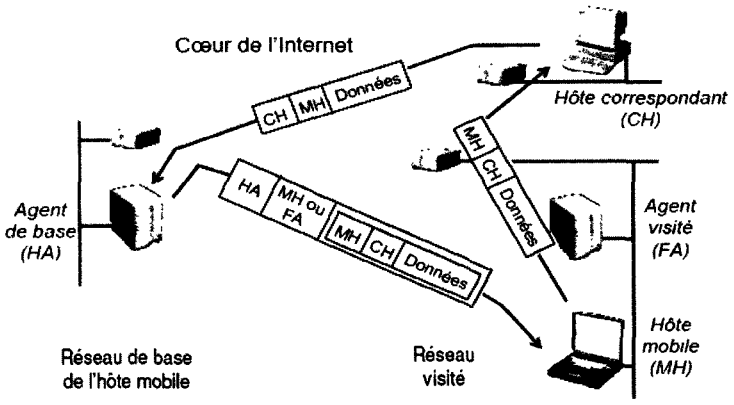

Figure 6.11: Architecture du Mobile IP

B. Protocole Mobile IPv6

Mobile IPv6 [65] admet les mêmes principes de base que son correspondant *IPv4*. Cependant, il comporte un nombre considérable d'améliorations grâce aux fonctionnalités supplémentaires présentes dans *IPv6*. Également, puisqu'il n'y a que peu de systèmes qui utilisent le protocole *IPv6*, le protocole *Mobile IPv6* bénéficie d'un avantage important puisqu'il vise la compatibilité avec les machines existantes. Cet avantage majeur est utile surtout pour optimiser le routage. Rappelons que la difficulté rencontrée dans l'extension similaire de *Mobile IPv4* était justement l'incompatibilité avec les clients correspondants (n'implémentant pas les mécanismes en question). Un autre élément important est que la protection de mises à jour envoyées aux clients correspondants par le client mobile ne demande ni l'établissement d'une association de sécurité auparavant, ni l'existence d'une infrastructure d'authentification. À la place, une méthode appelée *return routability* (un mécanisme qui garantit l'acheminement d'un message de réponse au bon destinataire) est utilisé pour s'assurer que le vrai client mobile envoie le message de mise à jour. Le client mobile peut acquérir son adresse temporaire dans le domaine visité par le mécanisme standard d'auto-configuration du protocole *IPv6* [66]. L'espace d'adressage du protocole *IPv6* est très large et il n'est plus nécessaire qu'un agent visité représente plusieurs clients mobiles par une seule adresse. Pour cela, les agents visités ont été

supprimés de l'architecture de *IPv6*, une différence très importante par rapport à la version *IPv4* où leur présence était préférée.

En revanche, la possibilité que les agents visités coopèrent pour minimiser la perte des paquets lors d'un Handover est éliminée. À la place, *Mobile IPv6* se sert du fait qu'une machine peut avoir plusieurs adresses *IPv6* par interface. Les terminaux peuvent ainsi garder l'ancienne connexion et continuer à recevoir des paquets à cette adresse même après qu'ils soient configurés avec de nouvelles adresses. Les correspondants d'un client mobile envoient les datagrammes en utilisant l'adresse temporaire du client mobile sur le réseau visité. L'adresse fixe est incluse dans le nouvel en-tête de routage *IPv6*. Dans le sens inverse, le client mobile utilise une autre fonctionnalité de *IPv6*, l'option de destination de type *home address*, pour s'identifier. Par ailleurs, l'utilisation des nouveaux mécanismes *IPv6* à la place de l'encapsulation réduit la surcharge observée dans *Mobile IPv4*, tout en permettant aux niveaux supérieurs de ne voir que l'adresse fixe de mobile et donc de continuer à fonctionner de manière transparente.

IV. Conclusion

Nous avons présenté dans ce chapitre un état de l'art détaillé sur les mécanismes modernisant le Handover actuel. Une étude sur l'estimation du temps du Handover a été réalisée dans une première étape. Ensuite, nous avons présenté l'essentiel des travaux d'amélioration de ce dispositif. Nous pouvons conclure que les meilleures techniques sont réalisées au niveau de la sous-couche *MAC* du standard 802.11 qui minimisent le temps de la phase du *Scan* du Handover. En réduisant le temps de cette phase, le nombre de paquets perdus est plus petit et le temps où la station reste non joignable est restreint. L'amélioration des ces deux derniers paramètres clés a comme conséquence directe un respect des contraintes temporelles du trafic multimédia.

CHAPITRE

7 Nouvelle Technique de Scan Préventif PSHP

I. Contexte et objectifs

Le mécanisme de Handover se produit toutes les fois qu'un client doit changer son *AP* d'association. Cependant, le standard IEEE 802.11 n'impose aucun mécanisme pour lancer le Handover à chaque fois que la valeur *RSSI* (*Receive Signal Strength Indicator*) de l'*AP* courant se dégrade au-dessous d'un seuil pré-spécifié (appelé dans la littérature *seuil du Handover*). L'augmentation du seuil du Handover ne résout pas le problème puisqu'une grande valeur conduit la station à exécuter des Handovers fréquents et inutiles. La technique proposée dans ce chapitre fournit une solution efficace en introduisant une nouvelle liste dynamique des meilleurs *RSSIs* des *APs* voisins pour d'éventuels Handovers.

Nous proposons également un mécanisme rapide de recherche d'un nouvel *AP* pour les réseaux sans fil 802.11 de l'IEEE supportant l'*IAPP* (*Inter Access-Point Protocol*), afin de diminuer la latence du Handover au dessous des 50*ms* et de soutenir la plupart des applications temps réel.

II. Prevent Scan Handoff Procedure

La latence typique du Handover dans la norme IEEE 802.11 fonctionnant avec le protocole *IAPP,* varie entre 40 et 300*ms* avec une latence de 40*ms* pour le protocole *IAPP* tout seul [67]. Dans notre méthode, nous imposons que le client s'authentifie dès son association avec le premier *AP* de l'*ESS* (*Extended Service Set*) concerné. Ainsi l'*IAPP* envoie l'information d'authentification à tous les *APs* de l'*ESS* à travers le système de distribution filaire *DS*

(*Distributed System*). Lorsque la réassociation est exigée, le client est alors déjà authentifié avec tous les *APs*. Le protocole *IAPP* basé sur la pré-authentification [68] est réalisé avant même que le client entre dans l'état de découverte. Donc il ne contribue plus à la latence du Handover. Cette première modification est éligible, puisque la norme IEEE 802.11 n'exige ni le fait que l'authentification doive immédiatement précéder l'association, ni qu'elle doive immédiatement suivre un cycle de balayage de canal.

Nous introduisons un nouveau seuil, appelé *seuil de prévention,* en plus du *seuil minimum* (ou *seuil du Handover*) existant dans la norme 802.11. Ce seuil est défini par l'*Equation* 1 suivante :

$$eq.1 \qquad Seuil_{prévention} = RSSI_{min} + (RSSI_{max} - RSSI_{min})/2$$

La valeur du $RSSI_{min}$ = $Seuil_{min}$ est celle déjà définie par la norme et déclenchant le Handover. Nous supposons aussi, que le $RSSI_{max}$ représente la meilleure qualité de lien qui peut être enregistrée entre le client et *AP* d'association (la valeur optimale du signal reçue).

Comme son nom l'indique, le *seuil de prévention* est une valeur évaluant la qualité du lien à partir de laquelle le client est sous la menace d'un Handover. Notre algorithme commence par détecter la mobilité d'un client quand la valeur du *RSSI* reçue de l'*AP* associé se dégrade et atteint ce seuil. Par conséquent la station entame la recherche d'un nouvel *AP* avec une meilleure qualité de lien.

Un mécanisme similaire a été discuté dans [59], appelé *scan préemptif* des *APs*, qui impose que la station mobile doive prédéterminer un nouvel *AP* avant le déclenchement du Handover. Une fois que le seuil du Handover est atteint, la station saute la phase de découverte et déclenche directement la phase de re-authentification. Ce processus réduit toute la latence du Handover. Cependant la manière dont un client prédétermine un nouvel *AP* n'a pas été discutée dans cette référence.

Le mécanisme de *SyncScan* [55] présente une solution à ce problème en supposant que la station scanne périodiquement les canaux et réduit ainsi la latence totale du Handover (voir chapitre précédent). Pour chaque procédure de *SyncScan*, le client doit commuter les canaux, attendre la balise relative à ce canal, puis commuter en arrière vers le canal d'origine. Ainsi pour chaque canal, la latence de *SyncScan* est donnée par l'*Equation 2* suivante :

$$eq.2 \qquad SyncScan_{delai} = (2 * Temps_{commutation}) + Temps_{attente}$$

où $Temps_{commutation}$ est le temps de commutation d'un canal à un autre et $Temps_{attente}$ est le temps nécessaire pour récupérer les balises émises par les *AP*s fonctionnant sur un canal donné. Le nombre de fois où ces surcoûts temporels sont encourus dépend du nombre des canaux balayés.

A titre informationnel, sur la plate-forme de test *Atherosbased NICs* (*chipset* 802.11 sur Lunix avec le pilote *madwifi*), le temps de commutation de canal est de $5ms$, ce qui implique un temps total de commutation de canal de $10ms$ en dehors du temps d'attente. Malgré la durée raisonnable de ces intervalles, ils sont tous plus longs que le temps maximal de retransmission pour les armatures 802.11 ($4ms$ et par conséquent quelques paquets seront perdus). Les points d'accès multiples fonctionnant sur le même canal essayeront de produire des balises en même temps, d'où un risque d'interférences entre eux. Ce procédé admet donc un coût caché : tandis qu'il supprime les surcoûts temporels de la phase de scan, il les remplace par des surcoûts réguliers. Lorsqu'un client écoute d'autres canaux, il ne peut être ni émetteur ni récepteur pour son propre point d'accès. Par conséquent le client peut manquer des paquets qui lui sont envoyés lorsqu'il est en train d'explorer d'autres canaux.

Tseng et Tsai [69] proposent un mécanisme de balayage sélectif pour réduire le temps de recherche d'un nouvel *AP* en combinant l'approche du graphe de voisinage (*NG*) et le protocole *IAPP*. La plupart des opérations liées au Handover sont exécutées avant même qu'il ne se déclenche, y compris le choix du prochain *AP* et le transfert du contexte d'un client. La méthode du balayage sélectif réduit de manière significative la latence totale du mécanisme de Handover. Cependant, elle exige que le client ait une connaissance de la structure du réseau, et qu'il connaisse exactement les *APs* qui lui sont adjacents pour pouvoir employer le balayage sélectif et éviter de balayer tous les canaux. Par conséquent cette méthode impose beaucoup de changements dans l'algorithme de base de la norme IEEE 802.11.

1. Nouvelle procédure d'association

Notre contribution sur l'algorithme de base du Handover permet d'améliorer le scan préemptif des *APs*. En effet notre approche exige d'effectuer le *Scan* (nous l'appelons *pré-scan* dans ce qui suit) avant même le déclenchement réel du Handover. Les informations instantanées concernant les *APs* seront accessibles au client par une liste dynamique triée selon l'ordre décroissant des *RSSIs* des *APs* voisins. Cette liste sera rafraichie à chaque exécution du *pré-scan*.

La liste dynamique est maintenue chez le client mobile et mise à jour périodiquement. Par conséquent, au moment d'un Handover, un client n'a plus besoin d'effectuer un scan complet. L'avantage de cette nouvelle méthode avec le pré-sacan est qu'elle élimine la latence de la phase de *Scan*. Par contre elle sélectionne directement l'*AP* classé en première position dans la liste et effectue une demande d'association avec ce dernier comme illustré dans la *Figure* 7.1.

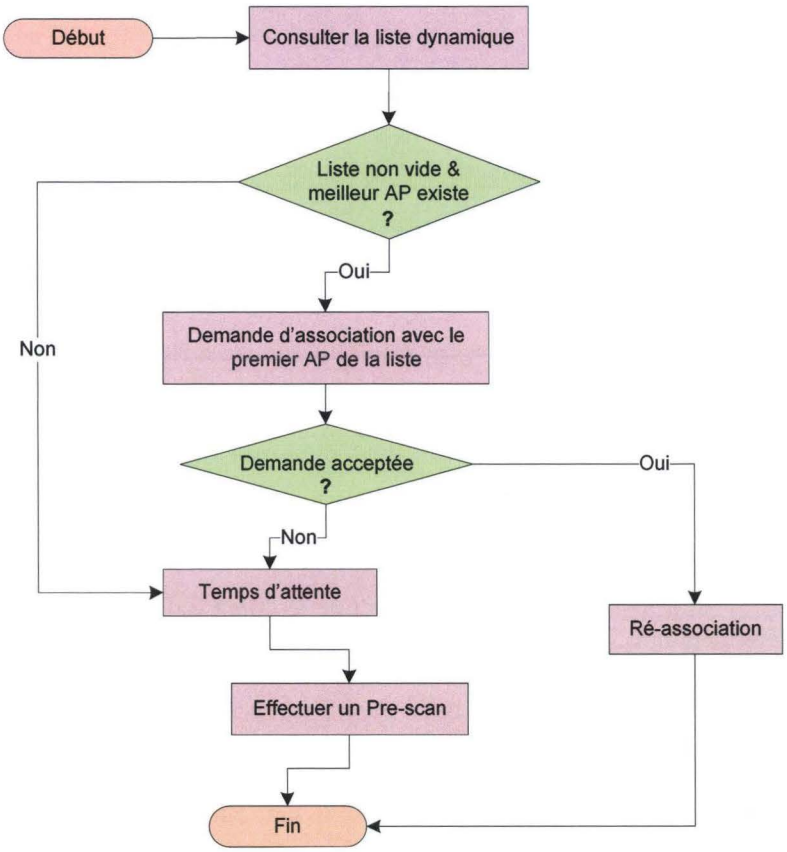

Figure 7.1 : Organigramme d'Association avec le meilleur AP

La demande d'association ne sera acceptée que si et seulement si la valeur du *RSSI* de l'*AP* classé en première position dans la liste dynamique est supérieure au maximum du seuil du Handover et de la valeur du *RSSI* de l'*AP* courant. Autrement, cette demande n'est acceptée que si le premier *AP* de la liste dynamique offre au client une qualité de lien meilleure que celle offerte par son *AP* courant et aussi suffisante pour son fonctionnement sans perdre la connectivité avec les autres entités du réseau. Si l'association avec le premier *AP* a échoué alors la liste sera purgée et le client effectuera de nouveau un *pré-scan*.

