

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté de Technologie
Département de Génie Electrique et Electronique



Ecole doctorale Sciences et Technologies de l'Information et Télécommunication

Mémoire de Magister en Systèmes et Réseaux de Télécommunication

Sous le thème :

Extension de la QoS du Wifi vers le WiMAX

Réalisé par :

Mme. Sedjelmaci Amina Nadjat

Membres du jury :

Président	:	Mr. Feham Med.	Professeur	UABB Tlemcen
Examineur	:	Mr. Chikh Med. Amin	MCA	UABB Tlemcen
Encadreur	:	Mme. Didi Fedoua	MCA	UABB Tlemcen
Co-encadreur	:	Mme. Labiod Houda	MC	TELECOM Paritech France
Invité	:	Mr Lehsaïni Med.	MCB	UABB Tlemcen

Année universitaire : 2010-2011

À MA GÉNÉREUSE PETITE FAMILLE

.

REMERCIEMENT

Je remercie Dieu le Tout-Puissant pour nous avoir donné le potentiel d'apprendre les lois de notre univers et de les exploiter pour notre bien-être.

Ce travail a été accompli à l'aide de plusieurs personnes que je tiens à remercier absolument. Je remercie en premier lieu mon encadreur Mme. *Didi Fedoua* ainsi que mon co-encadreur Mme. *Labiod Houda* de m'avoir apporté leurs valeureux conseils et soutiens durant la réalisation de cette thèse. Je leur transmets l'expression de ma reconnaissance et de ma plus profonde gratitude.

Je remercie énormément *Pr. Feham Med.* de m'avoir fait l'honneur de présider ma soutenance. Je remercie également *Dr. CHIKH Med. Amine* et *Dr. Lehsaïni Med.* de m'avoir fait l'honneur en acceptant d'examiner mes travaux de thèse de magister et faire partie de mon jury.

Je remercie sincèrement mes chères mère et père que j'aime énormément. Je remercie aussi mon très cher mari et mes beaux parents. Je tiens aussi à remercier mes chers trois frère Farid, Djallel et Fouad ainsi que mes deux belles sœurs Ranwa et Kamila.

Finalement, je remercie tous ceux qui ont participé de prêt ou de loin dans l'élaboration de ce travail avec un conseil ou autre parmi mes camarades de magistère, mes amis, mes étudiants et collègues à la chambre du commerce de Tlemcen.

RESUME

Le monde des réseaux informatiques a connu il y a quelques années une révolution explosive qui continue son escalade. Dans cette révolution, les diverses technologies se trouvent attirées par un champ de gravité fort et croissant, généré par l'étoile brillante de l'univers informatique qui est l'Internet. Elles s'orientent malgré elles vers un centre unique subissant ainsi un croisement de chemin inévitable illustré par la convergence de plusieurs solutions réseaux, considérées jadis comme mutuellement exclusives.

Dans notre travail nous avons abordé la problématique liée à l'optimisation de la convergence entre deux technologies de réseau sans fil très prometteuses qui sont le Wifi et le WiMAX. Nous proposons une solution pour une configuration pratique où les deux réseaux convergent en une structure complémentaire et constructive en contraste à d'autres cas où la convergence serait compétitive.

Notre contribution se résume ainsi, à la proposition d'une stratégie de couplage du contrôle d'accès dans les deux réseaux dans le but d'améliorer la gestion globale de la qualité de service dans le nouveau réseau composite. Nous proposons une implémentation de la stratégie sous forme d'un algorithme de contrôle d'admission combiné. Ce dernier est implémenté et évalué par la simulation en utilisant le logiciel spécialisé libre et open source « NS-3 ».

Mots clés : Convergence Wifi-Wimax, Gestion de la QoS, Réseaux locaux sans-fil, Réseaux métropolitains sans-fil, Contrôle d'accès au médium, Simulation NS-3, Interface NS-3 - Scilab.

ملخص

منذ عدة أعوام بدأ عالم شبكات الحاسوب يشهد تطورا مدويا مازال مستمرا وبتزايد. أمست التقنيات المختلفة خلال هذا التطور تحت وطأة مجال جاذبية ذي قوة متضاعفة مصدره شمس مجرة الحاسوب, ألا وهي شبكة "الإنترنت". تجد تلك التقنيات نفسها مرغمة على التوجه نحو مركز متوحد وبالتالي مضطرة للتقاطع بينها في ما يتمثل بالتقارب ما بين حلول شبكاتية كانت حتى أمد قريب تعتبر منفصلة تماما.

نحاول خلال عملنا هذا تناول إشكالية تحقيق تقارب مثالي ما بين شبكتين لاسلكيتين واعدتين هما شبكة "ويفي" وشبكة "ويماكس". بالتالي نقترح مثلا على طريقة تركيب عملية حيث تتقارب الشبكتين بشكل تعاوني بناء لا تنافسي.

هكذا تكون مشاركتنا الفعلية متمثلة باقتراح إستراتيجية لدمج التحكم في قبول التوصيل في كلتا الشبكتين بهدف تحسين الأداء العام للنوعية في الشبكة المركبة. نقترح بالتالي تطبيق هته الإستراتيجية على شكل خوارزمية للتحكم في قبول الاتصال حيث نبرمج ونختبر هذه الخوارزمية باستخدام البرنامج المتخصص والمفتوح NS3 .

كلمات مفتاحية : تقارب الويفي والويماكس, تسيير الجودة النوعية, شبكات محلية لاسلكية, برنامج التحاكي NS3 .

ABSTRACT

A number of years ago, the world of computer networking have witnessed an explosive evolution which continues to rise. In this evolution, the multiple technologies found themselves attracted in an increasingly strong gravity field, generated by the shining star in the computer universe that is known as the “Internet”. They are being helplessly directed to a common center, consequently facing a crossroad illustrated by the convergence of numerous networking solutions that have been yet considered mutually exclusives.

In our work, we try to approach the problematic related to the optimization of the convergence of two very promising wireless networking technologies which are Wifi and WiMAX. We propose the example of a practical configuration where the two networks converge in a complementary and constructive structure rather than a competitive one.

Thus, our contribution is summed in the proposition of a coupling strategy between the access controls in the two networks in the aim of enhancing the global quality of service management in the new composite network. We propose the implementation of that strategy in the form of a combined admission control algorithm that is implemented and verified through simulation, using the specialized free open source software “NS-3”.

Keywords : Wifi-Wimax convergence, QoS management, Wireless LAN, Wireless MAN, Media access control, NS-3 simulation, NS-3 – Scilab interface.

TABLE DES MATIERES

Liste des figures.....	8
Liste des tableaux.....	11
I Généralités sur les technologies réseaux sans fil.....	16
I.1 Introduction	16
I.2 Historique.....	16
I.2.1 Les Origines[1]	16
I.2.2 Historique Récent[2]	19
I.3 Architecture des réseaux sans fil.....	21
I.4 Classification des réseaux sans fil.....	26
I.5 Les Tendances actuelles et leur évolution	28
I.5.1 Développement Récent[4]	28
I.5.2 Evolution	31
I.6 Conclusion	37
II La QoS dans les réseaux Wifi et WiMAX.....	38
II.1 Introduction	38
II.2 Généralités sur le réseau Wifi [5].....	39
II.2.1 Normalisation	40
II.2.2 Architecture.....	42
II.2.3 La gestion de QoS dans le WiFi.....	53
II.2.4 Les Modèles de gestion de la QoS basés DCF	54
II.2.5 Modèle de gestion de la QoS basée EDCA.....	56
II.2.6 Conclusion	57
II.3 Généralités sur le réseau WiMAX.....	58
II.3.1 Normalisation et le standard IEEE802.16	58
II.3.2 La structure WiMAX.....	60
II.3.3 La gestion de la QoS dans le réseau WiMAX.....	76
II.4 Conclusion	79
III Extension de la QoS Wifi au WiMAX.....	80
III.1 INTRODUCTION	80

III.2	Généralité sur la convergence Wifi-WiMAX	80
III.3	Notre proposition	84
III.4	Choix du simulateur.....	86
III.5	Simulation NS-3	89
III.5.1	Implémentation du Wifi dans NS-3:	92
III.5.2	Implémentation du WiMAX dans NS-3:.....	95
III.5.3	Les détails de l'implémentation de notre proposition dans NS-3.....	98
III.6	Résultats et Discussion	102
III.7	Conclusion	114
	Conclusion et Perspectives	115
	Bibliographie	117
	Liste des Abréviations	119
	Annexe.....	123

LISTE DES FIGURES

Figure I.1 : Architecture d'un réseau de télécommunication standard	21
Figure I.2 : Architecture d'un réseau de télécommunication informatique	22
Figure I.3 : Les différentes couches réseaux.....	22
Figure I.4 : Construction des PDUs dans l'architecture OSI	23
Figure I.5 : Procédures de transmission	24
Figure I.6 : Types de connexion.....	25
Figure I.7 : Modèle de communication sans fil	25
Figure I.8 : Aperçue de différentes classes de réseau sans fil	27
Figure I.9 : L'environnement CRE avec des multiple RATs.	33
Figure I.10 : Architecture d'un terminal qui opère dans un environnement CRE.	34
Figure I.11 : Architecture d'un terminal intégrant le concept de reconfigurabilité	35
Figure I.12 : Les différents types d'allocation de spectre.....	36
Figure I.13 : Architecture d'un système MS-CRE	37
Figure II.1 : Mode infrastructure de l'IEEE 802.11	43
Figure II.2 : Le mode Ad hoc.....	44
Figure II.3 : Le mode mesh WLAN contenant des MPs, MAPs et STAs	45
Figure II.4 : Le modèle de référence OSI, IEEE 802.11 et la famille 802.....	46
Figure II.5 : Méthodes d'accès au media 802.11.	48
Figure II.6La méthode d'accès CSMA/CA	50
Figure II.7La méthode d'accès CSMA/CA avec RTS/CTS.....	51
Figure II.8 : Exemple d'accès au média basé EDCA	52
Figure II.9 : Les Différentes configuration du WiMAX.....	58
Figure II.10 : Structure des couches WiMAX	61
Figure II.11Connexion et flux de service dans le WiMAX.....	62
Figure II.12 Flux de données dans les différentes couches MAC.....	62
Figure II.13 : Une trame MAC WiMAX.....	64
Figure II.14 : Les deux types d'entête MAC du WiMAX.....	65
Figure II.15 : La Fragmentation de la trame MAC du WiMAX	66
Figure II.16 : La Concaténation de la trame MAC du WiMAX.....	66
Figure II.17 : Etablissement de connexion sécurisée WiMAX entre une BS et une station cliente	67
Figure II.18 : Modulation simple porteuse (à gauche) Modulation OFDM (à droite)	68
Figure II.19 : La Modulation OFDMA	69
Figure II.20 : Le duplexage FDD.....	69
Figure II.21 : Le duplexage TDD.....	70
Figure II.22 : Structure d'une trame OFDMA WiMAX	71
Figure II.23 : Topologie PMP	72

Figure II.24 : Topologie Mesh.....	72
Figure II.25 : Algorithme pour l'établissement d'une connexion	73
Figure II.26 : Algorithme d'adaptation	75
Figure II.27 : Architecture de la gestion de la QoS dans le WiMAX.	76
Figure II.28 : Un modèle de planification de ressource dans le WiMAX	78
Figure III.1 : Infrastructure IEEE802.21.....	81
Figure III.2 : Le fonctionnement de l'idée proposée dans [1].....	82
Figure III.3 : Couplage étroit Wifi-UMTS.....	83
Figure III.4 : Infrastructure réseau composite proposée.....	84
Figure III.5 : Comparaison du temps de calcul en fonction de la taille du réseau.	87
Figure III.6 : Comparaison de l'utilisation de la mémoire en fonction de la taille du réseau...88	
Figure III.7 : Les additions principales dans NS-3 durant la dernière année.	88
Figure III.8 : La croissance du cumule de nombre de ligne de code C++ (axe y) contribué dans NS-3 dans le temps (axe x).	89
Figure III.9 : Les principaux modules de NS-3	90
Figure III.10 : L'architecture de nœud NS-3	91
Figure III.11 : L'architecture du WifiNetDevice dans NS-3.	94
Figure III.12 : Modèle NS-3 de propagation dans le Wifi	95
Figure III.13 : L'architecture NS-3 du WiMAX	97
Figure III.14: Variation du délai moyen pour les flux Vidéo Wifi en fonction du rayon de distribution des nœuds.	102
Figure III.15 : Variation du nombre de paquets perdus moyen pour les flux Vidéo Wifi en fonction du rayon de distribution des nœuds.....	103
Figure III.16 : Variation du délai moyen pour les flux Voix Wifi en fonction du rayon de distribution des nœuds.	103
Figure III.17 : Variation du nombre de paquets perdus moyen pour les flux Voix Wifi en fonction du rayon de distribution des nœuds.....	104
Figure III.18 : Variation du délai moyen pour les flux de données Wifi en fonction du rayon de distribution des nœuds.	104
Figure III.19 : Variation du nombre de paquets perdus moyen pour les flux de données Wifi en fonction du rayon de distribution des nœuds.....	105
Figure III.20 : Variation du délai moyen pour les flux Vidéo WiMAX en fonction du rayon de distribution des nœuds.	106
Figure III.21 : Variation du nombre de paquets perdus moyen pour les flux Vidéo WiMAX en fonction du rayon de distribution des nœuds.....	106
Figure III.22 : Variation du délai moyen pour les flux Voix WiMAX en fonction du rayon de distribution des nœuds.	107
Figure III.23 : Variation du nombre de paquets perdus moyen pour les flux Voix WiMAX en fonction du rayon de distribution des nœuds.....	107

Figure III.24 : Variation du délai moyen pour les flux de données WiMAX en fonction du rayon de distribution des nœuds.....	108
Figure III.25 : Variation du nombre de paquets perdus moyen pour les flux de données WiMAX en fonction du rayon de distribution des nœuds.	108
Figure III.26 : Variation du délai moyen pour les flux Vidéo Mixte en fonction du rayon de distribution des nœuds.	109
Figure III.27 : Variation du nombre de paquets perdus moyen pour les flux Vidéo Mixte en fonction du rayon de distribution des nœuds.....	110
Figure III.28 : Variation du délai moyen pour les flux Voix Mixte en fonction du rayon de distribution des nœuds.	110
Figure III.29 : Variation du nombre de paquets perdus moyen pour les flux Voix Mixte en fonction du rayon de distribution des nœuds.....	111
Figure III.30 : Variation du délai moyen pour les flux de données Mixte en fonction du rayon de distribution des nœuds.	111
Figure III.31: Variation du nombre de paquets perdus moyen pour les flux de données Mixte en fonction du rayon de distribution des nœuds.....	112
Figure III.32 : Nombre d'applications admissibles en fonction du rayon de distribution pour les nœuds Wifi et WiMAX.	113
Figure III.33: Nombre d'applications admissibles en fonction du rayon de distribution pour les nœuds Wifi, WiMAX et Mixte.....	114

LISTE DES TABLEAUX

Tableau I-1 : Classification d'onde d'émission radio	28
Tableau II-1 : Les normes dérivées du Wifi.....	41
Tableau II-2 : les différents débits en fonction de la modulation	47
Tableau II-3 : Les valeurs du SIFS, CW_{min} , CW_{max} en fonction de la modulation	50
Tableau II-4 : Exemple de classification de TID en AC.....	53
Tableau II-5 : différentes normes du WiMAX harmonisez les titres	59
Tableau II-6 : Les différentes classes du QoS dans le WiMAX	63
Tableau II-7 : Les différents types de la couche physique dans le WiMAX.	68
Tableau II-8 : Une correspondance entre les principales classes de QoS dans les réseaux Wifi et WiMAX.....	79

INTRODUCTION GENERALE

« Que seront les réseaux du futur ? » une question qui fait le titre de plusieurs articles. Une préoccupation justifiée vu que le mobile est devenu un outil quotidien indispensable pour pratiquement une personne sur deux dans le monde, créant une concurrence rude poussant les opérateurs à toujours vouloir faire mieux et plus.