2. Nouvelle phase de Pré-Scan

Pendant la phase de *pré-scan*, le client doit commuter des canaux et attendre les *beacons* provenant des *APs* adjacents, ce qui engendre des surcoûts temporels composés des temps de commutation entre les canaux et des temps d'attente sur chacun d'eux. Ainsi pour chaque canal nous pouvons calculer le temps total du pré-scan comme suit :

$$eq.3 \qquad T_{pré-scan} = T_c + T_a$$

où T_c représente le temps nécessaire pour passer d'un canal à un autre et T_a représente la durée d'attente par canal. Les temps de commutation T_c et d'attente T_a sont relativement courts et plus grands que le temps maximum de retransmission pour les trames du standard 802.11 (*4ms*). Par conséquent, quelques paquets seront perdus lors de cette phase de *pré-scan*.

Pour surmonter cet inconvénient, la station est programmée de sorte qu'elle annonce son entrée en mode d'économie de puissance (*PSM*) avant la commutation des canaux. Ceci permet au point d'accès de conserver les paquets destinés au client dans des buffers jusqu'à ce que le client retourne à son canal d'origine et remette à zéro le mode d'économie de puissance. Puisque ces buffers ne seront pas trop remplis lors du mode *PSM* (très court en durée), ils sont rapidement vidés quand le client retourne en mode normal. Lorsque le client balaie un autre canal, il protège tous les paquets hors d'atteinte (*outbound*) pour s'assurer qu'ils ne seront pas perdus. Le *pré-scan* est programmé de façon à ce qu'il ne perturbe pas le trafic existant entre le client et son *AP* courant (ce point sera expliqué ultérieurement). Lors de notre implémentation, nous distinguons trois différentes formes de Handover qui peuvent avoir lieu dans le réseau 802.11. Ces formes dépendent de l'état courant de la liste. Après chaque *pré-scan* effectué le client doit vérifier son *RSSI* courant. Si sa valeur s'est dégradée au dessous du seuil de prévention, alors le client doit effectuer un Handover de forme 1 dans lequel il s'associe au premier *AP* de la liste comme déjà schématisé dans la *Figure* 7.1. Si la valeur du *RSSI* courant s'est dégradée au dessous du seuil du Handover, alors un Handover urgent est déclenché. Le client doit immédiatement décider d'effectuer un Handover de deuxième ou troisième forme. Le Handover de deuxième forme est effectué lorsque la liste dynamique contient au moins un *AP* offrant au client une qualité de lien meilleure que son *AP* courant. Ce Handover est suffisant pour préserver la continuité de connexion du client avec le reste du réseau. Dans le cas où la liste dynamique ne contient aucun *AP* auquel le client peut s'associer, le client procède à un Handover de troisième forme qui

correspond à l'algorithme de scan traditionnel du 802.11. Une fois le client associé à un nouvel *AP* il retourne à l'état initial (appelé l'état repos) et aborde à nouveau la phase de *pré-scan*. La *Figure* 7.2 illustre le fonctionnement du nouveau mécanisme de Handover proposé.

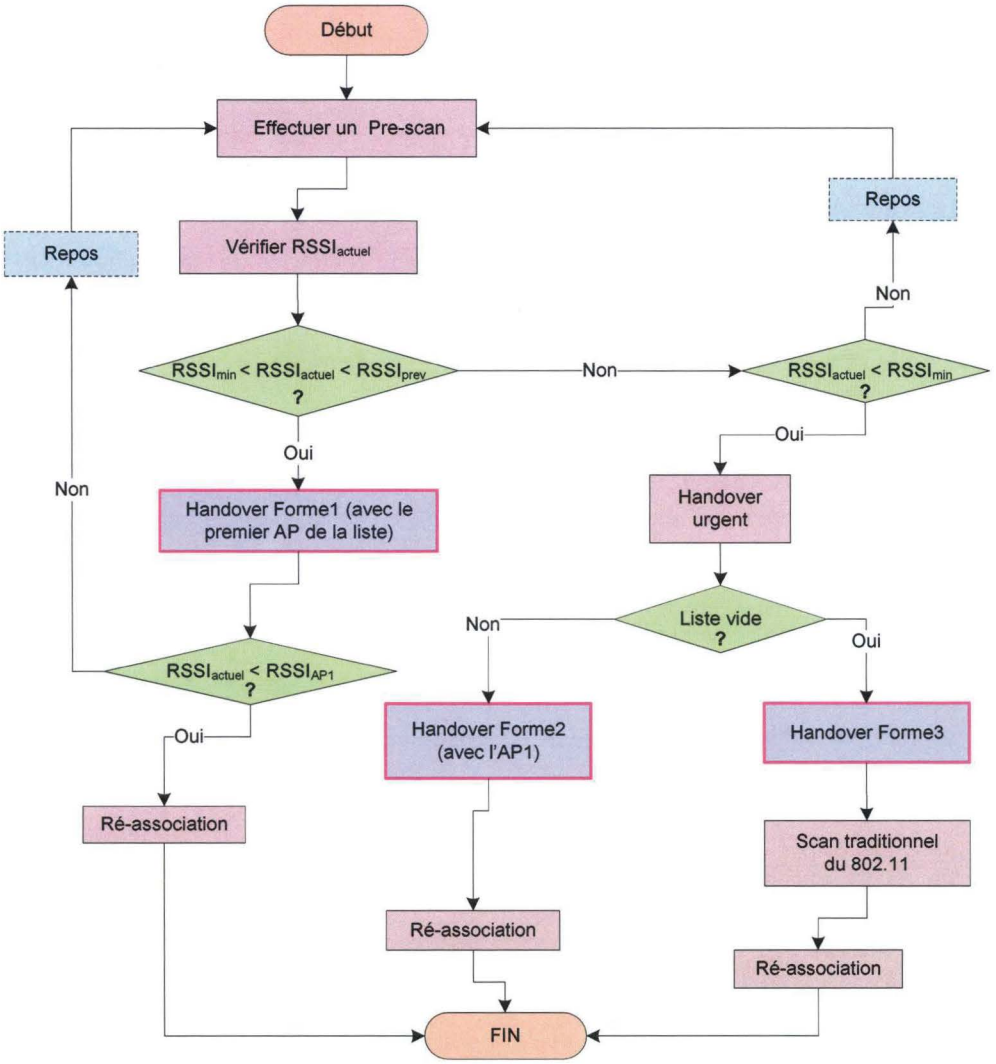

Figure 7.2 : Le nouveau mécanisme PSHP

3. Fonctionnement du nouveau mécanisme PSHP

La *Figure* 7.3 résume les différentes modifications que nous adoptons sur l'algorithme de base, et présente un nouveau diagramme d'état pour un client mobile. La mobilité d'un client induit forcément à des changements de la valeur de *RSSI*. La variation de cette valeur peut aussi être due à différents facteurs, à savoir : les phénomènes naturels, la surcharge de l'*AP*, les interférences, etc.

Dans la suite nous expliquons le fonctionnement de notre nouvelle approche en détaillant les différents états auxquels un client peut faire face, ainsi que les conditions déclenchant le passage d'un état à un autre.

1) Lors de l'association à un *AP* du réseau, le client doit directement procéder à une authentification (Pre-Authentification) avec tous les *APs* existants dans l'*ESS*. Une fois authentifié, il passe alors à l'état *repos*.

2) Suite à la pré-authentification le client passe à la phase *pré-scan*. Dans cet état il informe son *AP* courant qu'il entre en mode économie de puissance (*PSM*) de sorte que l'*AP* puisse protéger les données qui lui sont destinées. Pendant cette période, le client effectue un scan actif *périodique* (chaque α *ms*), qui dépend du type du trafic existant entre le client et son *AP* courant. Nous exigeons que le client ne puisse passer en mode *pré-scan* que si un trafic non prioritaire est transmis sur le canal (trafic de type *Background* ou *Best Effort*). Dans le cas où il s'agit d'un trafic de priorité élevée (flux multimédia) le client doit attendre la fin de ce transit avant d'aborder le *pré-scan*.

   L'avantage majeur de notre contribution est qu'elle favorise la transmission des flux à *QoS* exigée. Lorsqu'un trafic de priorité non significative est adopté entre le client et son *AP* courant, la phase de recherche d'un nouvel *AP* offrant une meilleur qualité de lien pour des *prochaines transmissions* est déclenchée.

   Nous avons défini la valeur de la périodicité α de la phase du *pré-scan* selon l'*Equation* 4 :

$$\alpha = 1.5 * N * (T_{commutation} + MaxChannelTime) \qquad eq.4$$

   où $T_{commutation}$ est le temps nécessaire pour passer d'un canal à un autre, *MaxChannelTime* le temps maximal pendant lequel un client restera sur un canal donné pour attendre les trames *probe response* des éventuels points d'accès, et N est le nombre total de canaux. Le paramètre α est défini de sorte que nous soyons certains qu'avant d'entamer une nouvelle phase de *pré-scan* le client ait déjà terminé le *pré-scan* actuel (mode *PSM*) pour revenir en mode actif et recevoir les trames enregistrées par son *AP* courant en mode *PSM*.

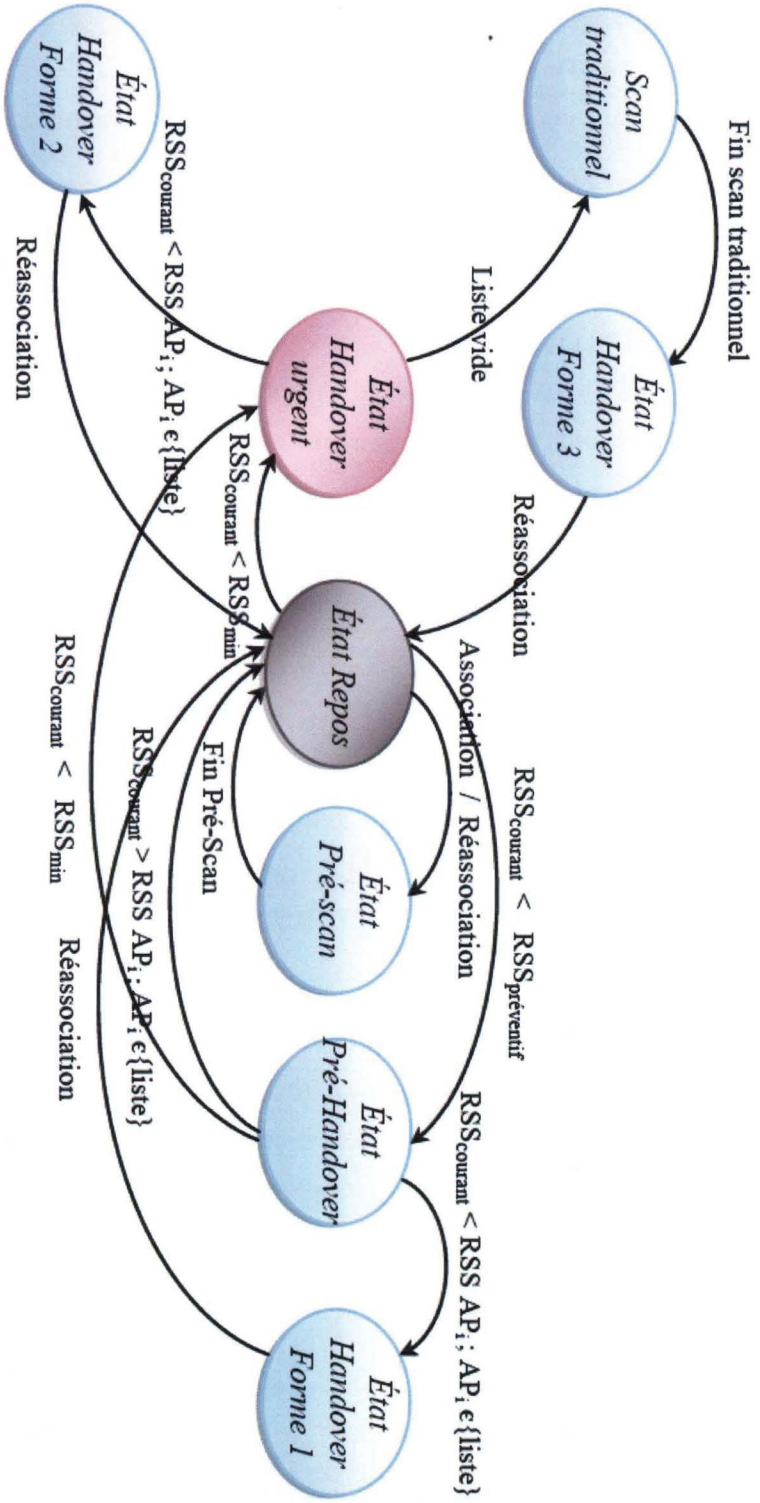

Figure 7.3 : Diagramme d'état pour un client mobile exécutant PSHP

Pendant la période de *pré-scan*, notre algorithme doit collecter des informations concernant chaque *AP* visité (valeur du *RSSI*, adresse *IP* et numéro du canal correspondant). Nous conservons ces détails dans une structure de stockage (une liste dynamique).