Une réponse à cette question primordiale devrait s'inspirer de l'histoire de la téléphonie mobile pour en faire sortir la dynamique d'évolution de ce domaine. On soupçonne bien que les éléments techniques ne sont pas les seuls facteurs de l'évolution des réseaux. En effet aujourd'hui, dans l'univers des réseaux sans fil, on ne parle pas d'une technique gagnante ou de la solution « x » à choisir absolument. En revanche, on parle de générations de technologies, en d'autres termes une évolution parallèle de plusieurs technologies. Cette évolution, quelque peu indépendante, n'est pas forcément concurrente, bien au contraire on aperçoit dernièrement un phénomène majeur de « convergence ». Ce dernier fait l'objet d'un grand nombre de recherches actuelles dans le domaine des réseaux sans fil où il est souvent associé au terme NGN (Next Generation Network) qui désigne l'ensemble des réseaux de la nouvelle génération.

Dans ce travail, nous avons décidé d'aborder la question de la convergence entre deux technologies des réseaux sans fil très intéressantes, le Wifi et le WiMAX. La première ayant déjà confirmé sa dominance dans le monde des réseaux sans fil locaux et le deuxième possédant tous les atouts du Wifi pour réussir à devenir le réseau sans fil métropolitain/global du futur. Nous allons principalement nous focaliser sur l'aspect QoS (Quality of Service) de cette convergence qui n'est désormais plus une idée abstraite ou un simple thème de recherche. On trouve déjà sur le marché des appareils qui intègrent simultanément une interface Wifi et une autre WiMAX.

De multiples travaux de recherche ont jadis ouvert la porte sur l'interopérabilité entre le réseau Wifi et le réseau WiMAX en considérant le scénario où ces deux réseaux coexisteraient entre eux et où chaque réseau assure sa propre gestion de la qualité de service. Ces travaux se sont concentrés principalement sur le passage entre les deux réseaux tout en essayant d'assurer une continuation de la qualité de service et un passage ou « handover » le plus transparent possible. Jusqu'à présent, la spécialisation était la caractéristique principale de ces moyens de communications puisque chaque réseau permettait l'accès à un service particulier à travers une infrastructure particulière. Ceci obligeait les utilisateurs à s'abonner à plusieurs réseaux et à utiliser un terminal spécifique pour accéder aux services transmis par chacun d'eux. Les barrières qui séparaient auparavant les différents types de réseaux de communications commencent à céder les unes après les autres et les terminaux d'accès deviennent multimédia et multiservices intégrant plusieurs interfaces de communication leur permettant de se connecter à différents réseaux.

Actuellement, les acteurs des télécommunications s'acheminent vers la notion de convergence qui regroupera tous les réseaux et tous les services sous une seule infrastructure censée représenter le réseau de nouvelle génération NGN (Next Generation Network). Les différents acteurs s'accordent à dire que la technologie IP sera la brique de base pour bâtir les NGNs. En effet, la simplicité et la puissance du protocole IP, démontrées dans le réseau Internet, fait de ce dernier la technologie de prédilection qui offre un compromis entre le coût de déploiement et l'efficacité de fonctionnement.

La technologie IP est confortée par le concept Tout-IP dont l'objectif est de faire migrer tous les services traditionnels vers la technologie IP. Toutefois, la réalisation des NGNs engendre plusieurs défis techniques (Qualité de Service, hétérogénéité, sécurité, Handover, etc.) qu'il faudra relever pour basculer définitivement d'un concept théorique vers une architecture réellement exploitable. La problématique de la qualité de service, représente l'un des plus grands défis. En effet, les services de téléphonie et de diffusion TV possèdent des contraintes de QoS difficiles à respecter dans les réseaux actuels de données et plus particulièrement dans les réseaux d'accès sans fil du standard 802.11. C'est dans ce cadre là que s'inscrivent les travaux de cette thèse, proposer des solutions de gestion de la QoS pour le support d'applications temps réel et multimédia dans les réseaux locaux sans fil afin de faciliter leur intégration dans les NGNs, et plus particulièrement lors de passages d'un réseau vers un autre.

D'où notre intérêt pour ce type de problème en particulier, comment garantir la QoS d'un utilisateur qui est déjà en train de transmettre avec certains paramètres de qualité de service sur un réseau et doit basculer sur un autre réseau car il est soit, mobile ou le réseau d'origine sature et donc pour éviter la perte de QoS. En particulier lors du basculement d'un réseau Wifi vers WiMAX, comment doit se faire ce basculement, et est ce que les paramètres de QoS sont conservés ? Donc dans cette thèse, nous nous intéresserons aux paramètres sur lesquels il faut se baser pour faire le basculement du Wifi vers le WiMAX, et mettre au point les mécanismes nécessaires pour garantir et gérer la QoS déjà allouée aux utilisateurs de sorte qu'elle continue d'être préservée et que le passage se fasse de manière transparente aux utilisateurs.

La question principale qui guidera notre travail est « Comment tirer le plus de bénéfice de l'association entre le Wifi avec son aspect libre et facile et le WiMAX avec sa longue portée et sa haute mobilité ? ». Ainsi nous allons proposer un scénario de collaboration entre les deux réseaux tout en tenant compte de la réalité économique. Par conséquent, nous étudierons le cas où un réseau Wifi est supposé déjà existant et où on vient lui superposer un réseau WiMAX dans une architecture collaborative. Les deux réseaux sont ainsi couplés par une gestion de la QoS centralisée. Notre but est de proposer une stratégie de couplage constructive qui guidera cette gestion centralisée dans la décision d'admettre ou pas un flux de données au sein de l'un des deux réseaux.

Afin d'aboutir à notre but, nous allons étudier les mécanismes de fonctionnement des deux réseaux en se concentrant sur ceux de la gestion de la QoS. Nous commencerons par le réseau Wifi où nous allons exposer quelques notions de base de la famille de normes IEEE802.11 pour pouvoir ensuite étudier avec plus de détail l'amendement IEEE802.11^e qui a introduit les principaux mécanismes de gestion de la QoS dans les réseaux Wifi. Nous exposerons ensuite le fonctionnement du réseau WiMAX avec sa gestion de la QoS, qui fut un pilier principal dès la création de la norme originale IEEE802.16.

En deuxième lieu nous allons introduire les principaux critères de notre stratégie de gestion de la QoS collaborative tout en construisant la méthodologie pour l'établissement des détails et l'implémentation pratique de cette stratégie. Cette construction devrait passer par une étape de simulation où nous allons utiliser le simulateur NS3 et une étape d'interprétation statistique en utilisant le logiciel Scilab, tous les deux fonctionnant sous Linux Fedora. A ce niveau, nous avons choisi d'opter pour ces outils informatiques pour supporter l'esprit de l'open source et la licence libre. Le potentiel d'évolution démontré par le simulateur NS3 nous a convaincu de considérer la maîtrise de cet outil comme partie intégrante des motivations de ce travail.

Le manuscrit est organisé en trois principaux chapitres en plus de cette introduction générale et la conclusion finale. Le premier chapitre présente un aperçu de la genèse du monde des réseaux sans fil en commençant par un bref historique soulignant les mécanismes majeurs jouant un rôle dans l'évolution de ce domaine. Le chapitre aborde par la suite l'architecture et la classification des réseaux sans fil récents pour nous permettre de bien situer nos deux réseaux (Wifi et WiMAX) par rapport aux autres réseaux sans fil. Finalement le chapitre introduit les principales tendances dans ce domaine tout en soulignant l'émergence de la notion de convergence et les futures architectures composites basées sur la collaboration de diverses technologies.

Le deuxième chapitre résume le fonctionnement des réseaux Wifi et WiMAX et les mécanismes de gestion de la QoS au sein des deux technologies. Le chapitre commence par un aperçu du réseau Wifi avec ses différents modes de fonctionnement pour présenter ensuite un résumé des méthodes clés utilisées pour garantir la QoS des connexions aux utilisateurs Wifi. En deuxième lieu le chapitre donne un aperçu des principes de fonctionnement des réseaux WiMAX et ses différentes sous-couches en mettant l'accent sur la richesse de ses fonctionnalités de gestion de la QoS déjà intégrées dans le WiMAX.

Le troisième chapitre détaille la solution proposée dans ce travail en commençant par une vue générale de la convergence Wifi-WiMAX suivie par l'introduction du cadre de travail choisi. Ce dernier va nous permettre d'illustrer la stratégie proposée pour le couplage de la gestion de la QoS des réseaux Wifi et WiMAX. Le chapitre liste ensuite les principales raisons pour lesquelles NS3 a été choisi pour la réalisation des diverses simulations. Finalement les résultats sont présentés et interprétés pour bien évaluer l'impacte de l'introduction de notre stratégie de gestion.

Le manuscrit est clôturé par une conclusion générale résumant les idées fondamentales que nous a apportées ce travail tout en discutant des perspectives futures ouvertes pour la suite.

I GENERALITES SUR LES TECHNOLOGIES RESEAUX SANS FIL

I.1 INTRODUCTION

Les réseaux sans fil ont été introduits dans le but de se substituer aux réseaux filaires à l'intérieur des bâtiments et de fournir un accès radio du type Ethernet en offrant des gammes de service avec des débits inférieurs, mais avec l'avantage d'une mobilité supplémentaire même si elle est faible par rapport à celle de réseaux cellulaires mobiles. Cet objectif initial a été étendu à un accès sans fil large bande et une connectivité aux réseaux IP mais a également donné naissance à d'autres types de réseaux sans fil qu'on peut désigner sous l'acronyme WxAN, qui se distinguent par la nature des services offerts et donc des débits, des bandes de fréquence et qui sont conçus pour être adaptés au mieux à leur environnement.

Un réseau sans fil est un réseau dans lequel au moins deux terminaux sont capables de communiquer entre eux grâce à des signaux radioélectriques. Les réseaux sans fil ne sont pas tout récents, mais avec le développement de l'informatique et des systèmes d'information, la technologie est venue répondre au besoin primaire de l'homme : la mobilité et la facilité d'utilisation de déploiement et d'extension. Ces réseaux dits sans fil sont de plusieurs sortes : WiFi, Bluetooth, BLR, UMTS...

Dans ce chapitre nous allons introduire quelques notions générales sur les réseaux sans fil en commençant par un petit historique suivi par un aperçu sur l'architecture et la classification des réseaux sans fil. La dernière partie du chapitre parlera des tendances actuelles en matière de gestion de qualité de service et l'évolution des réseaux sans fil, ceci nous amènera ainsi à justifier la motivation de cette thèse.

I.2 HISTORIQUE

I.2.1 LES ORIGINES[1]

Une meilleure étude des réseaux nouvelle génération NGN sans fil tel que le Wifi ou le WIMAX devrait passer par l'exploration et la compréhension de l'histoire de ces derniers. Ceci remonte aussi loin que 1885 avec les travaux de Heinrich Hertz à qui, les premières transmissions radio sont attribuées. Le travail d'Hertz a réveillé beaucoup d'intérêt scientifique et l'émergence du sens des affaires des scientifiques a déplacé ces découvertes vers le domaine commercial.

Un certain scientifique italien Guglielmo Marconi est considéré comme le père des systèmes de radio mobiles d'aujourd'hui puisqu'il a pu réaliser des transmissions de quelques kilomètres entre un émetteur et un récepteur. Marconi se voit accepter le soutien et le sponsoring du bureau de poste du Royaume-Uni pour appliquer son travail dans la marine marchande.

La Première guerre mondiale a été un facteur déclenchant pour démontrer la valeur de la radio mobile dans des manœuvres militaires en ce qui concerne la taille des stations radios. Après la guerre, le développement de la technologie radio était centré principalement autour de la diffusion audio (chaines radios). L'augmentation rapide du nombre de station radio aux USA a abouti à une commercialisation massive des récepteurs radios. Ceci a également généré des efforts pour la coordination de l'utilisation du spectre radio. La première conférence internationale pour la gestion du spectre radio a eu lieu à Washington en 1927. C'est le début de ce qu'on appelle « Standardisation ».

D'autres applications ont été expérimentées comme la Radio Mobile Privée (RMP) où en 1921, la police de Detroit l'a utilisé dans la transmission unidirectionnelle de la voix de la centrale vers les voitures. La seconde guerre mondiale a conduit à la fabrication en série d'équipements de radio mobile pour équiper le nombre croissant d'avions et des bateaux militaires. Après la guerre, les constructeurs ont commencé à viser le marché du RMP où des téléphones mobiles ont été installés dans des taxis à partir de 1950.

Par la suite, le développement du transistor a radicalement réduit la taille et la consommation électrique des systèmes radio. Ainsi en 1965, le premier téléphone mobile en format de poche a été produit. A ce moment, le déploiement du RMP a commencé à dépendre plus de la manière d'attribution des licences et des politiques régulatrices des gouvernements au lieu de dépendre de l'équipement et de l'accueil du marché.

Il faut noter qu'il y a une mince ligne de démarcation entre le RMP et ce que l'on connaît aujourd'hui comme la radio mobile (typiquement système radio cellulaire). Le premier service de téléphone mobile a été présenté par AT&T (American Télécommunication and Telegraphy) dans 25 villes américaines en 1946 et appelé le Service Téléphonique Mobile (STM). Ce n'était pas un système cellulaire parce que des cellules singulières sont utilisées uniquement.

En Suède, le premier système de radio mobile européen a été introduit en 1955 par Televerket alors qu'en Angleterre le premier système commercial appelé système-1 a été introduit à Londres en 1965. La variante suivante, système-3, a réduit la largeur de bande du canal vocal de 100 kHz à 25 kHz en augmentant ainsi la capacité. Une grande partie des travaux sur les systèmes cellulaires est originaire des États Unis. Le concept cellulaire a été d'abord développé par Bell Labs en 1948 et sa société mère, AT&T. En 1977, ceci a conduit vers l'affectation de la bande 800 MHz.

Des essais basés sur cette licence ont eu lieu jusqu'en 1981 et ont convaincu l'organisme de réglementation Américain FCC (Federal communications Commission) que le système cellulaire est un concept viable. Le déploiement cellulaire aux USA a été basé sur l'AMPS (Advanced Mobile Phone Service) ; un standard qui définit seulement l'interface aérienne. La plupart des opérateurs utilisaient des approches différentes pour la commutation et la facturation rendant ainsi le roaming impossible.

Dans les pays nordiques, le Téléphone Mobile à 450 MHz (NMT450) a été développé avec la possibilité d'effectuer un roaming avec d'autres pays scandinaves. Le NMT900 a été conçu par la suite. D'autres pays européens ont adopté divers systèmes dont certains ne sont développés et utilisés que pour ces pays alors que d'autres ont adopté une version modifiée de l'AMPS connu sous le nom de Total Access Communications System (TACS) qui fonctionne dans la bande des 900 MHz. En Europe, cette étape de développement de la radio mobile entre 1985 à 1991 a été marquée par une planification monopoliste, la plupart des pays ayant seulement fourni une licence pour l'Organisme de Poste et de Télécommunication (OPT). Le Royaume-Uni a fait le premier pas pour introduire la concurrence en publiant deux licences et empêchant l'OPT de posséder les deux. La concurrence est maintenant reconnue comme importante dans la planification des services radio mobiles.

Après 1991, la radio mobile a été dominée par le système GSM dont la standardisation a commencé en 1982 au sein du CEPT (Conférence Européenne des Administrations des Postes et Télécommunications). Typiquement, les standardisations au sein du CEPT ont été menées par des ingénieurs ayant peu de considération pour la réalité commerciale et un désir de voir leurs propres idées incorporées dans le standard d'où l'échec de plusieurs de ces derniers (ex. TETRA, DECT...). Les raisons du succès du standard GSM sont variées ; un point clé étant le fait que cette standardisation a commencé tôt avant que presque nul constructeur n'ait commencé à développer son propre standard. Le fait de l'absence d'un standard concurrent majeur et plus particulièrement du côté de l'USA a également contribué au succès du GSM. Un autre point clé est l'inclusion des constructeurs dans le processus de la standardisation du GSM, chose qui n'a pas été faite pour les standards précédents.

La standardisation du GSM s'est vue plus complexe que prévu et a duré 13 ans depuis son introduction en 1982 jusqu'au dépôt final en 1995. Les organismes de standardisation ne savent pas générer des standards simples. En effet, avec les multiples parties participants à la standardisation, les exigences de chaque partie tendent à être ajoutées aux spécifications finales. Les standards complexes peuvent être avantageux lorsqu'il y a un grand marché mais peut souvent causer l'échec du standard dans le cas contraire. En effet, le GSM a bénéficié d'un grand marché principalement du fait qu'il était mandaté (exigé) par la commission Européenne (choix obligatoire pour les opérateurs européens).

Les opérateurs des autres pays ont choisi initialement le GSM à cause de la fourniture compétitive des équipements, une large base d'expertise dans le déploiement du réseau et la possibilité du roaming dans d'autres pays.

Par la suite, un des points clés du débat dans la technologie radio était le choix du schéma d'accès approprié CDMA (code division multiple access) ou TDMA (time division multiple access). Ce débat a été en fait moins concerné par le choix d'un schéma d'accès « idéal » qu'il était par le choix entre un standard européen (GSM) et un standard américain (CdmaOne développé par la société Qualcomm). Clairement, Qualcomm n'aurait pas réussi en Europe où le standard européen a été mandaté. La clé du succès de Qualcomm était son lien avec d'autres constructeurs et le désir des États-Unis d'avoir un produit national plutôt que d'importer un produit européen. D'autres pays comme la Corée du Sud se sont aussi rebellés contre la dominance européenne et ont exigé de leurs opérateurs de choisir le cdmaOne avec des équipements fabriqués localement. Ainsi, la politique et le nationalisme sont susceptibles de jouer un rôle clé dans le développement des standards radio mobile.

1.2.2 HISTORIQUE RECENT[2]

Autour de 1998, les monopoles de télécommunication sont arrivés à leur fin dans beaucoup de pays Européens, ainsi les prix ont baissé significativement dès les premières semaines et mois de l'entrée en vigueur du nouveau règlement. Pendant ce temps, La technologie modem de ligne fixe avec des vitesses de 30-56 kbit/s s'était développée et a été lentement adoptés par les étudiants et par d'autres utilisateurs d'ordinateur pour l'Accès à Internet. La communication à base de texte a aussi commencé à se développer ainsi que l'apparition des navigateurs Internet qui permettait l'affichage des pages Web à contenance graphique. Tandis que les communications vocales sur des réseaux mobiles sont devenues rapidement un succès, l'accès mobile à Internet était toujours à ses débuts. À l'époque, les réseaux GSM permettait des débits de 9.6 à 14.4 kbit/s sur des connexions à commutation de circuit et peu de personnes se sont servies de données mobiles, cependant, principalement à cause des prix très élevés et du manque d'applications et des dispositifs. Néanmoins, la fin de la décennie a vu les premières applications de données mobiles comme des navigateurs Internet et le courrier électronique mobile sur des dispositifs comme des Assistants Numériques Personnels (PDAs), qui pouvait communiquer avec des téléphones mobiles via un port infrarouge.

Une des percées majeures durant la période 2000 et 2005 était la hausse d'Accès à Internet via des Lignes d'Abonné Numériques (DSL) qui pouvait atteindre des vitesses de 1 MBIT/S et même plus. Comparé avec les 56 kbit/s via des connexions du modem analogiques, les temps de téléchargement pour les pages Web avec un contenu graphique et des grands fichiers se sont améliorés significativement. À la fin de cette période, la majorité des gens dans beaucoup de pays avait accès à l'Internet haut débit qui leur a permis de voir de nouvelles manières de communication comme les Blogs et les Wikis. Soudainement, les utilisateurs n'étaient plus seulement des consommateurs de contenu, mais pouvaient aussi être des créateurs pour un auditoire mondial. C'est le Web 2.0.

Dans le monde de la téléphonie de ligne fixe, les prix des appels nationaux et internationaux ont continué à baisser. Vers la fin de cette période, il y a eu la découverte de l'utilisation de la téléphonie Internet pour effectuer des appels téléphoniques sur Internet via les lignes DSL. Des programmes propriétaires comme Skype ont soudainement permis aux utilisateurs d'appeler n'importe quel abonné Skype dans le monde. De plus, beaucoup de start-ups ont commencé à offrir des convertisseurs téléphoniques analogique-Internet qui utilisent le protocole standardisé SIP (Session Initiation Protocol) pour transporter de la voix téléphoniques sur Internet.

Au début 2001, le GPRS (Général Paquet Radio service) a été introduit dans les réseaux GSM publics pour la première fois. Comme les premiers téléphones mobiles GPRS ont rapidement suivi, l'accès mobile à Internet est devenu pratiquement faisable pour un large public. Cependant, des problèmes de débit, du temps d'établissement des appels et la nécessité de maintenir le canal même durant le temps d'inactivité n'étaient pas adéquats pour la plupart des applications Internet, d'où le non-succès du WAP 1.0.

Vers 2005, les dispositifs mobiles se sont développés avec de haute résolution d'affichage, des navigateur Internet mobiles WAP 2.0 et la facilité d'utilisation du courrier électronique mobiles clients, tout ceci en association avec GPRS comme une couche de transport à commutation de paquet a finalement permis à l'accès mobile Internet de faire le pas vers le marché de masse.

Un autre évènement important pour l'accès à Internet sans fil pendant cette période était l'activation des réseaux 3G dans beaucoup de pays en 2004 et 2005. Tandis que le GPRS s'est rapproché des vitesses du modem analogique, UMTS apporte en pratique un débit de plus de 384 kbit/s et l'expérience est devenue ainsi semblable au DSL. Cependant, la facturation des opérateurs de réseau ont freiné son adoption vis à vis du marché de masse pendant plusieurs années.

De 2005 à aujourd'hui, le pourcentage des gens dans des pays industrialisés ayant accès à Internet haut débit via DSL a continué à augmenter. De plus, beaucoup d'opérateurs de réseau ont commencé à déployer l'ADSL2+, ainsi les vitesses de téléchargement des nouveaux modems dépassent les 15 Mbit/s pour des utilisateurs vivant à côté des centres de commutation. VDSL et les déploiements domestiques par la fibre optique offrent des débits encore plus hauts. Une autre tendance qui a accéléré depuis 2005 est la Voix sur IP (VoIP) via un port téléphonique DSL.

Autour de 2007, le nombre d'utilisateurs de téléphone mobile a atteint les 3 milliards et les opérateurs de réseaux ont enregistré 1000 nouveaux utilisateurs par minute. La grande majorité de cette croissance a été conduite par le déploiement des réseaux GSM/GPRS dans des marchés émergents. Dans seulement quelques années, Les réseaux mobiles ont pratiquement changé le mode de travail et l'accès aux informations pour les petits entrepreneurs. Pour un historique plus détaillé voir [3]

I.3 ARCHITECTURE DES RESEAUX SANS FIL

Un réseau de télécommunication typique, historiquement concentré sur le traitement de la voix et des communications vocales est présenté dans la Figure 1.1. Le modèle réseau de transmission de données est une infrastructure symétrique qui connecte des CPEs (source et destination d'informations) et consiste en quatre systèmes majeurs: accès, transmission, commutation et la signalisation.

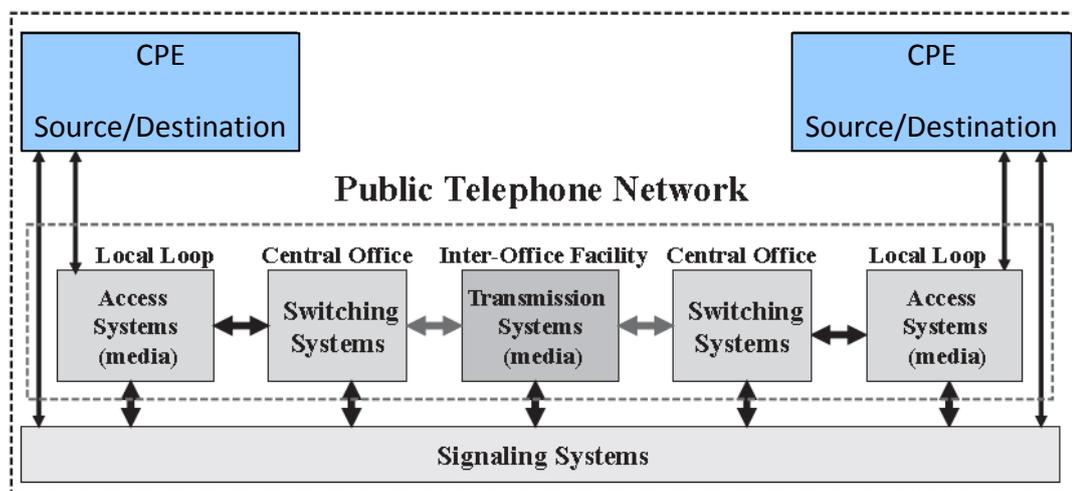


Figure I.1 : Architecture d'un réseau de télécommunication standard

Un réseau de transmission de données typique est présenté dans la Figure 1.2. Il consiste en quatre systèmes majeurs : conversion, accès, transmission et systèmes de commande.

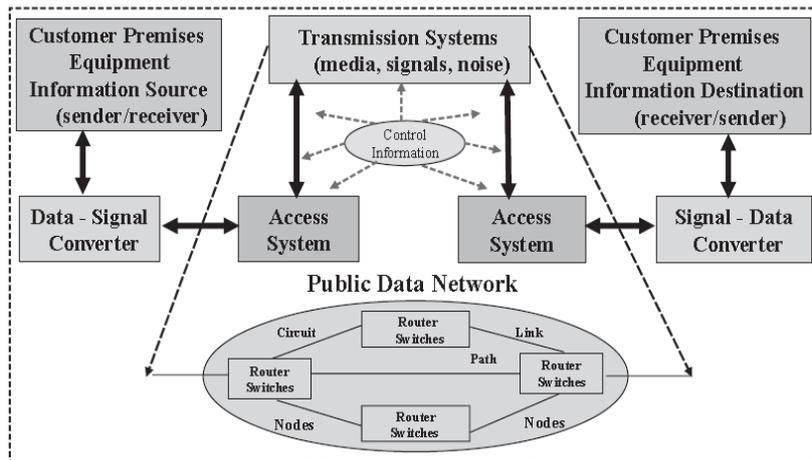


Figure I.2 : Architecture d'un réseau de télécommunication informatique

Il y a trois aspects majeurs impliqués dans les services de communications : interfaces de communication, protocoles de communication, piles de communication à couches. Les interfaces de communication sont des points de connexion entre des composants du réseau conçus selon des spécifications propriétaires ou standardisées. Une interface est définie selon les caractéristiques mécaniques, électriques et fonctionnelles qui permettent à la communication de se réaliser entre les dispositifs du réseau adjacents. Les protocoles de communication sont des descriptions formelles du format d'unité des données et des règles de transmission pour l'échange des messages entre les entités du réseau. Les protocoles de communication sont organisés dans des piles de communication à couches multiples où chaque couche fournit des services à la couche au-dessus. L'exemple d'architecture de communication à couche le plus commun serait l'OSI (Open System Interconnexion) standardisé par l'organisation internationale de normalisation ISO (International Standardisation Organisation) et est constitué de 7 couches.