Puisque les informations concernant chaque *AP* sont de différents types, et puisque notre algorithme suppose l'existence de plusieurs *APs*. Cette nouvelle structure de stockage est une liste simplement chaînée (les éléments de cette liste sont de type structure représentant un *AP*). Cette liste est vide au départ et doit contenir ensuite au minimum un nœud (*AP*) auquel un client peut s'affilier si la qualité du lien qui le relie avec son *AP* courant s'est dégradée au dessous du seuil de Handover, et au maximum six *APs* puisque la totalité des infrastructures adopte un déploiement hexagonal des cellules *APs*. Le résultat de ce pré-scan est enregistré dans une liste ordonnée suivant l'ordre décroissant des valeurs des *RSSI* des *APs*. Le client passe de l'état pré-scan à l'état repos s'il n'y a plus de canaux à balayer (tous les *APs* voisins ont été visités).

Ce procédé est répété périodiquement (chaque α *ms*) en tenant compte du type du trafic existant sur chaque canal afin de mettre à jour la liste. Ce déploiement répétitif nous permettra de bien suivre l'état du réseau et de garder la liste dynamique liée aux événements qui se produisent au cours du temps. Notons que si cette liste n'est pas mise à jour périodiquement, elle ne donne pas d'avantages sur la décision d'un Handover ni d'utilité dans la phase de *pré-Handover*.

3) Si la valeur de *RSSI* de l'*AP* courant mesurée se dégrade au dessous du *seuil de prévention* le client passe de l'état repos à l'état *pré-Handover* où il cherche dans la liste déjà construite s'il existe un *AP* avec une valeur de *RSSI* supérieure à celle du *RSSI* actuel. Si une tel *AP* existe alors il entame directement l'état *Handover*. Dans le cas échéant le client revient à l'état *repos*. Ce procédé est répété périodiquement tant que $RSSI_{courant} < RSSI_{prévention}$.

4) Nous détaillons ici les différentes formes de Handover qui peuvent être déclenchées avec la nouvelle approche :

   ✓ **Première forme de Handover :**

   Elle est déclenchée suite à un *pré-Handover* avec présence dans la liste d'un *AP* offrant une meilleure qualité. Dans cet état le client doit s'associer directement à l'*AP*

choisi dans la dernière phase. Une fois affilié à ce dernier, le client revient directement à l'état *repos*.

En utilisant cette forme un Handover réel peut être réalisé avec un coût réduit à celui de la phase de réassociation (plutôt qu'une latence presque de 400*ms* réalisée en utilisant la technique standard de la norme 802.11). Cette réduction est suffisante pour soutenir la plupart des applications interactives de voix par exemple.

État Handover urgent : Si le *RSSI* de l'*AP* auquel est affilié un client se dégrade brusquement et atteint le seuil minimum (*seuil Handover*), alors ce dernier passe directement de l'état *repos* à l'état *Handover urgent*. Dans cet état le client doit décider d'effectuer soit un Handover de deuxième forme soit de troisième forme. Le choix dépend uniquement du contenu instantané de la liste dynamique.

✓ **Deuxième forme de Handover :**

Suite à l'état du *Handover urgent*, le client doit s'associer directement à l'*AP* classé en première position dans la liste s'il existe et si son *RSSI* est supérieur au *seuil min* (cet *AP* est appelé *AP* d'urgence). Ce dernier peut permettre d'assurer au moins la force de signal nécessaire pour préserver l'illusion de la continuité requise par les applications interactives puisque le pré-scan est effectué de façon périodique. Après son affiliation avec l'*AP* choisi le client revient à l'état *repos*. En utilisant cette forme, la latence totale du mécanisme de Handover est réduite à un coût un peu supérieur à celui de la réassociation.

✓ **Troisième forme de Handover :**

Au moment d'un *Handover urgent* et dans le cas où la liste est vide, le client procède à un scan selon l'algorithme standard de la norme IEEE 802.11. Il passe donc de l'état *Handover urgent* à l'état de *scan traditionnel*. Dés que ce scan est terminé, le client passe de nouveau à l'état *Handover* de troisième forme pour s'associer au nouvel *AP* trouvé. Après son affiliation avec ce nouvel *AP*, le client revient à l'état *repos*. En utilisant cette forme de Handover, la latence totale augmente et atteint un coût similaire à celui du mécanisme traditionnel de la norme IEEE 802.11. Cependant, cette forme 3 est très rare en pratique (la liste est rarement vide).

5) Une fois que le client est affilié à un nouvel *AP* il revient à l'état *repos* en réinitialisant l'ancienne liste et déclenche ensuite périodiquement l'état de *pré-scan* en l'absence d'un trafic prioritaire pour débuter la construction de la liste associée au nouvel *AP* pour un éventuel Handover.

Un autre avantage du nouvel algorithme est qu'il est autonome puisque chaque client peut juger de l'évolution du réseau et prendre les décisions adéquates en conséquence. Ceci permet la réalisation de Handovers plus rapides et plus appropriés aux conditions instantanées du réseau améliorant la qualité de la connectivité des clients avec leurs *APs* et réduisant, de ce fait, les erreurs et les pertes inutiles des paquets.

En plus de ce dernier avantage, le scan périodique présente également d'autres opportunités pour améliorer la qualité du lien d'un *AP* avec un client. En effet il permet de découvrir d'autres *APs* qui possèdent une meilleure valeur de *RSSI* que celle de l'*AP* actuel, et procéder à des choix plus intelligents avant et/ou pendant un Handover.

Pendant l'exécution du nouvel algorithme un client prélève plusieurs mesures de *RSSI* pendant les phases de pré-scan périodiques. Sa décision sur le changement de la qualité de signal au moment d'un Handover est plus nuancée que toutes les approches étudiées puisqu'elles reposent toutes sur un seul échantillon mesurant le *RSSI* des *AP* sur chaque canal.

III. Implémentations et résultats

Notre nouvelle approche, qui minimise le temps d'exécution du mécanisme de Handover, est implémentée et comparée avec les autres mécanismes usuels. Cependant, l'absence d'une interface réseau qui soutienne ce genre de mécanisme rend impossible l'évaluation de cette solution à travers un test réel. Malheureusement, le modèle de mobilité utilisé par le simulateur *NS-2* permet uniquement de simuler des réseaux locaux sans fil en mode Ad-hoc. De ce fait, et comme c'est le cas pour toutes les autres solutions traitant du même problème, la nécessité d'implémenter un nouveau simulateur permettant l'évaluation de l'approche proposée a été vite ressentie. Dans l'Annexe A de cette thèse, une brève présentation de l'architecture est donnée, ainsi que les traitements du nouveau simulateur implémenté en langage *C++*.

Dans la première sous-section nous décrivons les différents paramètres, ainsi que les scénarii des tests concernant les simulations que nous allons mener. La deuxième sous-section sera consacrée à l'analyse et à la comparaison des résultats de simulations obtenus avec ceux des différents algorithmes étudiés (qui ont été présentés dans le chapitre précédent), ainsi que l'algorithme de base de la norme 802.11. Cette comparaison nous permettra de valider les performances offertes par la nouvelle approche *PSHP* (*Prevent Scan for Handoff Procedure*). Ces expériences sont réalisées sur un nouveau simulateur, baptisé *WHand_Sim* et implémenté dans ce but.

1. Paramétrage de l'algorithme

Afin de pouvoir évaluer et comparer notre nouvelle approche avec l'algorithme de base de la norme IEEE 802.11 ainsi que d'autres méthodes de Handover proposées, nous allons conduire quelques expériences en utilisant le nouveau simulateur. Pour les tests et les simulations que nous présentons dans ce chapitre, nous avons employé un total de 100 *APs*. Les autres paramètres de simulation sont décrits dans le *Tableau* 7.1.

Paramètre	***Valeur***
Vitesse du mobile	0.1 – 15 *m/s*
Modèle de mobilité	*Aléatoire (Random Way Point)*
$Max_{ChannelTime}$	11 *ms*
$Min_{channelTime}$	7 *ms*
$Temps_{Commutation}$	5 *ms*
Nombre de clients	500
Nombre d'APs	100
$Seuil_{Handover}$	-51 *dBm*
$Seuil_{prevention}$	-45 *dBm*
P_0	31.0 *dBm*

Tableau 7.1 : Paramètres de simulation

Lors de la phase de détection, chaque constructeur d'équipements du standard 802.11 utilise son propre algorithme pour calculer la qualité du lien avec le point d'accès courant et décider quand un scan est initié. Cet indicateur de qualité varie principalement en fonction de la puissance du signal observé à la réception de chaque trame envoyée par le point d'accès courant. Dans nos simulations, nous appuyons notre choix par un autre facteur, la distance ou l'éloignement de la station de son point d'accès.

Nous choisissons un modèle de propagation libre. Ainsi le nouveau simulateur utilise l'indicateur de la force du signal reçu basé sur la distance séparant un client de son *AP* d'affiliation (*RSSI-based positionning*) comme démontré dans [70]. La relation de distance entre un client et un *AP* est décrite dans l'*Equation* 5 indiquant le modèle de propagation libre choisi.

$$eq.5 \qquad P_r(d) = P_0 - 20log_{10}\frac{4\pi d}{\lambda}\ [dBm]$$

avec P_0 une constante définie par la norme et λ donné par l'*Equation* 6 suivante :

$$eq.6 \qquad \lambda = \frac{c}{f} = \frac{3\times10^8[m/s]}{2.4[GHz]}$$

où f est la fréquence de transmission et c est la célérité (la vitesse de propagation d'une onde). La *Figure* 7.4 illustre bien la variation de l'indicateur *RSSI* en fonction de la distance. Notre simulateur utilise la formule de la distance euclidienne pour calculer la distance qui sépare un client d'un *AP* donné. La valeur de P_0 est fixée à 31.0 *dBm* selon les auteurs de [70].

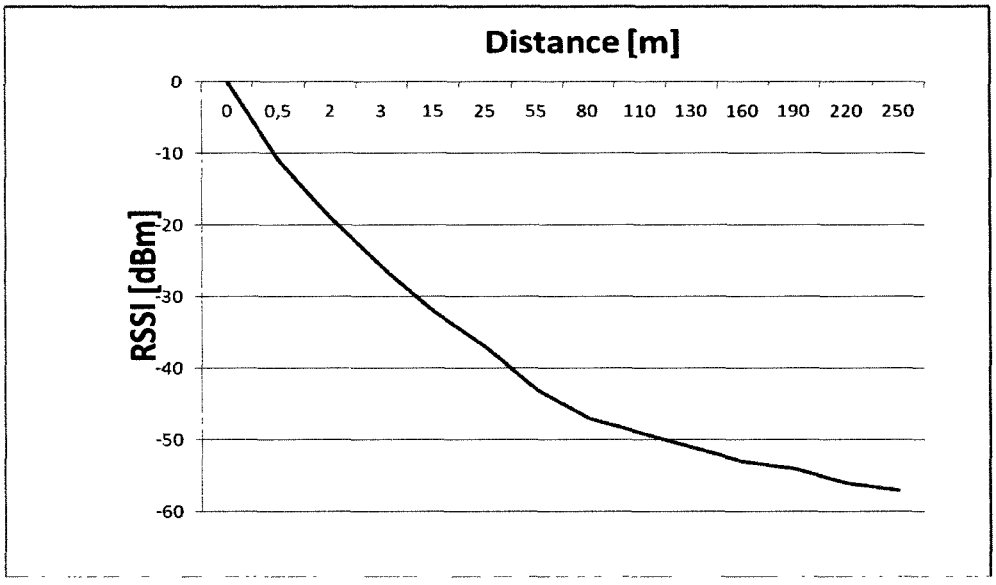

Figure 7.4 : Variation du RSSI d'un client en fonction de la distance

Une fois lancé, notre algorithme va générer 100 *APs* qui ne sont, initialement, en liaison avec aucun client et qui seront automatiquement ajoutés à la liste des *APs*. Ces *APs* seront éparpillés de façon aléatoire sur une étendue de 1 Km^2. Suite à la création des *APs*, l'algorithme génère 500 clients dispersés sur la même étendue et de la même manière que les *APs*.

Au moment de sa création, chaque client identifiera les *APs* qui lui sont voisins, s'authentifiera auprès de ces derniers et calculera ensuite la distance qui le sépare de chacun d'entre eux. En fonction de cette distance, il calculera la valeur de *RSSI* correspondante pour s'associer avec l'*AP* qui lui offre la meilleure qualité de lien.

Après son affiliation avec l'*AP* choisi le client commence à se déplacer aléatoirement dans différents sens et directions grâce à un module chargé de la mobilité des clients, en recalculant à chaque fois la valeur de son *RSSI* courant en fonction de la distance qui le sépare de l'*AP* auquel il est rattaché. Cette mobilité est basée sur le modèle de mobilité aléatoire *« Random Way Point Mobility model »* présenté dans [71]. Ce modèle de déplacement a déjà été adopté dans d'autres algorithmes [55, 56, 57, 59]. Plus d'informations sur ce modèle de mobilité peuvent être trouvées dans [72, 73].

Afin d'être le plus proche possible des conditions réelles du réseau 802.11, et grâce à la programmation multitâches adoptée par le simulateur *WHand_Sim*, le client effectue un pré-scan périodique qui dépend du type de trafic existant entre ce dernier et son *AP* d'affiliation. Le résultat de ce pré-scan est une liste dynamique triée selon l'ordre décroissant des valeurs des *RSSIs* des *APs* voisins et contenant toutes les informations qui peuvent être utiles au moment d'un Handover.