Dans cette architecture, la communication entre deux applications a lieu par des piles de protocole à pair identique comme indiqué dans la Fig. 1.3.

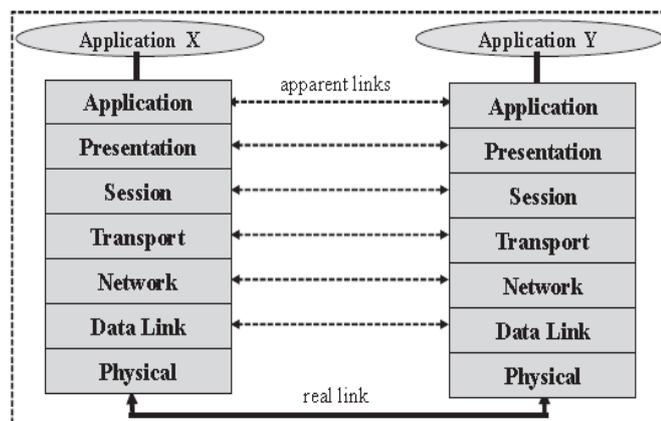


Figure I.3 : Les différentes couches réseaux

Les fonctions et les services OSI peuvent être résumés comme suit:

Couche OSI	Les services et les fonctions principales
Application	Fournit des processus d'application pour avoir accès à l'environnement de couches OSI.
Présentation	Fournit la structure du format de données et la représentation selon une syntaxe acceptée.
Session	Fournit l'établissement et le contrôle du dialogue entre les applications dans les systèmes finaux.
Transport	le mécanisme de bout à bout fiable pour l'échange d'informations entre des processus ou systèmes.
Réseau	Adressage, cheminement et relayage d'informations transmises à travers le réseau.
Liaison de donnée	Contrôle de flux du transfert de données, les mécanismes de détection et de correction d'erreur et récupération. des données.
Physique	Interfaces fonctionnelles, électriques, mécaniques et procédurales du média de communication.

Le format de base des informations échangées, qui permet aux parties communicantes de comprendre et interpréter les informations transmises est le PDU (Protocol Data Unit). Le noyau de n'importe quel PDU est l'actuelle information de l'utilisateur. Ce noyau est augmenté avec une succession de champs de données distincts spécifiques à chaque couche pour exécuter des fonctions et des services spécifiques telles que la synchronisation, l'adressage, le séquençement, contrôle de flux et la détection d'erreur. La construction des PDUs d'OSI couche par couche est donnée dans la Figure suivante:

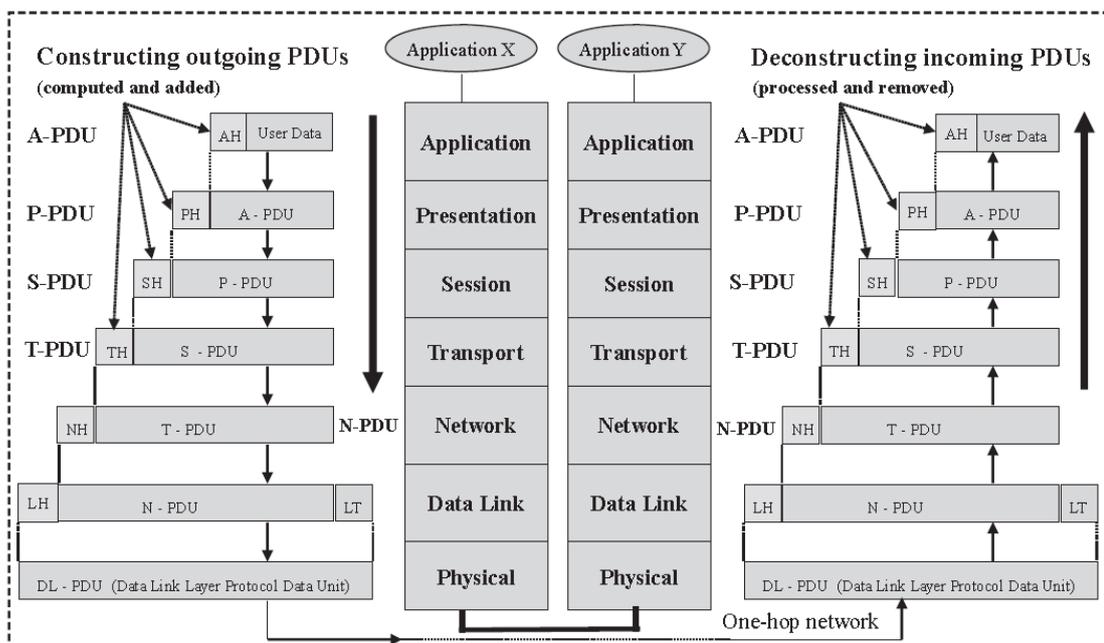


Figure I.4 : Construction des PDUs dans l'architecture OSI

Dans l'échange de message entre deux systèmes finaux représentés par l'application X (l'émetteur) et l'application Y (le récepteur), les données de l'utilisateur sont encapsulées par l'entête d'application (AH) pour créer le PDU d'application (A-PDU). De la même façon, l'opération est répétée pour les couches inférieures.

De l'autre côté de la liaison, le processus inverse se réalise en décapsulant l'entête du message au fur et à mesure de son passage par les couches supérieurs successives. Les procédures de transmission sont utilisées pour fournir la connectivité entre les dispositifs du réseau qui inclut trois phases distinctes, comme indiqué dans la Figure 1.5.

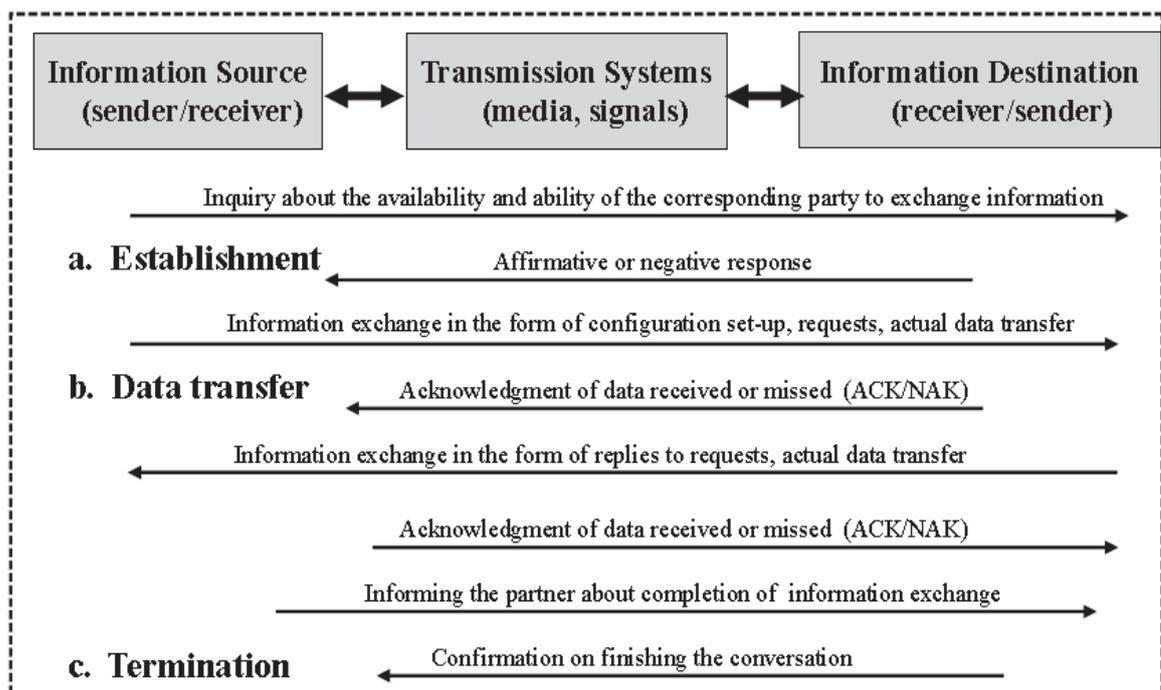


Figure I.5 : Procédures de transmission

Selon la procédure adoptée pour la connectivité, nous pouvons avoir deux types de services de communication, comme indiqué dans la Figure suivante:

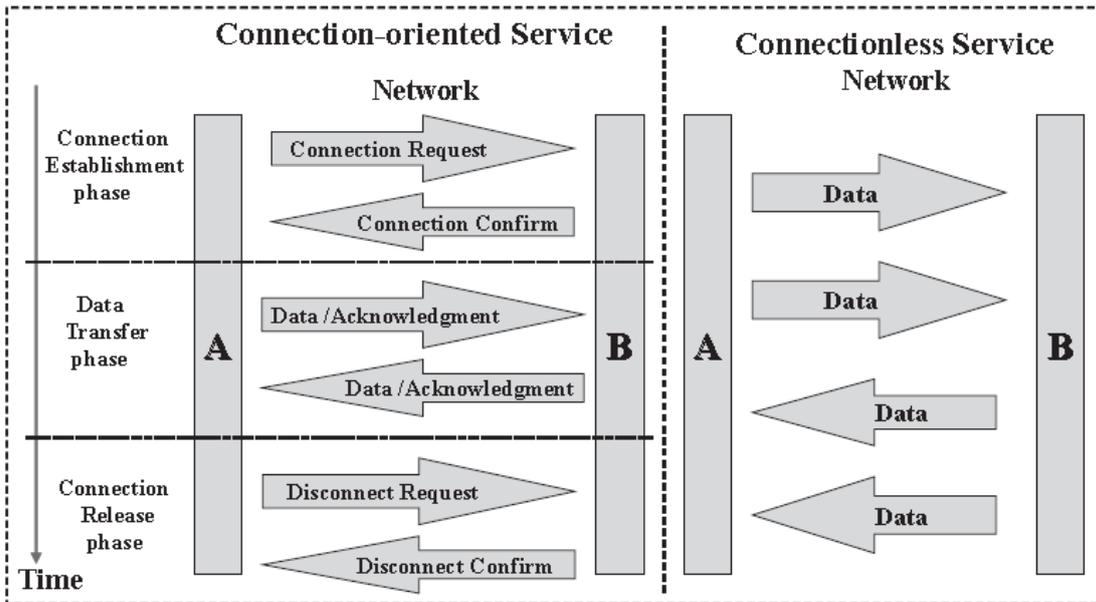


Figure I.6 : Types de connexion.

Le premier type de transmission fournit une assurance concernant la capacité des deux parties de la communication à échanger les informations avec un contrôle complet de tous les aspects du transfert des données, exemple TCP (Internet Transmission Control Protocol). Le deuxième type de transmission ne donne à aucune des deux parties la garantie d'être capable d'échanger les informations, exemple UDP (Internet User Datagram Protocol).

Un modèle de haut niveau de communication sans fil est très semblable avec le modèle général de communication, comme indiqué dans la Figure 1.7.

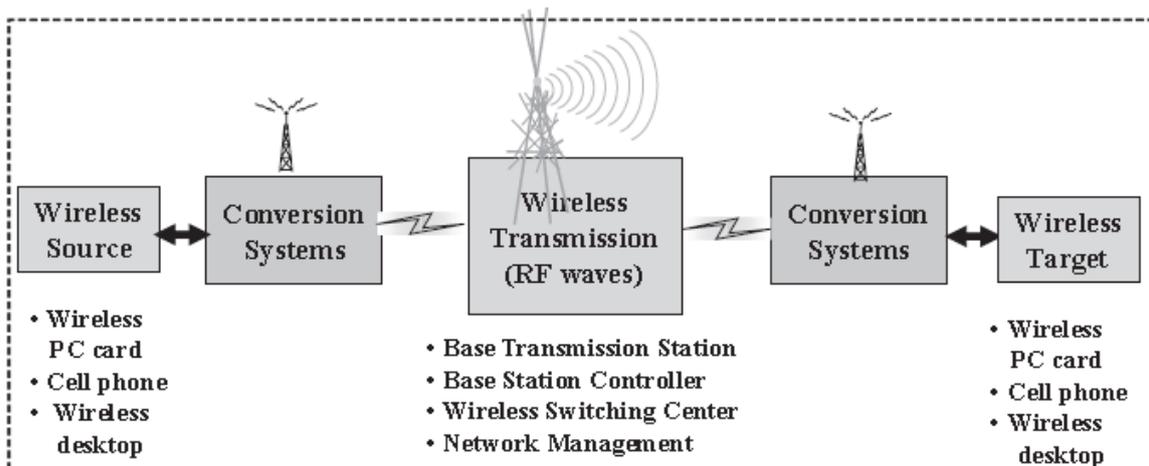


Figure I.7 : Modèle de communication sans fil

Les quatre moyens principaux utilisés pour la communication sans fil sont: à base de radio, à micro-ondes, à base de satellite et à optique infrarouge dans l'espace libre. Chaque ensemble de l'architecture réseau sans fil est pris en charge par une variété de technologies de réseaux concurrents. Chacune de ces technologies est soutenue par un ensemble de protocoles de communication et de spécifications standardisées. Des exemples de technologies des différentes classes réseaux sans fil seront donnés dans la section suivante.

I.4 CLASSIFICATION DES RESEAUX SANS FIL

La grande variété de réseaux sans fil peut être classifiée selon trois critères majeurs : la couverture géographique, le niveau de mobilité et l'assignation de spectre.

Selon la zone de couverture, il y a cinq types principaux de réseaux sans fil comme suit :

Wireless Wide Area Networks (WWAN): Ces réseaux fournissent une connectivité réseau directe à de grands secteurs y compris les sites isolés et distants.

Wireless Metropolitan Area Networks (WMAN): Ces réseaux fournissent un accès large bande direct et une connectivité réseau sans fil dans les zones métropolitaines.

Wireless Local Area Networks (WLAN): Ces réseaux fournissent un accès local sans fil pour le transfert des données et de la voix aux ressources partagées telles que les serveurs, les imprimantes, les routeurs dans une zone limitée d'une organisation ou d'une entreprise.

Wireless Personal Area Networks (WPAN): Ces réseaux fournissent une connectivité sans fil à divers dispositifs et appareils dans la zone limitée d'une résidence. Aussi connu sous le nom de réseau domestique

Near-field, Sensor-based Networks: Ces réseaux fournissent des communications à champ rapproché en utilisant l'identification par fréquence radio et la transmission ultra-large bande.

Compte tenu de la position relative des composants de réseau, les réseaux sans fil peuvent également être divisé en: sans fil fixe et sans fil mobile. Dans les réseaux sans fil fixe: les composants du réseau maintiennent une position fixe. La plupart des technologies associées avec WLAN, WMAN, WPAN, et les réseaux de capteur entrent dans cette catégorie.