Périodiquement, et parallèlement à la mobilité et au pré-scan, le client vérifie la force du signal qui le relie avec son *AP* courant. Lorsqu'un Handover s'impose (*RSSI* observé inférieur au seuil de prévention), le client accède directement à l'*AP* classé en première position dans la liste dynamique, afin de vérifier si ce dernier peut lui offrir une valeur de signal meilleure et nécessaire pour préserver la connexion avec les différentes entités du réseau. Si ce cas se produit, le client procède à une réassociation puisqu'il est déjà authentifié avec tous les *APs* de son voisinage. Cette commutation de cellule réalisée est un Handover de première forme. Si l'*AP* placé en première position dans cette liste n'offre pas une meilleure qualité de lien que celle de l'*AP* courant le client revient en mode de pré-scan et n'effectue aucune action.

Lorsque la dégradation du *RSSI* d'un client est perçue et atteint une valeur au dessous du seuil du Handover, elle entraine le déclenchement d'un Handover urgent qui peut être sous l'une des deux formes suivantes : un Handover de deuxième forme si la liste dynamique contient au moins un *AP* à qui le client peut s'associer ; ou un Handover de troisième forme dans le cas contraire.

Pour cette dernière forme le client doit procéder à un scan traditionnel comme celui de l'algorithme de base de la norme IEEE 802.11 pour s'associer ensuite à l'*AP* choisi.

2. Résultats et analyses

En général, toutes les solutions proposées visant l'optimisation du mécanisme du Handover ont pour objectif de réduire sa latence totale au-dessous de 50*ms* afin de pouvoir soutenir la plupart des applications multimédias. Nos soucis étaient tout d'abord de proposer une solution qui respecte cette restriction afin de réduire au maximum le trafic et les pertes engendrées suite à l'exécution d'un Handover. En deuxième lieu, nous souhaitions toujours préserver la priorité des trames à *QoS* exigée par rapport au processus enchainé (surtout à la phase de *pré-scan*).

Afin d'évaluer les performances de notre algorithme et le comparer avec d'autres approches, nous jugeons intéressant de comparer tout d'abord leurs réactions lors d'une variation de la charge en trafic. La *Figure* 7.5 illustre une variation de la latence totale du Handover en fonction de la charge du trafic pour l'algorithme de base de la norme IEEE 802.11, la solution la plus récente présentée dans [59] (*Novel MAC Layer Handoff Schemes for IEEE 802.11 Wireless LANs*), et notre nouvelle approche *PSHP* (*Prevent Scan for 802.11 Handoff Procedure*). Ces trois algorithmes sont évalués à travers le nouveau simulateur (*WHand_Sim*) en utilisant les paramètres décrits précédemment. Ainsi, à partir des fichiers traces correspondants à ces trois implémentations, nous pouvons dégager les temps totaux d'exécution pour les différents mécanismes de Handover accomplis lors de cette première simulation.

La contrainte des applications multimédias est représentée par un trait horizontal ($t = 50ms$). Les traits verticaux représentent la latence moyenne des Handovers réalisés. Le délai du Handover moyen de la nouvelle solution est représenté par les traits verticaux en vert pour différentes charges de trafic.

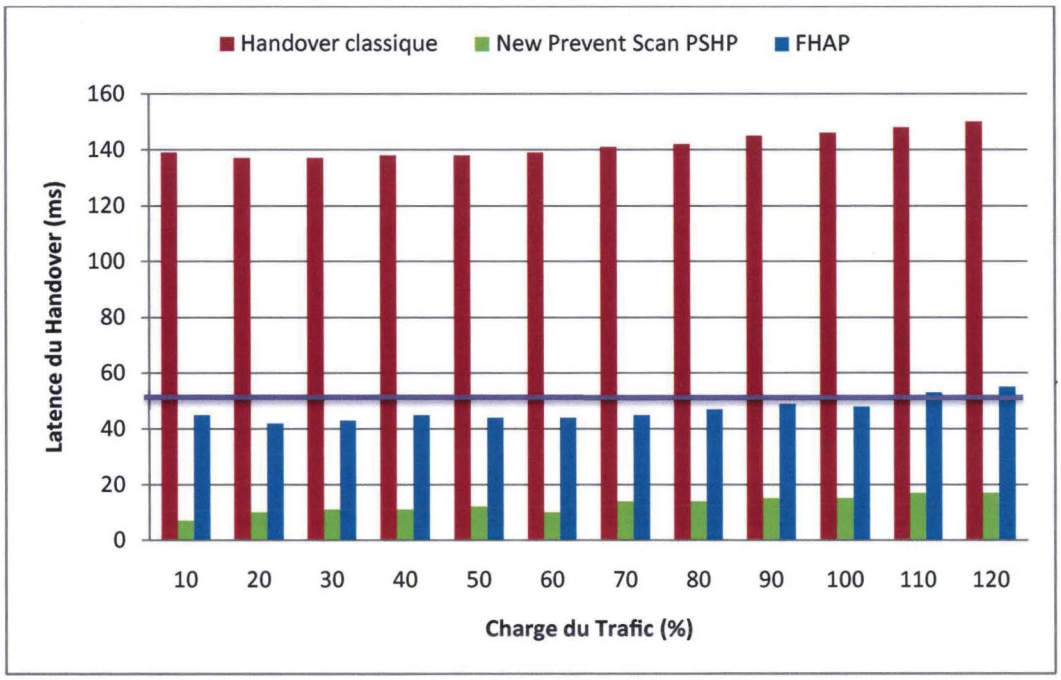

Figure 7.5 : La latence moyenne des Handovers en fonction de la charge du trafic

Dans la *Figure* 7.5 la charge du trafic représente le nombre de clients mobiles par rapport aux *APs* existants dans le scénario du trafic adopté. En effet le nombre maximal de stations mobiles qu'un *AP* peut recevoir simultanément en transmission est de l'ordre de 32 selon plusieurs travaux [55, 56, 59]. Par conséquent, la charge du trafic est représentée par l'*Equation* 7 suivante :

$$eq.7 \qquad Charge_{Trafic} = \frac{nb_{Station}}{nb_{AP} \times 32} \times 100$$

Les résultats obtenus montrent qu'une valeur optimale de la latence moyenne du Handover est obtenue par l'application de la nouvelle approche *PSHP* (*Prevent Scan for 802.11 Handoff Procedure*). La nouvelle valeur du délai respecte aussi bien la contrainte *QoS* recommandée par les applications multimédias que la préservation de la communication d'un client mobile se déplaçant vers un nouvel *AP*. Cette réduction est suffisante pour préserver l'illusion de continuité de la communication entre un client mobile et un nouvel *AP* au moment d'un Handover.

L'algorithme de *Prevent Scan* apporte ainsi une amélioration remarquable par rapport aux algorithmes usuels. En effet la latence moyenne du mécanisme de Handover est considérablement réduite (de l'ordre de 95.21%) par rapport à l'algorithme de base et aussi par

rapport à la solution présentée dans [59] (de 83.97%), ce qui est déjà conforme avec les restrictions imposées par les applications multimédias (latence totale inférieure à 50*ms*). Cette réduction est due principalement au fait que la latence de la phase la plus coûteuse en termes de temps et de trafic, le scan, est réduite à zéro. La latence de la phase d'authentification ne contribue pas non plus à la latence totale du Handover puisque nous utilisons la pré-authentification (le client doit s'authentifier avec tous les *APs* de son voisinage dés sa première association). En conclusion la durée totale du Handover est minimisée jusqu'à un temps similaire à celui de la phase de réassociation. Ce délai, comme signalé dans le chapitre précédent, est négligeable ($\approx 12ms$).

Cette première évaluation des performances de la nouvelle technique a ainsi permis de montrer son efficacité en faible, moyenne et forte charge de trafic. Les résultats obtenus par les simulations ont été bénéfiques et surprenants, ce qui nous encourage fortement à une implémentation réelle de cette technique sur des cartes WiFi. En effet la latence totale du Handover est réduite à un coût négligeable, ce qui permet de mettre en évidence les performances de la phase du *pré-scan* et du déploiement d'une nouvelle liste gérée dynamiquement. L'efficacité ne se limite pas à la réduction du temps d'exécution du Handover, mais également à la simplicité de la solution proposée qui ne nécessite aucun changement incompatible avec la norme de base du Handover contrairement aux autres approches qui imposent fréquemment des modifications sur les étapes de ce mécanisme.

En se basant sur les références [52, 55, 57], dans lesquelles sont apparues les solutions d'optimisation du Handover déjà étudiées, nous avons pu dresser le *Tableau* 7.2 résumant le délai total de ce mécanisme pour différents types de scan.

Technique utilisée	***Latence totale du Handover***
SyncScan	40 ± 5 *ms*
Scan Sélectif	48 ± 5 *ms*
Scan traditionnel du 802.11	entre 112 et 366 *ms*
Prevent Scan PSHP (nouvelle approche)	11 ± 5 *ms*

Tableau 7.2 : La latence moyenne de Handover avec différents types de scan

Ce tableau montre clairement que notre solution réduit considérablement la durée du Handover en restant conforme avec la restriction des applications multimédias (< 50*ms*). Cette réduction est très nette par rapport aux autres solutions, et plus spécifiquement avec l'algorithme de base du mécanisme de Handover. Nous remarquons aussi que la solution utilisant *SyncScan* présente une

réduction importante et permet aussi de respecter la contrainte des applications multimédias. Cependant, le scan sélectif dépasse dans certains cas les limites requises. Ceci est du dans la majorité des cas à la difficulté du graphe de voisinage à gérer les modifications de la topologie du réseau dues à la mobilité continue des clients.

Une troisième comparaison est réalisée entre le nouvel algorithme *PSHP* et la technique la plus récente *FHAP* présentée dans [59]. A travers une nouvelle simulation, nous présentons le temps qui sépare la réception de deux trames consécutives *« delay inter-frame »* lors de l'application de ces deux mécanismes. Les résultats obtenus par l'application des techniques *FHAP* et *PSHP* sont donnés respectivement dans la *Figure* 7.6 et la *Figure* 7.7. La charge du trafic est fixée pour ces simulations à 50% et le nombre de paquets transmis s'élève à 600 ($\approx 1s$). L'accomplissement d'un Handover est marqué par un trait vertical pointillé en vert. Nous notons que la réalisation des Handovers n'est pas simultanée pour les deux schémas simulés. Les stations adoptant le nouvel algorithme *PSHP* (*Figure* 7.6) détectent la détérioration de la qualité du lien avec leurs *APs* correspondants beaucoup plus tôt que celles adoptant le *FHAP* (*Figure* 7.7).