Les réseaux sans fil mobiles sont utilisés pour toutes les technologies mobiles cellulaires et dans certaines instances du WLAN et WMAN (Par exemple, WLAN nomade et WiMAX mobile). La figure 1.9 illustre une variété des systèmes de communications sans fil organisés suivant la couverture géographique.

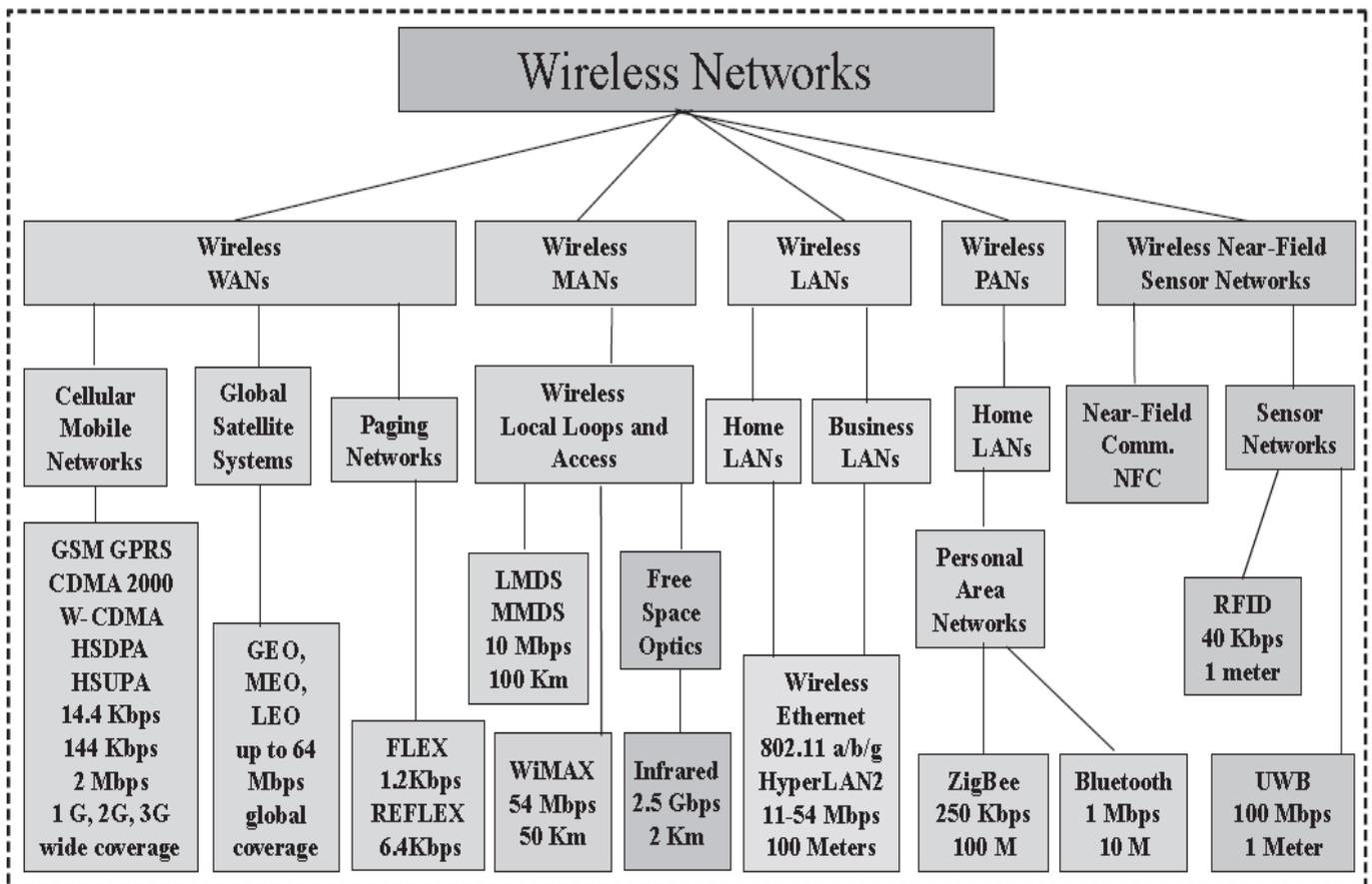


Figure I.8 : Aperçue de différentes classes de réseau sans fil

Compte tenu de la multitude des technologies sans fil, la nécessité d'éviter les interférences entre des groupes d'utilisateurs (militaires, commerciales, amateurs) et d'imposer une discipline pour l'utilisation des standards communs, une stricte répartition des fréquences pour différentes applications est nécessaire. Cette répartition peut être considérée comme une classification spectrale.

La correspondance entre les fréquences de transmission et la longueur d'onde avec également le terme des références couramment utilisé associés à différents types d'ondes, sont donnés dans le tableau suivant.

Désignation internationale	Désignation francophone	Fréquence	Longueur d'onde	Exemples d'utilisation
ELF (extremely low frequency)	EBF (extrêmement basse fréquence)	3 Hz à 30 Hz	100 000 km à 10 000 km	Détection de phénomènes naturels
SLF (super low frequency)	SBF (super basse fréquence)	30 Hz à 300 Hz	10 000 km à 1 000 km	Communication avec les sous-marins
ULF (ultra low frequency)	UBF (ultra basse fréquence)	300 Hz à 3 000 Hz	1 000 km à 100 km	Appareil de recherche de victimes d'avalanche
VLF (very low frequency)	TBF (très basse fréquence)	3 kHz à 30 kHz	100 km à 10 km	Communication avec les sous-marins, Implants médicaux, Recherches scientifiques...
LF (low frequency)	BF (basse fréquence)	30 kHz à 300 kHz	10 km à 1 km	Radionavigation, Radiodiffusion GO, Radio-identification
MF (medium frequency)	MF (moyenne fréquence)	300 kHz à 3 MHz	1 km à 100 m	Radio AM
HF (high frequency)	HF (haute fréquence)	3 MHz à 30 MHz	100 m à 10 m	Communication pour les vols long courrier, Radio-identification...
VHF (very high frequency)	THF (très haute fréquence)	30 MHz à 300 MHz	10 m à 1 m	Radio FM, Télévision
UHF (ultra high frequency)	UHF (ultra haute fréquence)	300 MHz à 3 GHz	1 m à 10 cm	GSM, GPS, Wi-Fi, Télévision
SHF (super high frequency)	SHF (super haute fréquence)	3 GHz à 30 GHz	10 cm à 1 cm	Micro-onde
EHF (extremely high frequency)	EHF (extrêmement haute fréquence)	30 GHz à 300 GHz	1 cm à 1 mm	Radars anticollision pour automobiles, Liaisons vidéo transportables
Terahertz	Téraherz	300 GHz à 3 000 GHz	1 mm à 100 µm	

Tableau I-1 : Classification d'onde d'émission radio

1.5 LES TENDANCES ACTUELLES ET LEUR EVOLUTION

1.5.1 DEVELOPPEMENT RECENT[4]

Dans les pays industrialisés, les réseaux de la troisième génération ont continué à se développer et l'année 2006 a vu les premières améliorations du réseau UMTS à HSDPA (High Speed Data Packet Access). Dans une première étape, ceci permettait à l'utilisateur d'atteindre des vitesses entre 1 et 3 Mbit/s. Dans le court terme, il est probable que le HSDPA sera aussi très avantageux pour des applications mobiles, les téléchargements de musique et de la vidéo streaming sur des dispositifs mobiles qui deviendront des applications grand public. Tandis que les réseaux 3G ont été disponibles pour quelque temps, leur démarrage était lent jusqu'à 2006/2007 lorsque des opérateurs de réseaux mobiles ont finalement proposé des plans de prix intéressants.

En 2006, le revenu des USA concernant seulement les données mobiles a atteint 15.7 milliards de dollar. Tandis que le roaming de données sans fil est toujours à ses débuts, on offre déjà l'accès à Internet sans fil via des cartes SIM prépayées dans beaucoup de pays aux prix semblables à ceux pour des clients avec un abonnement. C'est un autre pas important puisque il ouvre la porte à l'accès à Internet n'importe quand et n'importe où.

Dans l'avenir, la question principale que devrait se poser les opérateurs et les fournisseurs de technologie réseau est quand et pourquoi il serait nécessaire d'aller au-delà des réseaux 3G sans fil? Aujourd'hui, la capacité des réseaux 3G, 3.5G est encore suffisante en ce qui concerne les exigences de la bande passante pour différentes applications (e-mail, téléchargement mobile, les blogs, messages instantanés mobiles...etc.) ainsi que pour le nombre d'utilisateurs.

En revanche, il y a certaines tendances qui sont déjà visibles et qui vont augmenter les exigences de la bande passante dans le futur. Parmi ces tendances:

- Le nombre croissant d'utilisateur : à cause de la chute des prix, plus de personnes utiliseront les applications mobiles qui exigent l'accès au réseau.
- Contenu multimédia: une image augmente la quantité des données qui doivent être transférées pour une page Web. La Vidéo et les téléchargements de musique deviennent aussi plus populaires ce qui augmente aussi les exigences en bande passante.
- Réseaux mobiles sociaux: Comme les utilisateurs ne consomment plus seulement le contenu, mais utilisent le réseau pour partager leurs propres idées, images et vidéos avec des autres personnes. Des applications comme, par exemple, les réseaux sociaux tels que Facebook, le CHAT à travers MSN, SKYPE et Lifeblog permettent aux utilisateurs de faire des uploads d'images, vidéos et blogs du mobile vers le Web.
- La voix sur IP: La migration des réseaux sans fils vers la VoIP est beaucoup plus lente que dans les réseaux filaires, particulièrement en raison des exigences de la bande passante qui doit être plus élevée.
- Remplacement de la ligne fixe d'Internet: Dans beaucoup de pays, des opérateurs sans fil essayent de garder ou d'augmenter le revenu moyen par utilisateur en offrant l'accès à Internet pour des PCs, des PDA et des dispositifs mobiles via le réseau UMTS/HSDPA ou des réseaux CDMA. Ainsi, ils ont commencé à rivaliser directement avec les opérateurs DSL. De même, cela exige un ordre d'ampleur de largeur de bande supplémentaire.

Un certain nombre de technologies sans fil sont actuellement en voie de développement ou dans la première phase de déploiement, et sont conçues pour satisfaire ces futures demandes : 3GPP Évolution À long terme (LTE), HSPA + et WiMAX. De plus, le WiFi Mesh va aussi probablement être une technologie de réseau importante qui est exigée pour respecter les demandes de capacité futures. La question qui surgit dans ce contexte est: lesquelles de ces technologies sont de type 3G et qui seront appelées 4G dans le futur?

L'organisme responsable pour la catégorisation des réseaux sans fil est l'ITU l'International Télécommunication Union. Ce dernier classe les réseaux de Télécommunication mobile (IMT) comme suit :

- ✓ Les systèmes IMT-2000 - c'est ce que nous connaissons comme systèmes 3G aujourd'hui, par exemple UMTS et cdma2000. On donne la liste de tous les systèmes ITU-2000 dans ITU-R M.1457-6.
- ✓ Les systèmes Amélioré IMT-2000 – c'est l'évolution des systèmes IMT-2000, par exemple HSPA, CDMA 1xEvDo ainsi que les évolutions futures de ces systèmes.
- ✓ Les systèmes IMT-AVANCÉS - on considère que des systèmes dans cette catégorie sont des systèmes. 4G.

En ce moment, il n'y a toujours aucune définition claire en ce qui concerne les caractéristiques des futurs systèmes avancés 4G. Le ITU-R M 1645 de la recommandation donne de premières indications, mais laisse la porte grande ouverte. Il est prévu que la nouvelle et potentielle interface radio devra supporter un débit jusqu'à environ 100 Mbit/s pour la haute mobilité et jusqu'à environ 1 Gbit/s pour une basse mobilité. En comparant les spécifications du WiMAX à ces exigences potentielles, il devient clair que le WiMAX n'est pas qualifié comme un standard 4G IMT-AVANCÉS, puisque les taux de données sont beaucoup inférieurs, même dans des conditions idéales, LTE aura aussi des difficultés pour satisfaire ces conditions.

La recherche actuelle indique que la vitesse requise de transmission décrite dans le ITU-R M 1645 peut seulement être réalisée que dans une bande de fréquences de 100 MHz ou plus. C'est un vrai défi, tant d'un point de vue technique qu'aussi en raison d'un manque de spectre supplémentaire disponible.

En pratique, plusieurs technologies différentes de réseau coexisteront et se développeront dans l'avenir pour satisfaire les demandes naissantes en termes de largeur de bande et de capacité. Il est aussi probable qu'une combinaison de systèmes de radio différents, comme par exemple le WiMAX réuni avec le réseau local Sans fil Wifi, sera utilisé pour satisfaire des demandes de capacité.

D'un point de vue utilisateur ou service, cela n'a pas d'importance si on considère une technologie de réseau 3.5G, 3.9G ou 4G. Ainsi, les spécialistes réseaux utilisent le terme au-delà des systèmes 3G (Beyond 3G), qui inclut toutes les technologies qui pourront satisfaire les demandes de capacité futures.

1.5.2 EVOLUTION

Aujourd'hui, beaucoup de personnes utilisent déjà des points d'accès WiFi pour créer leur bulle d'Internet haut débit personnelle. Actuellement, les gens utilisent non seulement cette bulle avec des ordinateurs de bureau et des blocs-notes (PDA), mais aussi avec des dispositifs plus petits comme les téléphones portables avec les options WiFi incorporées.

Quand une personne quitte cette bulle personnelle à haut débit, les dispositifs mobiles commuteront vers le réseau cellulaire. Le déplacement entre la bulle Internet personnelle à domicile et le réseau cellulaire externe deviendra transparent lorsque les dispositifs et les services se développeront.

Un point clé dans ce développement est la convergence de réseaux sans fil hétérogènes qui sera étudié un peu plus loin dans ce manuscrit. Cette convergence est liée à un progrès constant dans le domaine de la recherche des réseaux sans fil, qui entraîne une accumulation importante de nouveaux résultats dont:

Dans de tels environnements, les antennes de transmission multiples peuvent être utilisées pour induire des fluctuations rapides de canal de sorte que l'aspect multiutilisateur puisse encore être exploité. Le concept peut être interprété comme un beamforming opportuniste et des gains de diversité peuvent être réalisés lorsqu'il y a suffisamment d'utilisateurs, même si le feedback du canal est très limité. En outre, dans un système cellulaire, le concept joue un rôle supplémentaire opportuniste de mise à zéro des interférences sur les utilisateurs des cellules adjacentes.