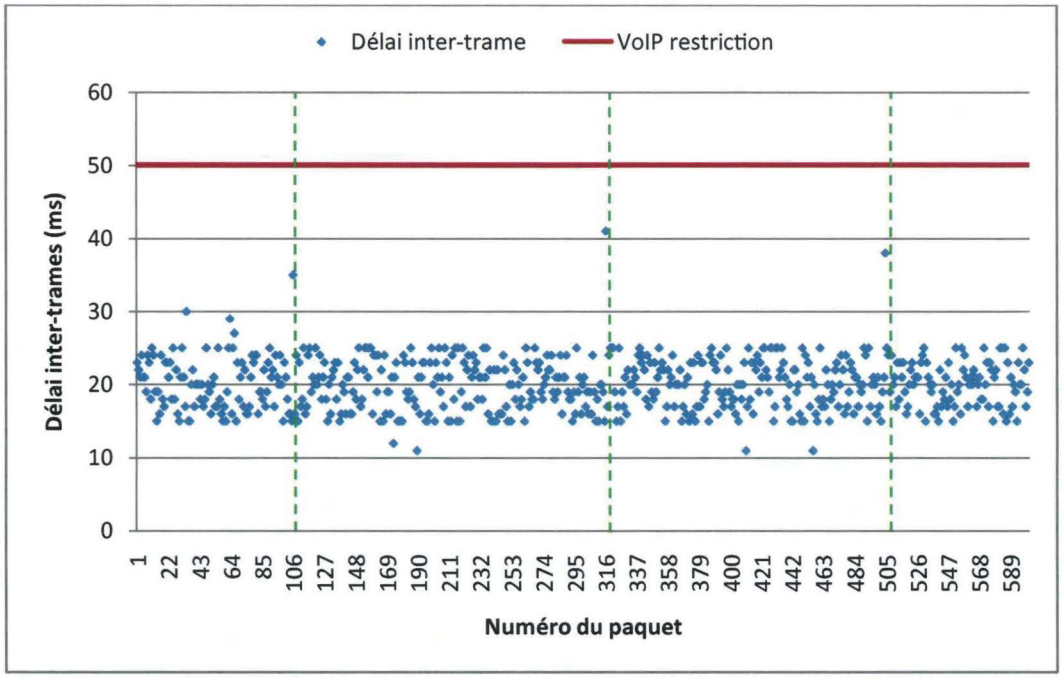

Figure 7.6 : Délais inter-trames obtenus avec le mécanisme FHAP

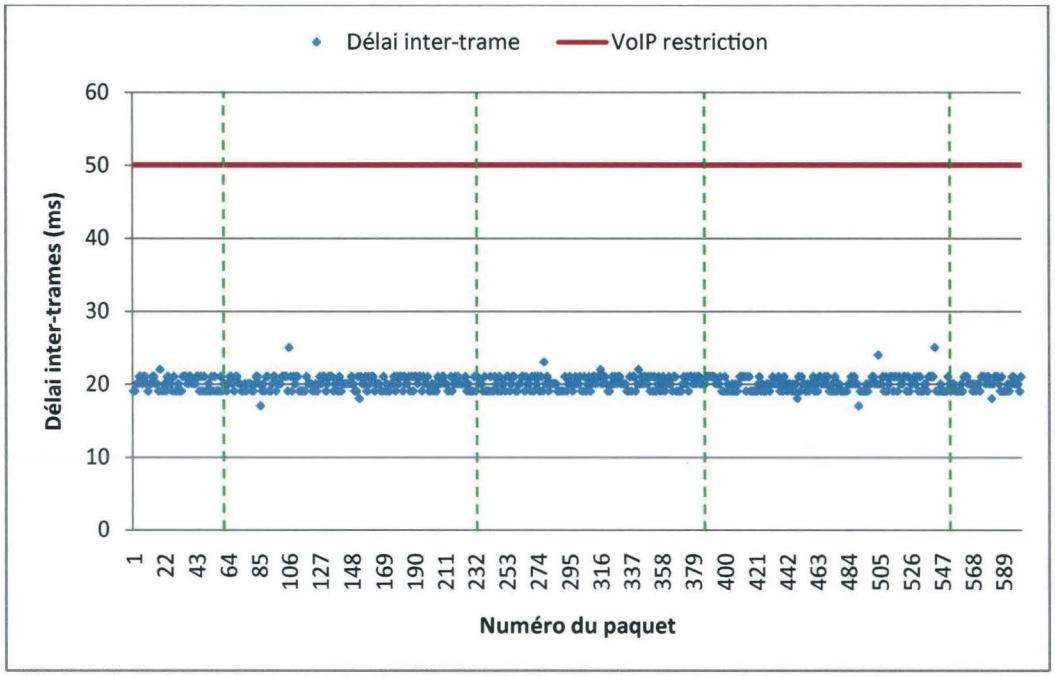

Figure 7.7 : Délais inter-trames par application de la technique PSHP

Nous remarquons que les deux techniques respectent la contrainte temporelle des applications temps réel concernant le délai enregistré entre la réception de deux trames de données successives en n'excédant pas l'intervalle requis ($50ms$). Cependant, cette dernière contrainte est mieux gérée par la nouvelle approche avec des périodes inter-paquets plus régulières et moins importantes en valeurs. Ceci est expliqué par les Handovers de première forme qui sont accomplis dans le réseau et améliorent ainsi la qualité de lien entre les stations et leurs *APs* d'association. Ainsi, nous percevons une amélioration des performances lors de l'application de la nouvelle technique qui est plus préventive et respecte mieux les contraintes des applications à caractères stricts concernant la *QoS* requise.

Le *Tableau* 7.3 reporte les valeurs de la probabilité de perte des paquets *VoIP* émis dans le canal pour les différentes solutions décrites dans ce chapitre. La probabilité de perte retournée est une moyenne calculée à partir de plusieurs échantillons appartenant à différentes valeurs de la charge de trafic adoptée lors des simulations réalisées. Les paquets de voix *VoIP* représentent 75% des trames transmises. Le temps de simulation pour chaque type de trafic est de $10s$ (ce qui équivaut à environ 8000 trames). Nous considérons dans ces simulations que la perte des paquets de voix transmis dans le réseau est avérée seulement si ces trames excèdent $50ms$ comme délai

d'inter-paquet (nous considérons ainsi que les Handovers réalisés sont à l'origine des pertes rencontrées dans le réseau).

Technique utilisée	***Probabilité de perte VoIP***
SyncScan	0.92%
Scan Sélectif	1.28%
FHAP	0.72%
Prevent Scan PSHP (nouvelle approche)	0.53%
Scan traditionnel du 802.11	1.62%

Tableau 7.3 : Probabilité moyenne de perte des paquets VoIP

Nous remarquons clairement l'avantage de la nouvelle approche par rapport aux autres techniques usuelles. Elle minimise en effet le taux d'erreur général de transmission en diminuant considérablement celui causé par des latences inter-trame. La probabilité de perte mesurée des paquets de voix est réduite de deux fois comparée à celles observées chez *SyncScan* et Scan Sélectif, et plus de trois fois par rapport à l'approche traditionnelle du 802.11. Par ailleurs, nous pouvons noter que le nouveau mécanisme est orienté plutôt transmission temps réel multimédia (tel que la voix et la vidéo) puisqu'il est le plus respectueux des contraintes temporelles dans le contexte des réseaux 802.11, comparé aux autres techniques étudiées.

Une analyse plus particulière de la nouvelle approche a aussi été réalisée, et les résultats sont reportés dans la *Figure* 7.8. Elle donne le nombre d'occurrences de Handover en fonction de la charge du trafic. Les traits verticaux en rouge représentent le nombre d'apparitions de Handover de première forme et ceux en bleu représentent le nombre de fois où un Handover de deuxième ou de troisième forme a lieu. Nous rappelons qu'un Handover de première forme n'est déclenché que lorsque la valeur de la qualité du signal qui relie un client à un *AP* se dégrade au dessous du *seuil de prévention* et sans atteindre le *seuil minimal.* Le Handover de deuxième ou de troisième forme n'est déclenché que si cette valeur se dégrade au dessous du *seuil du Handover.*

Nous fixons le temps de simulation à 10*s* pour chaque type de trafic examiné. Les mêmes simulations sont aussi effectuées avec l'algorithme *FHAP* afin de comparer le nombre d'occurrences des Handovers accomplis par les stations. Les résultats sont présentés dans la *Figure* 7.9.

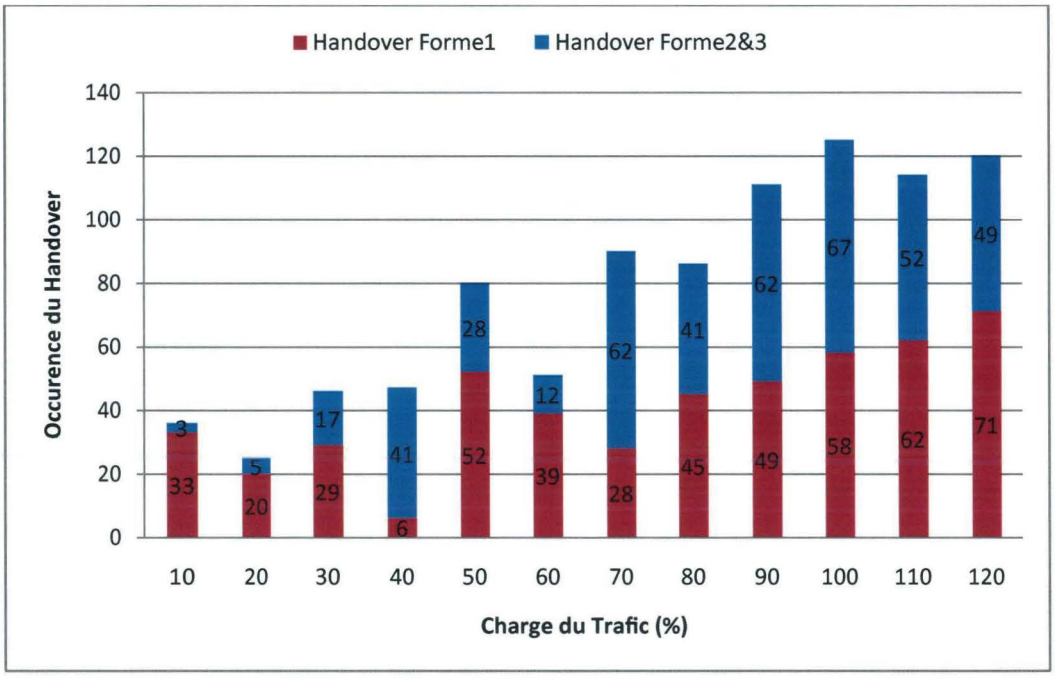

Figure 7.8 : Occurrence des Handovers dans PSHP

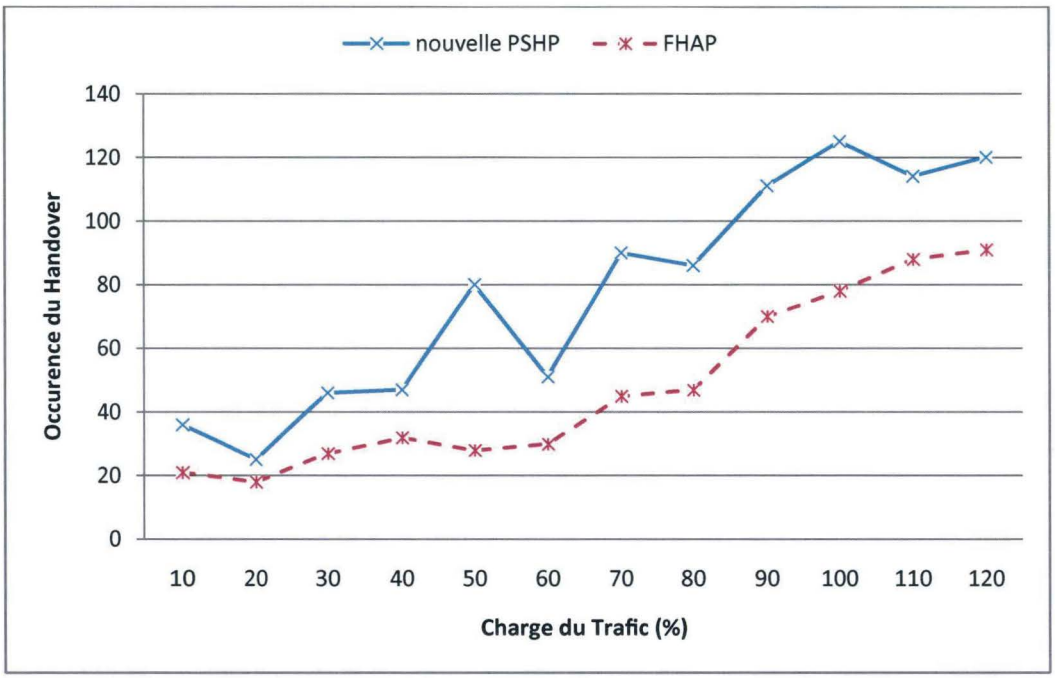

Figure 7.9 : Occurrence des Handovers en fonction de la charge du trafic

En analysant les résultats présentés dans la *Figure* 7.8, nous dégageons qu'en général les Handovers de deuxième et de troisième forme (Handovers urgents) sont moins fréquents que ceux de première forme. Ceci nous permet de confirmer que le nouvel algorithme améliore en général la qualité du signal pour un client mobile puisque les Handovers sont déclenchés avant même que la valeur de *RSSI* se soit dégradée au dessous du *seuil minimal*. En Ainsi nous pouvons noter qu'en moyenne la moitié des Handover réalisés par les stations sont de première forme, ce qui justifie le gain obtenu en appliquant la nouvelle approche en termes de latence et de qualité du lien entre un client mobile et son *AP*.

En comparant les valeurs obtenues par application des deux algorithmes dans la *Figure* 7.9, nous constatons que la technique *FHAP* de [59] effectue moins de Handovers dans le réseau que notre nouvelle technique. Ceci s'explique par l'adoption d'une nouvelle forme préventive de Handover (appelée Forme 1). Grâce à celle-ci, nous n'allons plus attendre que la qualité enregistrée atteigne un *seuil minimal* pour déclencher un Handover. Cette nouvelle technique détecte la détérioration de la qualité du lien avec l'*AP* courant et procède à une commutation de cellule avec un *AP* améliorant les conditions actuelles du lien. En conclusion, le *pré-scan* périodique adopté par la nouvelle technique présente de nouvelles occasions pour l'amélioration de la qualité du lien d'un *AP* avec un client, ainsi qu'une réduction considérable du temps total du mécanisme du Handover. En effet, elle permet de découvrir d'autres *APs* qui possèdent une valeur de *RSSI* meilleure que celle de l'*AP* actuel et de fournir les moyens de faire des choix bien plus intelligents avant et au moment d'un Handover. Ce nouvel algorithme *PSHP* n'influe pas sur le trafic en compétition sur le canal de transmission puisqu'il a été programmé de telle sorte qu'il ne sera exécuté que si et seulement si un trafic de priorité minimale est adopté entre le client et son *AP* d'affiliation. Aussi, en exécutant cet algorithme, un client prélèvera des mesures de *RSSI* périodiques. De ce fait, il peut employer le changement de la qualité de signal comme entrée à sa décision au moment d'un Handover (plutôt que de compter sur un seul échantillon comme dans les approches usuelles de la littérature).

IV. Conclusion

L'approche classique de Handover adoptée dans la norme IEEE 802.11 oblige une station mobile à se dissocier de son ancien point d'accès (*AP*) avant de s'affilier à un nouvel *AP*. Ceci

peut avoir comme conséquence une longue latence du Handover. Le procédé de Handover 802.11 prend jusqu'à plusieurs centaines de millisecondes, et presque 90% du coût de Handover est dû à la recherche d'un nouvel *AP*. Ce facteur est inacceptable dans le cadre des applications de Voix sur *IP* qui nécessitent une latence maximale de 50*ms*. Nous avons proposé une nouvelle solution baptisée *PSHP* (*Prevent Scan for Handoff Procedure*). Elle s'appuie sur une technique permettant de réduire à la fois le temps et le trafic engendré par un Handover. Elle minimise par ailleurs le temps pendant lequel une station reste hors de contact avec son propre point d'accès. Un nouveau simulateur a été conçu afin de pouvoir analyser et comparer les performances de la technique proposée avec différents algorithmes de la littérature y compris l'algorithme de base de la norme. Les résultats de simulation montrent une amélioration significative d'exécution d'un Handover offerte par la nouvelle approche. De plus, notre méthode permet d'obtenir des performances plus intéressantes que les autres techniques élaborées avec le même objectif puisqu'elle respecte la transmission des trames multimédias. Enfin, et à partir des expériences conduites, un taux de 95.7% du nombre total de Handovers réalisés par les stations du réseau sont effectués avec l'*AP* choisi par la *phase de pré-scan* introduite dans la nouvelle solution, offrant ainsi des gains en qualité de lien entre l'*AP* et le client. Ces travaux de recherche ont fait l'objet de trois publications dans trois conférences internationales « *The 2nd IEEE International Conference on Multimedia Computing and Systems* » (*ICMS'09*) [77], « *The 5th International Conference on Networking and Services* » (*ICNS'09*) [78], et « *The annual IEEE International Conference on Ultra Modern Telecommunications* » (*WMCNT-ICUMT'09*) [80].

Conclusion générale

Le travail de recherche conduit dans le cadre de cette Thèse de Doctorat en Informatique, spécialité Automatique et Informatique des Systèmes Industriels et Humains à l'Université de Valenciennes et du Hainaut-Cambrésis, s'inscrit dans le cadre des travaux de recherche portant sur l'amélioration du support des services multimédias dans les réseaux sans fil WiFi. En quête de nouvelles idées non encore explorées par la communauté scientifique, nous avons proposé dans notre travail de recherche, de modifier les techniques standards appliquées pour la gestion de la qualité de service (*QoS*) dans les réseaux sans fil de la norme IEEE 802.11. L'étude de faisabilité de ces approches originales a nécessité d'aborder des aspects théoriques liés à l'adaptation du débit physique, à la différenciation de service, et à l'amélioration du mécanisme de commutation inter-cellules (Handover).

Du point de vue théorique, les travaux de recherche menés ont concerné dans un premier temps l'approfondissement de l'étude de l'apport des réseaux sans fil, et plus spécifiquement les caractéristiques relatives au fonctionnement du standard IEEE 802.11. L'objectif de cette étude était de définir les champs de recherche courants dans cette norme et de démontrer les éventuelles difficultés rencontrées lors du déploiement de ce type de *WLAN* pour effectuer des transmissions multimédias (appelées aussi à *QoS* exigée).

Du point de vue pratique, nous avons proposé une conception originale de gestion de la qualité de service en adoptant un nouveau schéma global de support des trafics multimédias. L'objectif à ce niveau était d'apporter une amélioration des performances de telles émissions à *QoS* exigée dans le réseau WiFi par rapport aux travaux antérieurs qui n'ont pas réussi à adopter un meilleur débit disponible selon les conditions instantanées du canal, ni à respecter les contraintes temporelles de livraison (le cas de la Voix et de la Vidéo).

Nous avons tout d'abord étudié les mécanismes existants dans le domaine de la gestion de la qualité de service, de manière à identifier les champs d'investigation potentiels permettant une contribution dans ce domaine. Nous avons analysé les techniques responsables du choix du débit physique d'émission et les algorithmes capables de gérer les états instantanés du canal. A travers une étude bibliographique, nous avons d'abord dégagé que les principaux algorithmes déployés

par la norme 802.11 pour ce fait sont l'*ARF* (*Auto Rate Fallback*) et l'*AARF* (*Adaptive Auto Rate Fallback*). Nous avons, par la suite, montré que ces algorithmes de sélection de débit ne permettent pas une prise de décision précise et instantanée lorsque le canal est relativement bruité. En effet, ces algorithmes reposent sur un seul paramètre, à savoir le nombre d'acquittements positifs. De ce fait, nous avons proposé une nouvelle technique d'adaptation de débit afin d'améliorer la décision en fonction des conditions instantanées du canal. Cette technique se base sur d'autres paramètres existants et non exploités (à savoir le facteur temps d'aller-retour des paquets connu aussi sous le nom de *RTT* (*Round Trip Time*)) tout en respectant les exigences de la norme et en restant conforme au standard. Cette mesure est intégrée dans le nouvel algorithme de contrôle de lien que nous avons appelé *MAARF* (pour *Modified Adaptive Auto Rate Fallback*) afin de corriger le débit de transmission de données en fonction de la capacité observée du canal. Le nouveau mécanisme *MAARF,* dont le but est de prévoir et de minimiser les pertes inutiles de données, est basé sur le choix d'une valeur de débit convenable de la trame suivante transmise en fonction de la valeur de *RTT* mesurée. L'algorithme effectue alors une correspondance entre la valeur du *RTT* observée et la décision sur le choix du débit.

Les résultats des simulations sous la plateforme *NS-2* pour la validation de ce nouvel algorithme ont permis de mettre en évidence d'importantes améliorations de débit. Au niveau du débit physique, le mécanisme proposé *MAARF* offre des valeurs plus importantes de l'ordre de 17%, et jusqu'à 100% en les comparant avec celles des algorithmes classiques. Il offre aussi des débits de réception allant du simple au double dans le cas d'un canal très bruité. Ce résultat est une conséquence directe de la chute en valeur du nombre de trames erronées par l'application du nouveau mécanisme. Nous avons ainsi agi sur la sélection du débit d'émission physique par l'introduction de nouveaux paramètres afin d'assurer une meilleure connectivité, de minimiser le nombre de trames perdues dans le réseau, et de fournir par conséquent des liens sans fil stables pour le support des applications a caractère multimédia.

L'étude des méthodes existantes pour la différenciation de service dans les réseaux WiFi a conduit à une deuxième contribution de recherche. Nous avons présenté et étudié la complexité et les performances de différentes méthodes de gestion d'accès au canal de transmission favorisant les trafics de priorité haute, y compris la technique la plus récente *EDCA* (*Enhanced Distributed Channel Access*) implémentée dans la nouvelle norme 802.11e du standard de l'IEEE. Cette étude a montré que le mécanisme *EDCA* assure une différenciation de service entre

les différents *ACs* (*Access Category*) des flux d'une même station, mais pas entre les différentes stations du même réseau (accès parallèle au canal par plusieurs flux de la même classe de priorité par plusieurs stations du réseau). Le résultat de cette étude a orienté notre contribution vers l'élaboration d'un nouvel algorithme basé sur l'*EDCA* et améliorant le schéma de gestion du trafic *QoS* en introduisant une nouvelle classification entre les nœuds mobiles. Cette nouvelle priorité concerne l'ordonnancement des stations actives du réseau. Pour départager deux *ACs* du même type et appartenant à différentes stations, nous avons considéré l'historique de la transmission pour chaque station comme un facteur de décision. Notre choix s'est porté sur l'historique des trafics transmis sur le réseau afin de mieux caractériser les stations en terme de qualité des paquets déjà communiqués. De ce fait, nous avons amélioré la méthode d'accès au canal *EDCA* en favorisant les « stations à caractère multimédia », tout en gardant l'ancienne classification entre les types de flux déjà proposée par le 802.11e.

Dans le but de confirmer les résultats théoriques, nous avons complété nos travaux par des simulations. Dans ce cadre, nous avons mené des séries de tests sous la plateforme *NS*-2, nous avons prouvé l'apport de cette révision et avons ainsi confirmé ses avantages pour le support des flux multimédias. En fait, les simulations réalisées montrent que cette nouvelle approche améliore la méthode *EDCA* classique par une réduction nette du taux de collision, ce qui en fait une bonne candidate pour fournir une meilleure gestion de la *QoS* par l'ajout d'une priorité inter-stations.

La dernière contribution réalisée dans le cadre de cette thèse est la proposition d'un nouveau mécanisme de Handover entre les cellules WiFi fournissant une solution efficace pour la réduction de la latence totale de la commutation inter-*APs*. Le but était de diminuer le délai de ce dernier au dessous des 50*ms*, et ainsi de soutenir la plupart des applications temps réel. Nous avons d'abord recensé et détaillé les différents travaux effectués récemment sur ce sujet par une étude bibliographique. Celle-ci nous a permis de montrer leurs insuffisances pour le support de trafic multimédia. Par la suite, une seconde étude portant sur l'estimation de la latence du mécanisme du Handover 802.11 a été conduite afin de détecter les phases pour lesquelles un gain en terme de temps était envisageable. Nous avons conclu qu'en réduisant le temps de la phase de Scan (représentant presque 90% de la latence totale du Handover), le nombre de paquets perdus serait diminué et le temps où la station reste non joignable serait par conséquent réduit. De plus, l'amélioration des ces deux derniers paramètres clés conduit à un respect des contraintes

temporelles des trafics multimédias. Ainsi, nous avons développé une nouvelle solution appelée *PSHP* (pour *Prevent Scan for 802.11 Handoff Procedure*). Elle s'appuie sur une technique permettant de réduire à la fois le temps et le trafic engendré par un Handover. Cette nouvelle technique minimise le temps pendant lequel une station reste hors de contact avec son propre point d'accès *AP* par l'introduction d'une nouvelle phase appelée *phase de pré-scan*. De plus, un nouveau simulateur a été conçu afin de pouvoir analyser et comparer les performances de la technique proposée avec différents algorithmes de la littérature, y compris l'algorithme de base de la norme 802.11. Les résultats des simulations montrent une amélioration significative d'exécution d'un Handover offerte par la nouvelle approche. Les expériences réalisées ont conduit à de meilleures performances que les autres techniques récentes, en respectant les contraintes temporelles liées à la transmission de paquets temps réel. Enfin, un taux de réussite des Handovers supérieur à 95% a été enregistré dans le réseau par la nouvelle technique, offrant ainsi des gains en qualité de lien entre l'*AP* et le mobile.

Les résultats encourageants obtenus tout au long de ces travaux de recherche nous incitent à continuer de proposer de nouvelles techniques et à les implémenter sur des cartes réseau WiFi. Nous envisageons donc de réaliser des implémentations des différents algorithmes élaborés sur des cartes Atheros/MadWifi.

Par ailleurs, nous proposons d'intégrer dans le nouveau mécanisme d'adaptation de lien *MAARF* une procédure de fragmentation pour un choix idéal de la taille des paquets qui deviendra dépendante de la qualité du canal estimée afin de mieux satisfaire la transmission d'applications multimédia à QoS exigée dans ce type de réseau.

De plus, des améliorations peuvent être apportées à la nouvelle révision du schéma d'accès *EDCA* affinant le contrôle du canal par une seule station pour une durée assez longue. En effet avec la nouvelle méthode, il se peut qu'une station prioritaire contrôle le canal jusqu'à ce que ses paquets soient totalement transmis, et qu'aucune autre station ne puisse transmettre des paquets. Nous pouvons ajouter dans une seconde version du mécanisme proposé une procédure de vérification du contrôle d'accès. Ainsi par exemple, si une station contrôle l'accès pour un nombre fixe de paquets, cette nouvelle procédure dégradera la classe de priorité de cette dernière afin d'équilibrer l'accès au médium.

Quant à la nouvelle technique *PSHP* de commutation entre les *APs* du réseau 802.11, nous avons proposé dans d'autres travaux, ne rentrant pas directement dans le cadre de cette thèse, l'optimisation du choix du prochain *AP* pour la procédure du Handover. Nous pouvons en effet envisager que ce choix ne soit plus basé uniquement sur une simple mesure de la qualité du signal entre le mobile et l'*AP*, mais sur d'autres paramètres. Ces idées ont déjà été validées [75] par l'élaboration d'une fonction heuristique pour conduire la station mobile vers un meilleur *AP* produisant un profit maximum pour le Handover réalisé.

ANNEXE

A Nouveau simulateur de Handover 802.11 WHandSim

La modélisation réseau est la définition d'une architecture rendant possible la simulation et l'analyse de performances des éléments constitutifs du réseau. Cette modélisation implique l'utilisation d'un simulateur logiciel capable de définir un réseau en termes de nœuds, liens et technologies. Nous allons présenter, dans cette annexe, une modélisation réseau du Handover intercellulaire dans le cadre des réseaux sans fils de la norme IEEE 802.11. Le traitement de ce nouveau simulateur est entièrement en langage C++. Nous décomposons les traitements complexes de notre simulateur en des fonctions et des procédures de tailles réduites et simples à implémenter afin d'évaluer et comparer la nouvelle technique élaborée *PSHP* (*Prevent Scan for 802.11 Handoff Procedure*) de commutation intercellulaire par rapport aux mécanismes usuels.

Selon l'approche fonctionnelle, la découpe successive d'un problème complexe permet d'aboutir à des problèmes moins complexes et faciles à résoudre. La *Figure A.1* qui suit schématise les trois grands sous-modules qui apparaissent suite à la découpe du traitement global de notre simulateur en des traitements moins complexes : application *AP*, application Client et gestion du trafic *AP*/Client. Le premier sous-module concerne les points d'accès : responsables de créer un nouvel *AP* puis d'afficher ses coordonnées. Le second sous-module regroupe les différents traitements concernant un client à savoir la création d'un client, son association avec un *AP*, l'identification des *APs* voisins, la gestion de la mobilité, etc. Le troisième sous-module est considéré comme un trait d'union entre les deux autres sous-modules. En effet, il permet de visualiser le trafic existant entre les différentes entités réseau (*APs*/clients). Dans la suite, nous nous concentrerons sur les deux sous-modules application client et gestion des *APs*, afin de dégager les traitements et les données qui nous seront utiles lors de la phase d'implémentation.