1.5.2.1 LES RESEAUX MESH

Dans un réseau sans fil avec de nombreuses paires source-destination, la transmission coopérative par des nœuds relais a le potentiel d'améliorer la performance globale du réseau.

Dans un réseau mesh/relais multisauts distribués (par exemple réseau ad hoc/capteurs sans fil, réseaux cellulaires multisauts), chaque nœud agit comme un nœud relais pour transmettre des paquets de données provenant des autres nœuds. Ces nœuds sont souvent limités en énergie et ont également un espace mémoire tampon limité. Par conséquent, des mécanismes efficaces d'économie d'énergie (par exemple les mécanismes du sommeil) sont nécessaires pour que la durée de vie de ces nœuds puisse être prolongée tout en satisfaisant les exigences en termes de qualité de service (QoS) (délai des paquets par exemple et le taux de perte de paquets) pour les paquets transmis.

I.5.2.2 LE CONTROLE DE LA TOPOLOGIE RESEAU

L'efficacité énergétique et la capacité du réseau sont peut-être les deux questions les plus importantes dans les réseaux sans fil ad hoc et les réseaux de capteurs. Des algorithmes de contrôle de topologie ont été proposés pour maintenir la connectivité du réseau tout en réduisant la consommation d'énergie et d'améliorer la capacité du réseau.

L'idée clé pour le contrôle de topologie est que, au lieu de transmettre une puissance maximale, les nœuds dans un réseau sans fil multi sauts en collaboration, déterminent leur puissance de transmission et définissent la topologie du réseau en formant leur propre relation de voisinage selon certains critères.

I.5.2.3 OPTIMISATION DU RESEAU

Les protocoles réseaux dans les architectures à couches ont toujours été obtenus sur une base de liaisons couche à couche indépendantes. Un bon nombre des dernières conceptions inter-couche (crosslayer) sont également menées par des approches fragmentaires. Alternativement, les piles de protocole réseau peuvent être systématiquement analysées et conçues comme des solutions distribuées à certains problèmes d'optimisation globale.

Les différents résultats de recherche se combinent pour donner un nouveau concept qui est le CRE (composite radio environment), ce dernier considère que différents réseaux radio peuvent être des composants en coopération dans une infrastructure d'accès sans fil hétérogène, à travers laquelle les fournisseurs de réseaux peuvent atteindre plus efficacement la capacité et la qualité de service (QoS) exigées.

La reconfigurabilité permet aux terminaux et aux éléments dynamiques du réseau de sélectionner et de s'adapter aux technologies d'accès radio les plus appropriées pour la gestion des conditions rencontrées dans certaines régions de la zone de service et certaines heures de la journée. De nos jours, une multitude de standards de technologies d'accès radio RAT (Radio Access Technology) sont utilisées dans les communications sans fil. Comme le montre la figure 1.9.

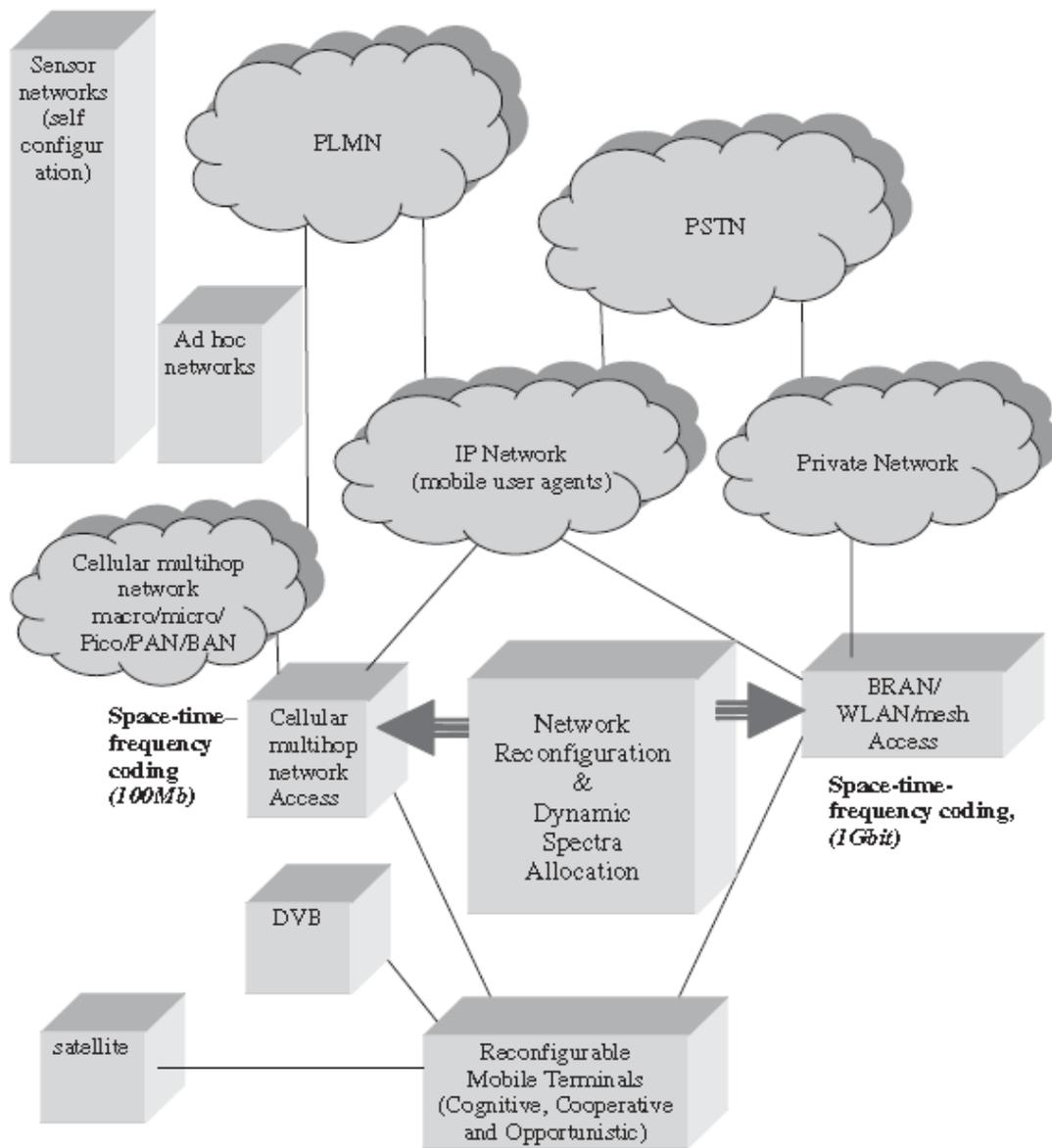


Figure I.9 : L'environnement CRE avec des multiple RATs.

Afin d'accroître encore l'efficacité spectrale, en plus du codage fréquentiel spatio-temporel dans la couche physique, De nouveaux paradigmes comme les solutions cognitives, de coopération et opportunistes seront utilisées. Bien que la technologie 4G soit ouverte pour les nouveaux schémas à accès multiples, le concept du CRE reste attractif pour accroître l'efficacité de prestation de services et les possibilités d'exploitation des RATs disponibles.

Un fournisseur de réseau (NP) peut posséder plusieurs composantes de l'infrastructure du CR (en d'autres termes, peut détenir des licences pour le déploiement et l'exploitation de différents RATs), et peut également coopérer avec les NPs affiliées. Dans tous les cas, un NP peut compter sur plusieurs réseaux et des technologies radio de remplacement, pour atteindre la capacité et les niveaux de QoS requises, de manière rentable.

Les utilisateurs sont dirigés vers les réseaux et les technologies radio les plus appropriées en fonction des critères des besoins du profil et la performance du réseau. Les différents RATs sont donc utilisés de manière complémentaire plutôt que concurrente. Le déploiement des systèmes CRE peut être facilité par le concept de reconfigurabilité du profil radio défini par logiciel.

La reconfigurabilité fournit des technologies essentielles qui permettent dynamiquement aux terminaux et éléments du réseau de sélectionner et de s'adapter à l'ensemble des RATs les plus appropriés. Selon le concept de reconfigurabilité, la sélection du RAT ne se limite pas à ceux qui sont préinstallés dans l'élément réseau. En fait, dans le cas de l'émergence d'un nouveau RAT, les composants logiciels nécessaires peuvent être dynamiquement téléchargé, installé et validé.

Les figures 1.10 et 1.11 illustrent l'architecture d'un terminal qui est capable de fonctionner dans un contexte CRE.

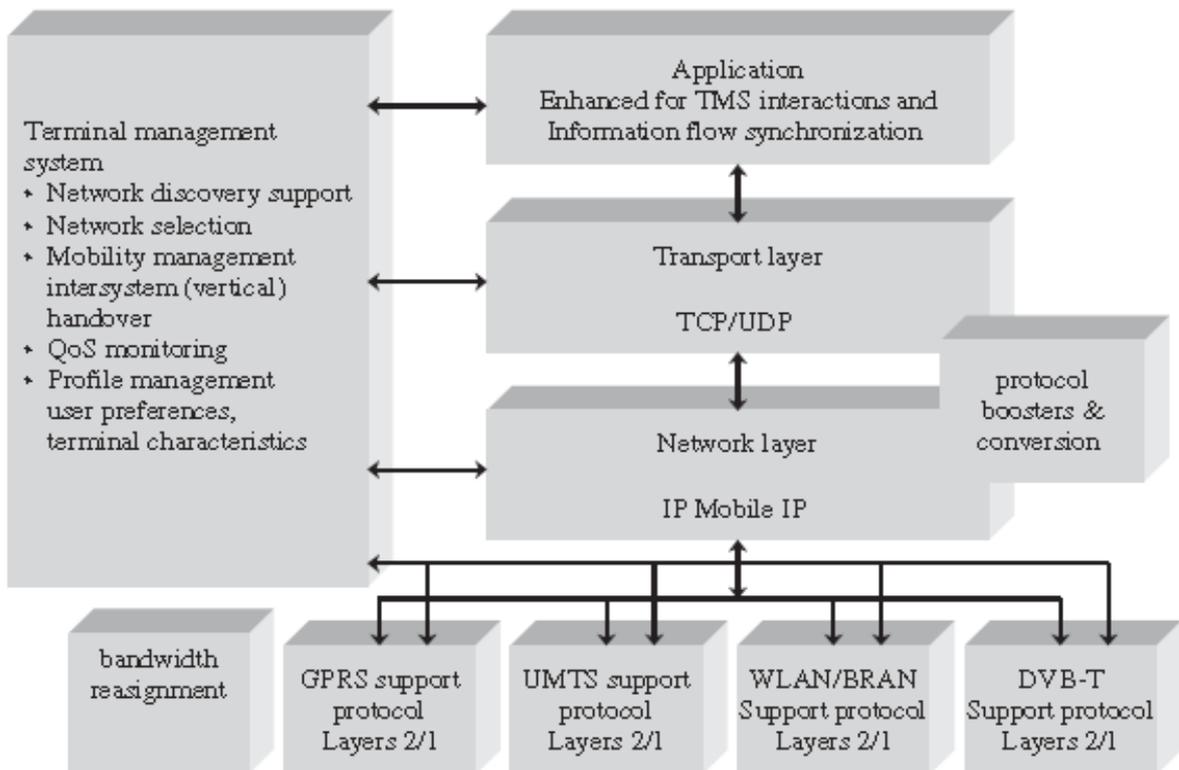


Figure I.10 : Architecture d'un terminal qui opère dans un environnement CRE.

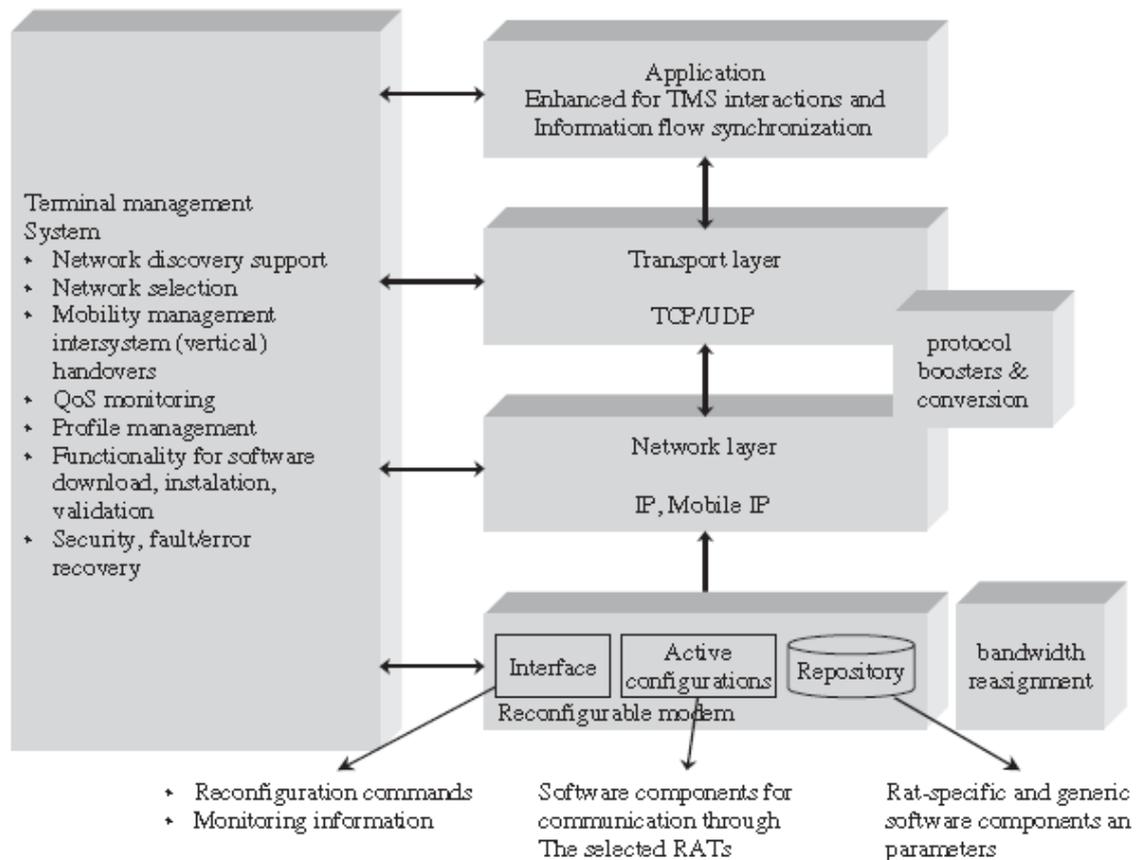


Figure I.11 : Architecture d'un terminal intégrant le concept de reconfigurabilité

La plupart des réseaux de communication sont soumis aux variations spatiotemporelles de la demande du flux, qui entraînent des variations dans le degré d'exploitation du spectre. Par conséquent, le spectre radioélectrique d'un service peut être sous-utilisé à certains moments et dans certaines zones géographiques, tandis qu'un autre service peut rencontrer un manque en même temps et lieu. Ces points fournissent la motivation pour un schéma d'allocation de spectre dynamique DSA (Dynamic Spectrum allocation), qui vise à gérer le spectre utilisé par un système radio convergent et le partager entre les réseaux radio membres du CRE dans l'espace et dans le temps afin d'augmenter l'efficacité spectrale globale, comme illustré dans les figures 1.12.