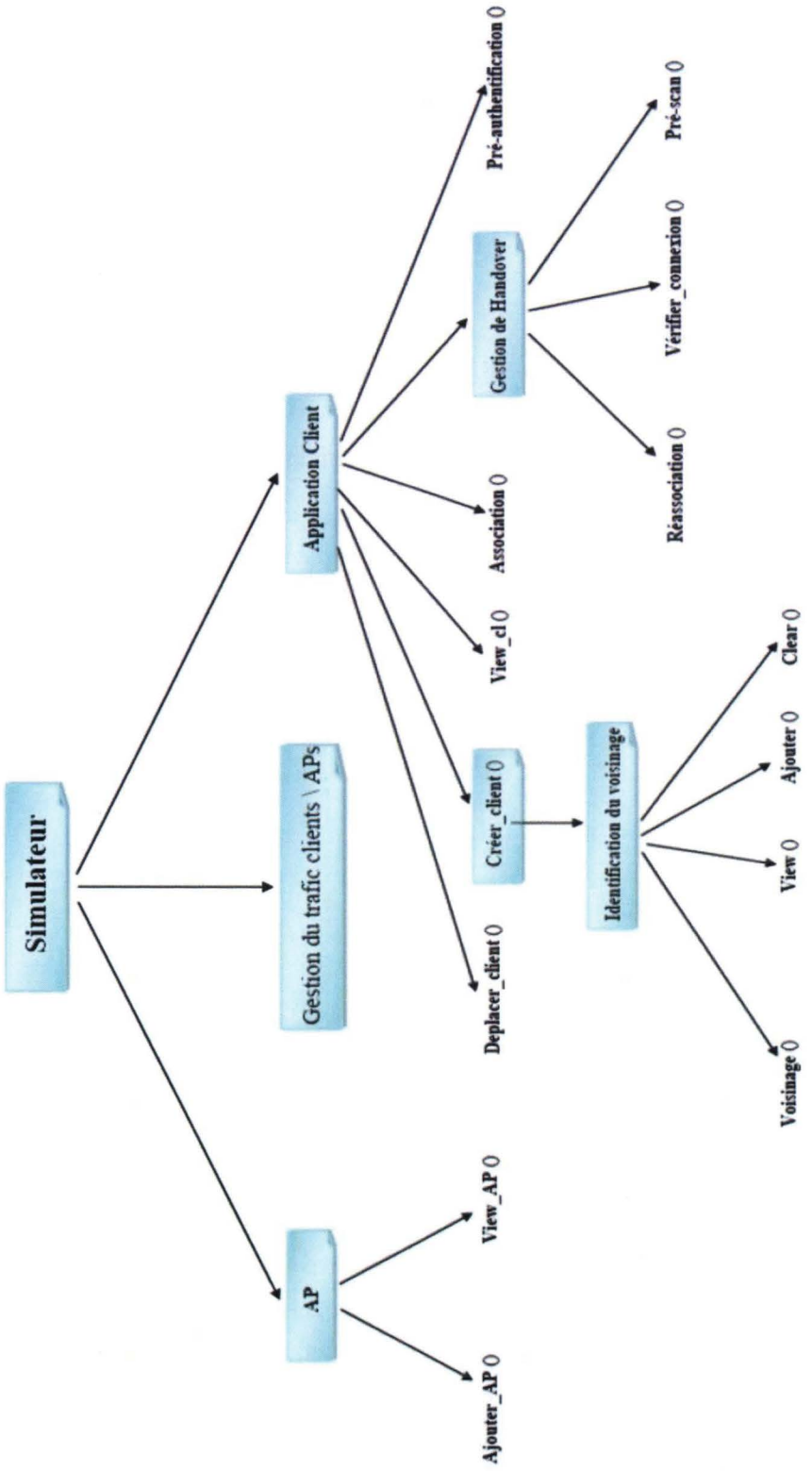

Figure A.1 : Découpe des traitements du simulateur en traitements élémentaires

A. Gestion des APs

La gestion des *APs* est un traitement très important. Le nouveau simulateur doit comporter un sous-module qui se charge de créer un ou plusieurs *APs* de coordonnées (x, y) et de les stocker dans une liste. Les coordonnées (x, y) seront générées aléatoirement lors de la création d'un nouvel *AP*.

Avant de créer un nouvel *AP*, il sera intéressant de définir sa structure. Nous avons défini la structure d'un *AP* comme suit :

```
struct AP
    {    char idap[50];              /* l'identifiant ou le numéro d'un AP */
         int x, y ;                  /* coordonnées d'un AP par rapport au point (0, 0)*/
         char adrip[200];            /* l'adresse IP d'un AP */
         char ESSID[200];            /* l'identifiant de l'ESS à laquelle appartient l'AP */
         struct liste_cl *clients ; /* tableau des clients associés à cet AP*/
         struct AP *suiv;            /* pointeur sur l'AP suivant */
    } ;
struct liste_AP
    {
         struct AP *premier;         /* le premier de la liste des APs */
         struct AP *dernier;         /* le dernier de la liste des APs */
    };
```

Une fois la structure d'un AP définie, la procédure Create_AP prend en charge la création d'un nouvel *AP* en fixant les valeurs de ses paramètres. Ce nouvel *AP* n'est initialement en liaison avec aucun client mobile et sera immédiatement ajouté à la liste des *APs*.

B. Application Client :

Le client mobile est l'élément réseau moteur sur lequel est basée la nouvelle approche d'optimisation du mécanisme de Handover. En effet ce dernier doit être capable de s'adapter

avec les différents scénarii de changement de cellules en particulier lorsqu'un trafic en temps réels est perçu sur le canal de transmission au moment d'un Handover. Cependant, la vérification de la qualité du lien entre un client et son *AP* d'affiliation ainsi qu'avec les autres *APs* existants sur le réseau s'avère primordiale.

Cette vérification est réalisée à travers une opération de *pré-scan* et le mécanisme qui stocke à chaque exécution de cette dernière opération les différentes informations concernant les *APs* ainsi que la qualité du lien que peut offrir chacun d'entre eux dans une liste dynamique triée. La structure de cette liste dynamique est définie comme suit :

```
struct rss
    {
        float valeur;       /* la valeur de RSS */
        char *id;           /* l'identifiant de l'AP auquel correspond la valeur de RSS */
        struct rss *suiv;   /* pointeur vers le nœud rss suivant */
    } ;
struct rss_liste
    {
      struct rss *debut;    /* le premier de la liste des RSS */
      struct rss *fin;      /* le dernier de la liste des RSS */
    };
```

Le sous-module application Client représente la portion qui tient compte des différents traitements qui concernent un client mobile. Cette composante du simulateur doit définir et créer un nouveau client de coordonnées (x, y) et non associé initialement à aucun *AP*. Les coordonnées (x, y) seront générées aléatoirement lors de la création de cette station mobile.

Nous avons défini la structure d'un client comme suit :

```
typedef struct cl
    {
```

```
    char idcl[100];              /* l'identifiant ou le numéro d'un client */
    float rss_courant;           /* le RSS actuel de l'AP */
    int x, y ;                   /* coordonnées du client par rapport au point (0, 0)*/
    AP *ap_courant;              /*l'AP auquel est associé le client*/
    struct liste_AP *voisins;    /* la liste des APs voisins pour le client*/
    char adrmac[200];            /* l'adresse physique de la carte Wifi du client */
    struct rss_liste *rss;       /* liste du RSS du client % à chaque AP */
    struct cl *suiv;             /* pointeur sur le client suivant */
  } cl;
struct liste_cl
        {
            cl *premier;         /* le premier de la liste des clients */
            cl *dernier;         /* le dernier de la liste des clients */
        };
```

Lors de sa création, un client doit identifier les *APs* qui lui sont adjacents (voisins) afin de pouvoir s'associer avec celui qui offre une meilleure qualité de lien.

L'identification des *APs* voisins est considérée comme un sous-module de l'application Client, et par conséquent, elle peut être découpée à son tour en plusieurs traitements élémentaires :

- La création d'une liste chaînée initialement vide et pouvant contenir par la suite les *APs* considérés comme voisins pour le client concerné.
- L'insertion d'un *AP* dans cette liste seulement si ce dernier est considéré comme voisin.
- L'affichage des coordonnées des *APs* présents dans la liste.
- La mise à jour de cette liste dans le cas d'un changement de la structure du voisinage.
- La purge de la liste (lors de l'opération de mise à jour ou de changement de l'*AP* d'association).

- Etc.

Après avoir identifié tous les *APs* présents dans son voisinage, le client doit calculer la distance qui le sépare de chaque *AP* de son voisinage et, en fonction de cette distance, il évalue la valeur de RSSI correspondante pour s'associer à l'*AP* qui possède la meilleure qualité de signal. Dés qu'il est créé, le client est ajouté à la liste des clients.

Suite à son association, le client doit effectuer une opération de *pré-authentification* avec tous les *APs* présents dans l'*ESS* c'est-à-dire ceux qui existent dans la liste des *APs* voisins. La procédure qui s'occupe de la *pré-authentification* du client est Pré-authentification() qui admet comme paramètres le client concerné ainsi que sa liste des *APs* voisins.

Une autre composante intéressante de l'application Client est celle qui gère sa mobilité et qui prend les décisions concernant un Handover. Pour cette raison, cette dernière tâche est considérée comme un sous-module à part qui est responsable des trois traitements suivants :

- **Le pré-scan** : s'effectue périodiquement mais d'une manière discontinue proportionnellement au trafic en compétition sur le canal de transmission. Le résultat de ce pré-scan est une liste triée par valeurs de *RSSI* des *APs* voisins.

  La procédure pré-scan est chargée de construire une nouvelle liste triée selon l'ordre décroissant des valeurs de *RSSI* à partir de la liste des *APs* voisins admise comme une entrée ou paramètre. Le tri par insertion sera choisi lors de l'implémentation (le cas de la majorité des listes chaînées en langage C++). Cette méthode nécessite une boucle principale (séquentielle), dans laquelle on appelle une seule fonction de recherche pour placer l'élément, les déplacements se font rapidement et sans se consacrer au reste de la liste.

  La liste construite sera mise à jour chaque αms. Nous devons tenir compte du type du trafic courant sur chaque canal afin de mettre à jour cette liste. Ce déploiement répétitif nous permettra de bien suivre l'état du réseau et de garder une liste dynamique relative et liée aux événements qui se produisent au cours du temps.

- **La réassociation** : s'effectue quand un client décide de changer son *AP* d'affiliation par un autre AP choisi suite à un scan traditionnel ou à un *pré-scan*. La procédure Réassociation() comportera comme paramètres le client ainsi que le nouvel *AP* choisi.

Cette procédure sera appelée avec les trois différentes formes du Handover qui co-existent avec la nouvelle solution : Handover de première forme, Handover de deuxième forme et Handover de troisième forme.

- **Verifier_connexion** : cette procédure vérifie d'une façon instantanée et périodique la qualité du lien d'un client avec son *AP* d'affiliation. Elle est responsable du passage d'un client d'un état à un autre (de *repos* vers *pré-scan*, de *repos* vers *Handover forme1* ou de *repos* vers *Handover urgent*), ainsi que le déclenchement des deux procédures pré-scan() et Réassociation().

  La procédure réassociation est appelée par la méthode vérifier_connexion() pour les trois formes de Handover discutées antérieurement lors de la proposition de la nouvelle solution.

Les procédures de mobilité, de pré-scan ainsi que l'accomplissement des trois formes de Handover sont réalisées à travers la programmation multitâches (en utilisant la notion des threads) avec une gestion de l'exclusion mutuelle entre les processus et ceci afin d'éviter que les traitements de notre simulateur soient exécutés séquentiellement. La ressource jugée la plus critique de notre simulateur est sans doute la liste dynamique.

C. Visualisation et statistiques

Le nouveau simulateur fournit un support pour analyser les résultats lors d'une expérience. En effet, il inclut une fonctionnalité pour le suivi, le calcul et l'enregistrement de diverses statistiques sur l'ensemble des nœuds mobiles.

Ce dernier interprète un fichier trace (*.log*) contenant des évènements réseau indexés par le temps. Ces évènements sont principalement l'association, la mobilité des clients, l'opération du *pré-scan*, les différentes formes de Handover, *ainsi* que la rupture de lien. De plus, la localisation et les mouvements des nœuds s'ajoutent aux évènements interprétés. A partir de ce fichier *.log* contenant tous les renseignements sur les événements accomplis dans le réseau, nous pourrons extraire et filtrer les informations requises à chaque évaluation, et tracer par la suite, les courbes associées.

Bibliographie

[1] IEEE, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-1999, Aug. 1999.

[2] IEEE 802.11a, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band. Supplement to IEEE 802.11 Standard, Sept. 1999.

[3] IEEE 802.11b, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer Extension in the 2.4 GHz Band. Supplement to IEEE 802.11 Standard, Sept. 1999.

[4] IEEE 802.11d, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specifications for operation in additional regulatory domains. Supplement to IEEE 802.11 Standard, July 2001.

[5] IEEE 802.11e, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service Enhancements. Supplement to IEEE 802.11 Standard, Nov. 2005.

[6] IEEE 802.11f Standard, IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11™ Operation, June 2003.

[7] IEEE 802.11g, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2.4 GHz Band. Supplement to IEEE 802.11 Standard, June 2003.

[8] IEEE 802.11h, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Spectrum and Transmit Power Management Extensions in the 5GHz band in Europe. Supplement to IEEE 802.11 Standard, May 2003.

[9] ETSI TS 101 761-1 V1.3.1, "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 1: Basic Data Transport Functions", January 2002.

[10] IEEE 802.11i, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements. Supplement to IEEE 802.11 Standard, June 2004.

[11] IEEE 802.11j, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: 4.9 GHz–5 GHz Operation in Japan. Supplement to IEEE 802.11 Standard, Sep. 2004.

[12] IEEE 802.11k, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Radio Resource Measurement of Wireless LANs. Supplement to IEEE 802.11 Standard, May 2008.

[13] IEEE 802.11n, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Higher Throughput. Draft Supplement to IEEE Standard 802.11-1999, IEEE 802.11n/D9.0, Mars 2009.

[14] IEEE 802.11r, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Fast basic service set (BSS). Supplement to IEEE 802.11 Standard, May 2008.

[15] IEEE 802.1s Standard, IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks – Multiple Spanning Trees, Dec. 2002.

[16] IEEE Standards for Information Technology. Telecommunications and Information Exchange between Systems– Local and Metropolitan Area Network– Specific Requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. ISO/IEC DIS 8802-11:1997, IEEE Std. 802.11-1997.

[17] IEEE Standards for Information Technology. Telecommunications and Information Exchange between Systems– Local and Metropolitan Area Network– Specific Requirements– Part 11: Supplement to 802.11-1997, Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band. IEEE Std. 802.11-1999.

[18] IEEE Standards for Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements— Part 2: Logical Link Control, May 1998.

[19] IEEE Standards for Local Area Networks— Carrier Sense Multiple Access with Collision Detection (CSMA/CD) — Access Method and Physical Layer Specifications, June 1983.

[20] Ronald L. Rivest. Rc4. Dans Bruce Schneier, Cryptographie Appliquée : protocoles, algorithmes et codes source en C. 2e éd. Paris : Vuibert, 2001, page 419-420.

[21] J. Habetha and D. C. de No, "New Adaptive Modulation and Power Control Algorithms for HIPERLAN/2 Multihop Ad Hoc Networks", in Proc. European Wireless (EW'2000), Dresden, Germany, Sept. 2000.

[22] D. Qiao, S. Choi, and K. G. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs", IEEE Trans. on Mobile Computing (TMC), vol. 1, pp. 278–292, Oct. 2002.

[23] J. del Prado Pavon and S. Choi, "Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement", in Proc. IEEE ICC'03, Anchorage, AK, May 2003.

[24] D. Qiao and S. Choi, "Fast-Responsive Link Adaptation for IEEE 802.11 WLANs", in Proc. IEEE ICC'05, Sept. 2005.

[25] P. Chevillat, J. Jelitto, A. N. Barreto, and H. Truong, "A Dynamic Link Adaptation Algorithm for IEEE 802.11a Wireless LANs", in Proc. IEEE ICC'03, Anchorage, AK, May 2003.

[26] A. Koepsel, J.-P. Ebert, and A. Wolisz, "A Performance Comparison of Point and Distributed Coordination Function of an IEEE 802.11 in the Presence of Real-Time Requirements," In Proc. of 7th Workshop on Mobile Multimedia Comm., Tokyo, Oct. 2000.

[27] IEEE 802.11 WG. Draft Supplement to Standard 802.11-1999: Medium Access Control (MAC) Enhancements for Quality of Service (QoS). IEEE 802.11e/D2.0a, November 2001.

[28] IEEE 802.11 WG. Draft Supplement to IEEE Standard802.11-1999: Medium Access Control (MAC) Enhancements for Quality of Service (QoS). IEEE 802.11e/D4.3, May 2003.

[29] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," Informational RFC; July 1999.

[30] R. Braden, L. Zhang, et al., "Resource ReSerVation Protocol (RSVP)" - Version 1: Technical Specification; Standards Track RFC; septembre 1997.

[31] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," In Proc. of the European Wireless, Vol. 1, pp. 32–39, Florence, Italy, Feb. 2002.

[32] IEEE Standard 802.11k, "Radio Ressource Management", IEEE Standard 802.11- 2003.

[33] A. Mishra, M. Shin, N. Petroni, T. Clancy, and W. Arbaugh, "Proactive Key Distribution Using Neighbor Graphs," IEEE Wireless Communications Magazine, vol. 11, no. 1, pp. 26-36, Feb. 2004.

[34] A. Kamerman and L. Monteban, "WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band", Bell Labs Technical Journal, pp. 118–133, Summer 1997.

[35] M. Lacage, M. H. Manshaei, and T. Turletti. "IEEE802.11 Rate Adaptation: A Practical Approach". INRIA Research Report number 5208, May2004.

[36] J. Tourrilhes, "Fragment Adaptive Reduction: Coping with Various Interferers in Radio Unlicensed Bands", in Proc. IEEE ICC'01, Helsinki,Finland, July 2001.

[37] A. Kumar, J. Holtzman, "Performance analysis of versions of TCP in WLAN", Indian Academy of Sciences Proceedings in Engineering Sciences, Sadhana, Feb 1998.

[38] Stéphane Mocanu, "évaluation de performances TCP Reno et Vegas", Laboratoire d'Automatique de Grenoble, Département Télécom 2003-2004.

[39] ns-2, available at http://www.isi.edu/nsnam/ns/

[40] Mingzhe Li, "An Extension of Rate-Adaptive MAC Protocol for NS2 Simulator", Computer Science Department of Worcester Polytechnic Institute, Worcester, MA 01609.

[41] D. Estrin, M. Handley, J. Heidemann, S. McCanne, Y. Xu, H. Yu, "Network Visualization with the VINT Network Animator NAM", Technical report, Technical Report 99-703, University of Southern California, Los Angeles, Mars 1999. Accepted to Appear in IEEE Computer Magazine, Nov. 1999.

[42] IEEE 802.11 WG. Draft Supplement to IEEE Standard 802.11-1999: Medium Access Control (MAC) Enhancements for Quality of Service (QoS). IEEE 802.11e/D5.0, 2003.

[43] ANSI/IEEE. 802.1D: Media Access Control (MAC) Bridges. IEEE, 1998.

[44] Jeng Farn Lee, Wanjiun Liao, and Meng Chang Chen, "A Per-Class QoS Service Model in IEEE 802.11e WLANs," In Proc. of the 2nd Int'l. Conf. on Quality of Service in Heterogeneous Wired/Wireless Networks (Qshine'05), Orlando, August 2005.

[45] M. Malli, Q. Ni and C. Barakat, "Adaptive Fair Channel Allocation for QoS Enhancement in IEEE 802.11," The IEEE Conference on Communications (ICC), Paris, France, June 2004.

[46] M. Shreedhar and G. Varghese, "Efficient Fair Queuing Using Deficit Round-robin," IEEE Trans. Networking, vol. 4, pp. 375-85, 1996.

[47] Q. Ni, L. Romdhani and T. Turletti, "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN", Journal of Wireless Communications and Mobile Computing, Wiley. 2004: Volume 4, Issue 5: pp.547-566.

[48] I. NIANG, H. AFIFI and D. SERET, "Couplage de services différenciés pour une QoS de bout en bout", Colloque Africain sur la Recherche en Informatique, CARI'04, Hammamet, Tunisie, Novembre 2004.

[49] A. Bedoui, K. Barkaoui and K. Djouani, "Approche Globale pour la Qualité de Service dans les Réseaux Locaux sans Fil IEEE 802.11", ISPS'05, Alger, mai 2005.

[50] Y., S. Park, S. Choi, G. Lee, J. Lee, and H. Jung, "Enhancement of a WLAN-Based Internet Service", ACM Mobile Networks and Applications, vol. 10, no. 3, June 2005, pp. 303-314.

[51] H. Velayos, G. Karlsson, "Techniques to Reduce IEEE 802.11b MAC Handover Time Layer", Technical Report TRITA-. IMIT-LCN R 03:02, ISSN 1651-7717, April 2003.

[52] A. Mishra, M. Shin, and W. Arbaugh. "An empirical analysis of the IEEE 802.11 MAC layer handoff process". ACM Computer Communication Review, 33(2) :93 102, 2003

[53] H. Velayos, G. Karlsson, "Techniques to Reduce IEEE 802.11b MAC Handover Time Layer", Technical Report TRITA-. IMIT-LCN R 03:02, ISSN 1651-7717, April 2003.

[54] S. Waharte, K. Ritzenthaler and R. Boutaba, "Selective active scanning for fast handoff in wlan using sensor networks", IEE International Conference on Mobile and Wireless Communication Networks (MWCN'04), Paris, France, October 2004.

[55] I. Ramani and S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks," in Proc. IEEE Infocom 2005, March 2005.

[56] Ping-Jung Huang, Yu-Chee Tseng, Kun-Cheng Tsai,"A Fast Handoff Mechanism for IEEE 802.11 and IAPP Networks" IEEE 802.11 Working Group, 2006.

[57] H. Kim, S. Park, C. Park, J. Kim, and S. Ko, "Selective Channel Scanning for Fast Handoff in Wireless LAN using Neighboor Graph," ITC-CSCC 2004, July 2004.

[58] Radius, RFC 2865 et 2866, http www.ietf.org rfc rfc2865.txt, http www.ietf.org rfc rfc2866.txt

[59] *Venkata M. Chintala and Qing-An Zeng, "Novel MAC Layer Handoff Schemes for IEEE 802.11 Wireless LANs". The IEEE Wireless Communications and Networking Conference 2007, WCNC 2007, Mars 2007.*

[60] *P. Roshan and J. Leary, "802.11 Wireless LAN Fundamentals, CISCO Press", ISBN No.1587050773.*

[61] *M.Raghavan, A. Mukherjee, H. Liu, Q-A. Zeng, and D. P. Agarwal, "Improvement in QoS for Multimedia Traffic in Wireless LANs during Handoff," Proceedings of the 2005 International Conference on Wireless Networks ICWN'05, Las Vegas, Nevada, USA, pp. 251-257, June 27-30, 2005.*

[62] *IEEE Standard 802.11F, "IEEE trial-use recommended practice for multi-vendor access point interoperability via an inter-access point protocol across distribution systems was supporting IEEE 802.11 operation", 2003.*

[63] *C. Perkins, "IP Mobility Support". RFC 2002 (Proposed Standard): ftp.rfc-editor.org in-notes rfc2002.txt, October 1996.*

[64] *C. Perkins, "IP Mobility Support for IPv4".RFC 3344 (Proposed Standard): ftp.rfc-editor.org in-notes rfc3344.txt, August 2002.*

[65] *D.Johnson, C.Perkins, and J.Arkko, "Mobility Support in IPv6". RFC 3775 (Proposed Standard): ftp.rfc-editor.org in-notes rfc3775.txt, June 2004.*

[66] *S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration". RFC 2462 (Draft Standard): ftp.rfc-editor.org in-notes rfc2462.txt, December 1998.*

[67] *Orinoco Technical Bulletin TB 034/A, "Inter Access Point Protocol (IAPP)", Agere Systems, February 2000.*

[68] *P. Roshan and J. Leary, "802.11 Wireless LAN Fundamentals", CISCO Press, ISBN No.1587050773, January 2004.*

[69] *Ping-Jung Huang, Yu-Chee Tseng, Kun-Cheng Tsai, "A Fast Handoff Mechanism for IEEE 802.11 and IAPP Networks," IEEE 802.11 Working Group, 2006.*

[70] *Teruaki Kitasuka, Kenji Hisazumi, Tsuneo Nakanishi and Akira Fukuda, "Positioning Technique of Wireless LAN Terminals Using RSSI between Terminals", Proc. the 2005 International Conference on Pervasive Systems and Computing (PSC-05), pp. 47-53, Las Vegas, Nevada, USA, June 27-30, 2005.*

[71] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks", in T. Imelinsky and H. Korth, editors, Mobile Computing, pp. 153-181, Kluwer Academic Publishers, 1996.

[72] D. Pong and T. Moors, "The Impact of Random Waypoint Mobility on Infrastructure Wireless Networks", Proceedings of the 11th International Conference on Parallel and Distributed Systems, vol.2, pp. 140-144, Fuduoka, Japan, July 2005.

[73] J-Y Boudec, "On the Stationary Distribution of Speed and Location of Random Waypoint", the IEEE Transactions on Mobile Computing, vol. 4, pp. 404-405, Jul/Aug, 2005.

*[74] **A. R. Rebai**, M. Fliss, S. Jarboui and S. Hanafi, "A New Link Adaptation Scheme for IEEE 802.11 WLANs", The 2nd IEEE International Conference on Technologies, Mobility and Security NTMS'08, Tangier, Morocco, Nov. 2008.*

*[75] **A. R. Rebai** and S. Hanafi, "An Enhancement Heuristic Algorithm for Efficient WLAN IEEE802.11 Handoff Procedure", The 2nd International Conference on Metaheuristics and Nature Inspired Computing META'08, Hammamet, Tunisia, Oct. 2008.*

*[76] **A. R. Rebai** and S. Hanafi, "A Quality Improvement Algorithm for 802.11e EDCA Model", The 8th IEEE International Conference on Networks ICN'09, Cancun, Mexico, March 2009.*

*[77] **A. R. Rebai**, B. Haddar and S. Hanafi, "Prevent-Scan: A novel MAC Layer scheme for the IEEE 802.11 handoff", The 2nd IEEE International Conference on Multimedia Computing and Systems ICMS'09, Page(s):541 – 546, Ouarzazate, Morocco, April 2009.*

*[78] **A. R. Rebai**, B. Haddar and S. Hanafi, "An efficient fast WLAN IEEE 802.11 Handoff procedure", The 5th IEEE International Conference on Networking and Services ICNS'09, Valencia, Spain, April 2009.*

*[79] **A. R. Rebai**, S. Hanafi and H. Alnuweiri, "A New Inter-Node Priority Access Enhancement Scheme for IEEE_802.11 WLANs", The 9th International Conference on ITS Telecommunication ITST'09, Lille, France, Oct. 2009.*

*[80] **A. R. Rebai**, H. Alnuweiri and S. Hanafi, "A Novel Prevent-Scan Handoff Technique for IEEE 802.11 WLANs", The annual IEEE International Conference on Ultra Modern Telecommunications ICUMT'09, St.-Petersburg, Russia, Oct. 2009.*