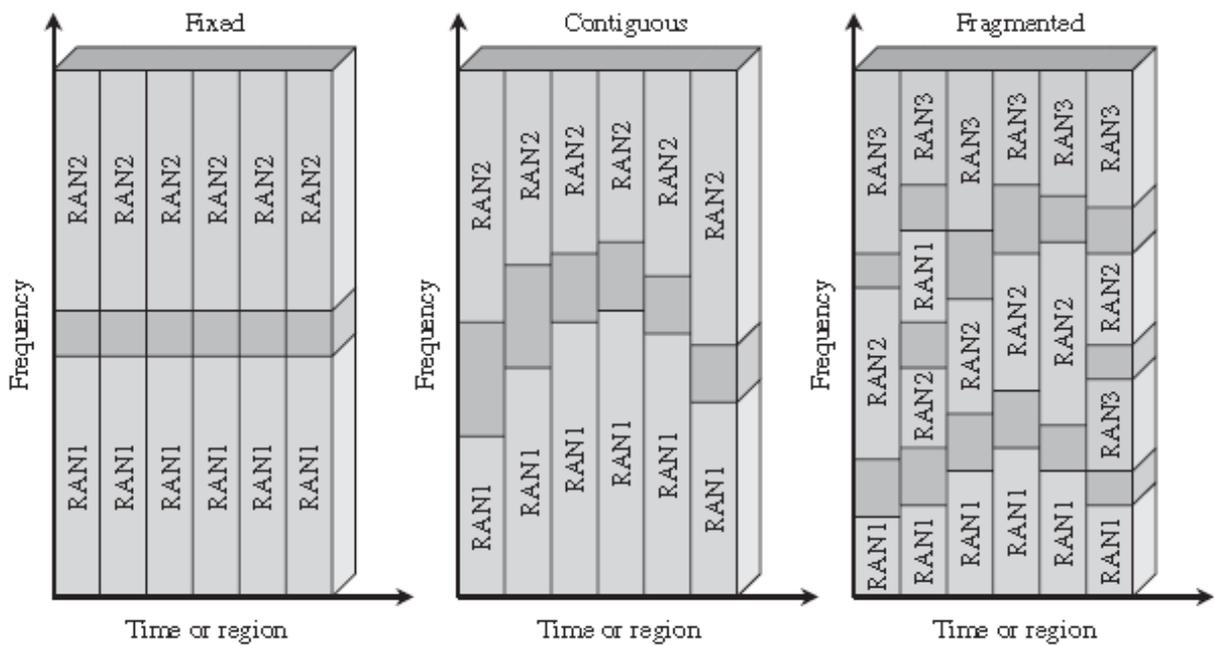


Figure I.12 : Les différents types d'allocation de spectre.

L'environnement CRE est géré par un système de gestion central appelé MS CRE (Management System CRE). D'une façon réactive, le MS-CRE est supposé réagir aux nouvelles conditions de la zone de service, tel que l'apparition inattendue d'un hot spot. De façon proactive, le système de gestion doit anticiper les changements dans la structure de la demande. Des parties de la MS-CRE devraient être capable de diriger les utilisateurs vers les réseaux plus appropriés de l'infrastructure du CRE. L'architecture se compose de trois entités logiques principales (Figure 1.13) :

- ✓ Surveillance, information de niveau de service et la redistribution des ressources(MSRB).
- ✓ Stratégies de gestion des ressources (SGR).
- ✓ Gestionnaire de sessions (SM).

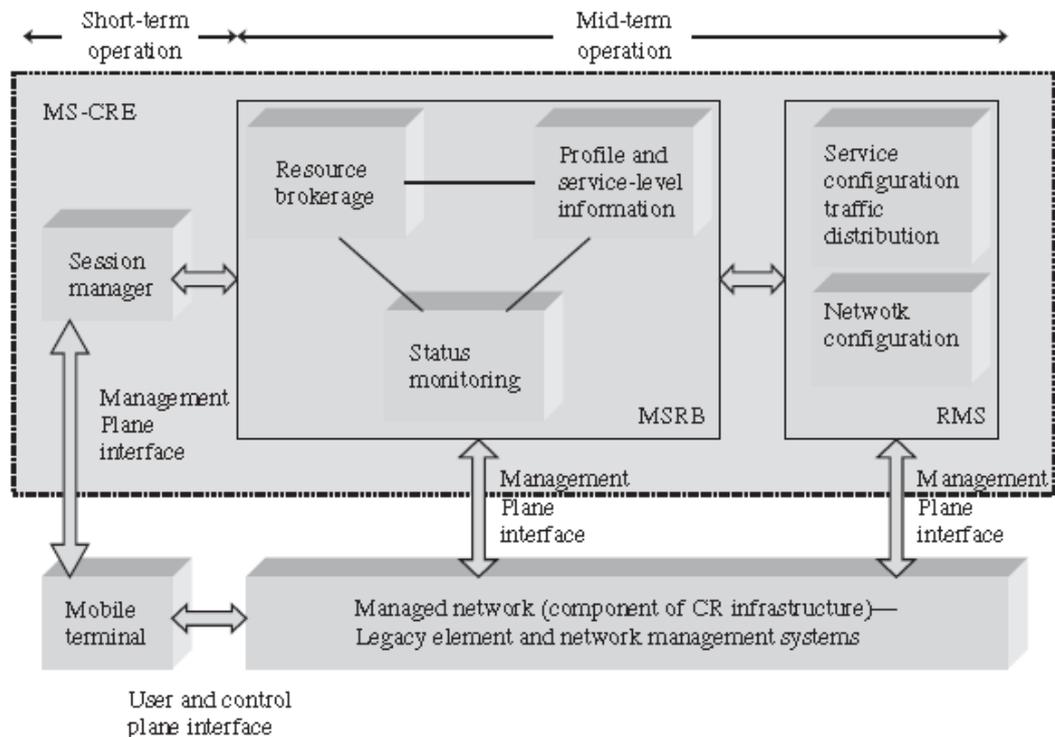


Figure I.13 : Architecture d'un système MS-CRE

L'entité MSRB identifie les événements qui doivent être manipulés par le MS-CRE et fournit les fonctionnalités de soutien auxiliaire correspondantes. L'entité RMS fournit les fonctionnalités d'optimisation nécessaires. L'entité SM est responsable d'interagir avec les utilisateurs/terminaux actifs souscrits.

I.6 CONCLUSION

Les différentes étapes de la genèse des réseaux sans fil ainsi que la tendance générale soulignent clairement l'importance de la convergence entre les réseaux hétérogènes dans l'avenir proche. Aussi, l'augmentation des capacités de transmission a permis la multiplication des services sur toutes les infrastructures. Les différents moyens de télécommunications ont permis la création de nouveaux marchés mondiaux sans considérer les frontières des nations ni les distances qui les séparent. Cette situation commence à changer graduellement grâce aux évolutions technologiques réalisées durant ces dernières années. En effet, la simplicité et la puissance du protocole IP, démontrées dans le réseau Internet, fait de ce dernier la technologie de prédilection qui offre un compromis entre le coût de déploiement et l'efficacité de fonctionnement. Dans le reste du manuscrit, nous allons nous concentrer sur l'optimisation de la QoS dans une architecture de convergence Wifi-WiMAX, en commençant par l'étude de la structure et de la gestion de la qualité de service pour chacun des deux réseaux.

II LA QoS DANS LES RESEAUX WIFI ET WIMAX

II.1 INTRODUCTION

Les technologies des réseaux sans fil sont nombreuses et sont classées selon leur couverture géographique, et chaque technologie a adopté des mécanismes pour garantir la QoS qui sont totalement différents les uns des autres. Il faudrait donc connaître ces mécanismes si l'on veut faire un basculement d'une technologie vers une autre dans des conditions optimales et efficaces et surtout continuer à garantir les mêmes paramètres de qualité de service qui sont variables d'une application à une autre, par exemple la vidéo est plus gourmande en débit constant que l'audio, qui est exigeante en termes de temps de transmission, qui doit être inférieur à un seuil garanti. Tous ces aspects font qu'il faut maîtriser ces mécanismes et dans ce chapitre, on va passer en revue justement les mécanismes de QoS instaurés dans Wifi puis dans WiMAX et proposer un mécanisme de passage des uns vers les autres. Dans ce chapitre nous allons aussi nous familiariser avec l'architecture, la structure des couches ainsi que le fonctionnement et la gestion des deux technologies précédentes.

Des efforts de normalisation pour les réseaux informatiques locaux et métropolitains ont commencé en 1979 dans l'IEEE; l'institut des ingénieurs électroniciens et électriques. En février 1980 (80/2), le groupe de travail d'IEEE 802 (ou le comité) a été fondé, consacré à la définition des normes d'IEEE pour LANs et MANs [5].

Depuis, beaucoup de sous-comités d'IEEE 802 ont été créés. Les technologies réseaux les plus répandues basées sur l'IEEE 802 sont les suivantes:

- ✓ IEEE 802.2, Logical Link Control (LLC).
- ✓ IEEE 802.3, L'Ethernet.
- ✓ IEEE 802.5, Token ring.
- ✓ IEEE 802.11, WiFi. Regroupant des réseaux locaux sans fil.
- ✓ IEEE 802.15, Bluetooth, (UWB), ZigBee.
- ✓ IEEE 802.16, Consacré à l'accès sans fil étendu métropolitain comme le WiMAX.

II.2 GENERALITES SUR LE RESEAU WIFI [5]

Le mot Wifi désigne une technologie réseau sans fil local WLAN conçue sur la base des travaux du groupe de travail IEEE 802.11. On peut concevoir le Wifi comme une extension naturelle du réseau LAN réussi « Ethernet » dans le monde sans fil. Les réseaux WLANs sont généralement déployés dans les entreprises comme le lien final qui lie les ordinateurs des clients au réseau local filaire de l'entreprise. On peut également les trouver dans les campus des universités et les espaces publics comme les aéroports et les gares de train permettant aux utilisateurs anonymes d'accéder à internet ou autres réseaux publics.

Ce succès des réseaux WLANs est très lié aux standards développés par les équipes de l'IEEE. Outre cet organisme de normalisation, les principaux acteurs de l'industrie sans fil se sont rencontrés au sein de l'Alliance Wi-Fi (Wireless Fidelity), précédemment appelé Wireless Ethernet Compatibility Alliance (WECA) pour le développement d'une technologie WLAN.

La mission de l'alliance Wi-Fi est de certifier l'interopérabilité et la compatibilité entre les différents équipements du réseau IEEE 802.11 et aussi de promouvoir cette norme. L'alliance Wi-Fi regroupe des fabricants des circuits intégrés, des fournisseurs de matériels et des fournisseurs de logiciels. Parmi eux, nous pouvons trouver des entreprises comme 3Com, Cisco Aironet, APPLE, Breezecom, Cabletron, Compaq, Dell, Fujitsu, IBM, Intersil, Lucent Technologies, Nokia, Samsung, Symbol Technologies, Wayport et Zoom.

Il y a trois standards réseaux WLAN principaux, chacun avec un degré différent d'acceptation et de déploiement: IEEE 802.11a, IEEE 802.11b et IEEE 802.11g. Le standard 802.11b est le plus populaire, également connu sous le nom Wifi ou Ethernet sans fil. Actuellement, le terme Wifi est attribué à l'ensemble de la famille de norme IEEE 802.11a/b/g/n. Le Wifi fonctionne dans l'une des bandes de fréquences sans licence 2,4 GHz, 5 GHz avec une zone de couverture allant de 3 à 500m.

Le Wifi 802.11b a été conçu pour fonctionner dans la bande 2,4 GHz (2,4 à 2,485 GHz). Au départ, cela a commencé avec des débits de données de 1 Mbps et 2 Mbps, et plus tard, en 1999, le débit a atteint 5,5 Mbit/s, pour arriver finalement à 11 Mbps dans la même année. Le Wifi 802.11b utilise comme technique de transmission le DSSS (Direct Sequence Spread Spectrum) ou le FHSS (Frequency Hopping Spread Spectrum). Le DSSS divise la bande passante en 14 sous-canaux de 22MHz chacun. En utilisant un système redondant de motif de bits, il peut fournir un mécanisme pour la détection et la correction d'erreurs.

Dans le FHSS, la bande passante disponible est répartie en 79 sous-canaux de 1 MHz chacun, partagés par plusieurs émetteurs et récepteurs. En utilisant un modèle de saut convenu lors du début de la session par les parties de transmission, il offre une bonne protection contre les bruits. Si la force du signal et la qualité se dégradent, les cartes sans fil 802.11b peuvent fonctionner à des débits diminués de 5, 2 et 1 Mbps.

Le Wifi 802.11g a été conçu pour fonctionner dans la même bande 2,4 GHz que le 802.11b. La norme IEEE 802.11g a été adoptée au 3^{ème} trimestre 2003. Elle utilise également comme technique d'accès soit le DSSS ou le FHSS. Par conséquent, le 802.11g est totalement rétro-compatible avec la norme 802.11b. Ceci assure une migration facile à partir de solutions 802.11b. Le Wifi 802.11g fonctionne à une vitesse maximale de 54 Mbit/s.

La norme IEEE 802.11a adoptée en 1999, a été conçue pour fonctionner dans les 5 GHz dans deux bandes différentes: une de 200 MHz (de 5.15 à 5.35 GHz) et l'autre de 100 MHz (5,725 à 5,825 GHz). Le Wifi 802.11a utilise un système de codage appelé OFDM (Orthogonal Frequency Division Multiplexing) qui divise la fréquence porteuse de données principale en plusieurs sous-porteuses qui sont transmises en parallèle. Cette approche évite les problèmes d'interférence due à la différence de retard des signaux arrivant sur le récepteur. La bande passante du 802.11a 200 MHz est divisée en 8 canaux indépendants de 20 MHz chacun. Les canaux de 20 MHz sont divisés en 52 sous-canaux (porteuses) de 300 KHz de large.

II.2.1 NORMALISATION

Le tableau suivant donne une liste des standards dérivés du Wifi :

Norme	Nom	Description
802.11a	Wi-Fi 5	La norme 802.11a (baptisée Wi-Fi 5) permet d'obtenir un haut débit (dans un rayon de 10 mètres : 54 Mbit/s théoriques, 27 Mbit/s réels)
802.11b	Wi-Fi	La norme 802.11b est la norme la plus répandue installée actuellement. Elle propose un débit théorique de 11 Mbit/s (6 Mbit/s réels) avec une portée pouvant aller jusqu'à 300 mètres (en théorie) dans un environnement dégagé.
802.11c	Pontage 802.11 vers 802.1d	La norme 802.11c n'a pas d'intérêt pour le grand public. Il s'agit uniquement d'une modification de la norme 802.1d afin de pouvoir établir un pont avec les trames 802.11.
802.11d	Internationalisation	Permet la récupération dynamique des contraintes de transmissions (puissance max., canaux autorisés) en fonction des réglementations locales.
802.11e	Amélioration de la qualité de service	La norme 802.11e vise à donner des possibilités en matière de qualité de service au niveau de la couche liaison de données. Ainsi, cette norme a pour but de définir les besoins des différents paquets en termes de bande passante et de délai de transmission de manière à permettre, notamment, une meilleure transmission de la voix et de la vidéo.

802.11f	Itinérance (roaming)	La norme 802.11f est une recommandation à l'intention des vendeurs de points d'accès pour une meilleure interopérabilité des produits.
802.11h		La norme 802.11h vise à rapprocher la norme 802.11 du standard Européen (Hiperlan 2, d'où le h de 802.11h) et être en conformité avec la réglementation européenne en matière de fréquences et d'économie d'énergie.
802.11i		La norme 802.11i a pour but d'améliorer la sécurité des transmissions (gestion et distribution des clés, chiffrement et authentification). Cette norme s'appuie sur l'AES (Advanced Encryption Standard) et propose un chiffrement des communications pour les transmissions utilisant les standards 802.11a, 802.11b et 802.11g.
802.11R		La norme 802.11R a été élaborée de manière à utiliser des signaux infra-rouges. Cette norme est désormais dépassée techniquement.
802.11j		La norme 802.11j est à la réglementation japonaise ce que le 802.11h est à la réglementation européenne.
802.11n	WwiSE (World-Wide Spectrum Efficiency) ou TGn Sync	La norme 802.11n est disponible depuis le 11 septembre 2009. Le débit théorique atteint les 300 Mbit/s (débit réel de 100 Mbit/s dans un rayon de 100 mètres) grâce aux technologies MIMO (Multiple-Input Multiple-Output) et OFDM (Orthogonal Frequency Division Multiplexing).
802.11s	Réseau Mesh	La norme 802.11s est actuellement en cours d'élaboration. Le débit théorique atteint aujourd'hui 10 à 20 Mbit/s. Elle vise à implémenter la mobilité sur les réseaux de type Ad-Hoc.
802.11p	WAVE	Accès sans fils dans un environnement véhiculaire
802.11r		Handoff rapide entre différents points d'accès
802.11T	WPP	Recommandations pour les méthodes de test et les métriques des performances des réseaux sans fil
802.11u		Inter-réseautage avec des réseaux sans fil non 802(ex. les réseaux cellulaires)
802.11v		Management des réseaux sans fils
802.11w		Trames de management protégées
802.11y		Wifi opérant dans les fréquences 3.65 à 3.7 GHz. Exclusivement Aux USA.

Tableau II-1 : Les normes dérivées du Wifi

Il y a encore d'autres normes qui sont en voie de développement parmi lesquelles sont le 802.11z, le 802.11aa, le 802.11ad, le 802.11ae et le 802.11af dont la sortie est prévue dans les deux prochaines années.

Selon [5], on peut percevoir l'évolution des réseaux Wifi en quatre générations :

- ✓ La première génération depuis 1997 (IEEE802.11) :
 - Connectivité entre des terminaux PCs.
 - Des points d'accès basés sur une conception de bridge.
 - Roaming.
- ✓ Deuxième génération depuis 1998 (IEEE802.11b) :
 - Une gestion plus efficace du WLAN.
 - Inter-réseutage et interopérabilité.
- ✓ Troisième génération depuis 2000 (IEEE802.11a/g) :
 - Haut débit.
 - Conception des réseaux plus ouvert et intégrés.
 - Minimisation des tailles des antennes.
 - Amélioration de la sensibilité des récepteurs.
- ✓ Quatrième génération depuis 2007 (IEEE802.11n) :
 - Débit très élevé
 - Couverture plus large.
 - Technologie plus robuste par rapport aux obstacles et bruits.

II.2.2 ARCHITECTURE

La norme IEEE 802.11 considère deux types de composants : une station cliente (en général un PC équipé d'une carte d'interface réseau sans fil) connu sous le nom d'une station (STA) et un point d'accès (AP) ou parfois appelé relais sans fil, qui fonctionne comme un pont entre la partie filaire et la partie sans fil du réseau local. Cet AP est généralement composé d'un élément radio émetteur récepteur, d'une carte réseau (par exemple Ethernet 802.3) et un logiciel de relais de couche-2 en conformité avec la norme IEEE 802.1d (la norme qui définit le langage entre les bridges). L'AP se comporte comme la station de base du réseau sans fil. Les stations sans fil comportent une carte d'accès au réseau qui est disponible dans de nombreux formats (PCI, PCMCIA, USB et aujourd'hui, puces Wifi).

Le standard IEEE 802.11 définit trois modes opératoires : le mode infrastructure, le mode ad-hoc et le mode mesh (en maille).

II.2.2.1 MODE INFRASTRUCTURE.

Dans le mode infrastructure, le réseau sans fil se compose d'au moins un point d'accès connecté à l'infrastructure du réseau fixe et d'un ensemble de stations sans fil. Cette configuration est basée sur une architecture cellulaire où le système est divisé en cellules. Chaque cellule appelée BSS (Basic Service Set) est contrôlée par une station de base (AP). Les stations dans une BSS exécutent le même protocole MAC pour concourir à l'accès au canal sans fil partagé. Afin d'étendre la zone de couverture, de multiples BSS sont utilisées avec des points d'accès qui sont reliés par un réseau filaire central appelé système de distribution (DS). L'ensemble des BSSs interconnectés et leur système de distribution forment un unique réseau IEEE802 logique qui est appelé ESS (Extended Service Set).

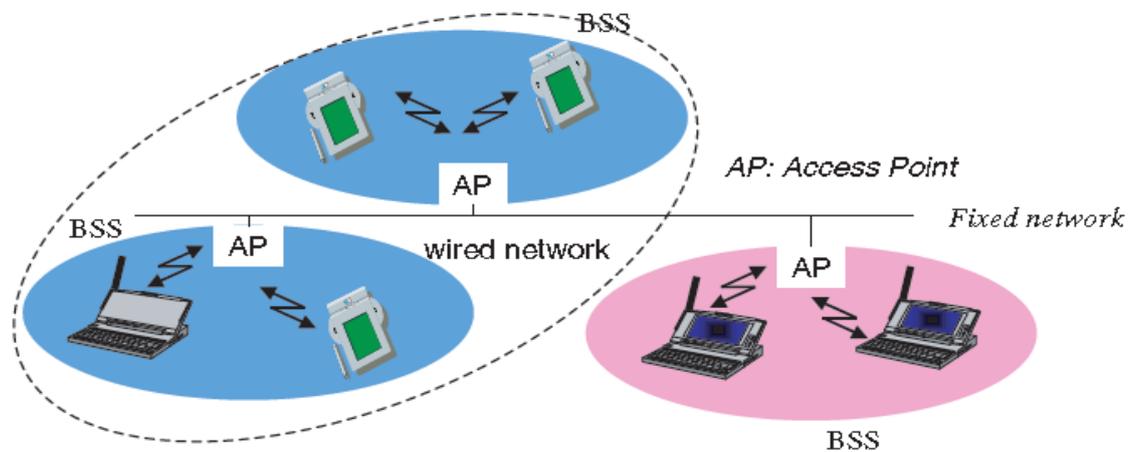


Figure II.1 : Mode infrastructure de l'IEEE 802.11

L'AP est donc l'élément central dans une cellule car il comprend les fonctionnalités nécessaires pour contrôler les communications entre les stations au sein de sa zone de couverture avec laquelle une zone de service est associée. Chaque cellule est identifiée par une adresse de 6 octets (48 bits) appelé adresse BSSID (Basic Service Set Identifier) qui correspond à l'adresse MAC de l'AP qui gère la cellule. Cependant, pour éviter les connexions indésirables, une adresse complémentaire appelé SSID (Service Set Identifier) est utilisé pour identifier l'ESS, c'est une chaîne de caractère d'une largeur maximale de 32 octets. Le SSID est utilisé afin de garantir l'authentification et l'identification entre un AP et un client. Toute station qui souhaite se connecter à l'ESS doit donc savoir au préalable la valeur du SSID. Il convient de noter que ce mécanisme n'est pas considéré comme une protection de sécurité parce que l'identificateur du réseau est généralement transmis en clair dans des trames balises. Le mode infrastructure est le plus utilisé parmi les modes du Wifi.

II.2.2.2 LE MODE AD HOC

Le mode ad hoc (Figure 2.2) représente simplement un groupe de stations IEEE 802.11 sans fil qui communiquent directement entre elles sans avoir un lien avec un point d'accès ou d'une connexion à un réseau fixe à travers le DS. Cette configuration est parfois appelée une configuration peer-to-peer. Chaque station peut établir une communication avec une autre station dans la cellule qui est appelé une cellule indépendante IBSS (Independent Basic Service Set).

Comme dans le mode infrastructure, un réseau ad hoc est également identifié par un identificateur SSID.

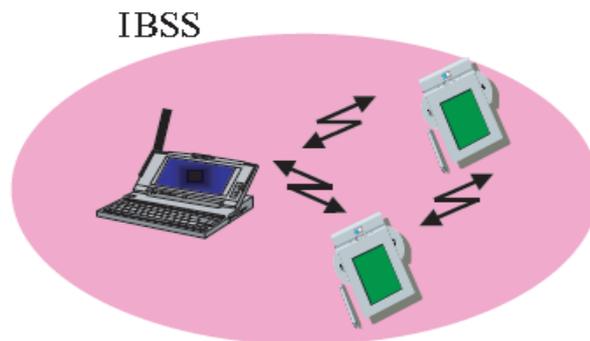


Figure II.2 : Le mode Ad hoc

II.2.2.3 CONFIGURATIONS MAILLES SANS FIL

Récemment, les réseaux hybrides sans fil ont émergé comme une solution prometteuse. Dans de tel réseau, tout nœud mobile peut disposer d'une connectivité soit directement, soit via un nœud relais à un réseau infrastructure.

Suivant les définitions de la norme IEEE 802.11s, une architecture maillée typique divise les nœuds sans fil en deux classes majeures :

- ✓ Les nœuds de classe maillée : sont des nœuds capables de supporter des services relais (ex. routage). Ils peuvent éventuellement être des points d'accès.
- ✓ Les nœuds de classe non-maillée : comprend les simples stations clientes

Un exemple de réseau maillé est illustré dans la figure 2.3. Les dispositifs qui font partis du réseau maillé sont appelés points de maille ou MP (Mesh Point). Un MP particulier est le point d'accès de maille MAP (Mesh Access Point) qui fournit des services AP. Les simples nœuds clients sont appelés stations STA.

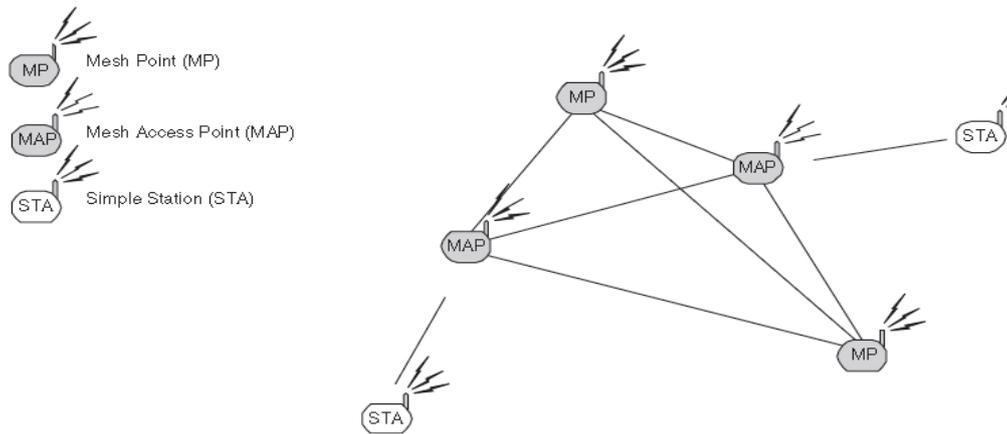


Figure II.3 : Le mode mesh WLAN contenant des MPs, MAPs et STAs

II.2.2.4 LES SERVICES DE BASE DU RESEAU 802.11

La norme IEEE 802.11 prend en charge une série de services de base qui sont :

- ✓ Association / dissociation / réassociation
- ✓ La livraison des trames MAC/MSDU
- ✓ Authentification / désauthentification
- ✓ Diffusion.
- ✓ Balise et sondage
- ✓ Protection des données / confidentialité
- ✓ La synchronisation temporelle des couches supérieures/ planification de QoS
- ✓ Les Mesures radio

II.2.2.5 LA STRUCTURE DES SOUS COUCHES

D'une façon similaire aux autres normes, la norme IEEE 802.11 se concentre plus sur les deux sous-couches du modèle OSI, la couche physique et la couche liaison de données qui est divisée en deux sous-couches (MAC et LLC) comme le montre la Figure 2.4. La couche physique, qui est en charge de la transmission des trames MAC sur le support sans fil, utilise plusieurs techniques de modulation et de codage binaire.

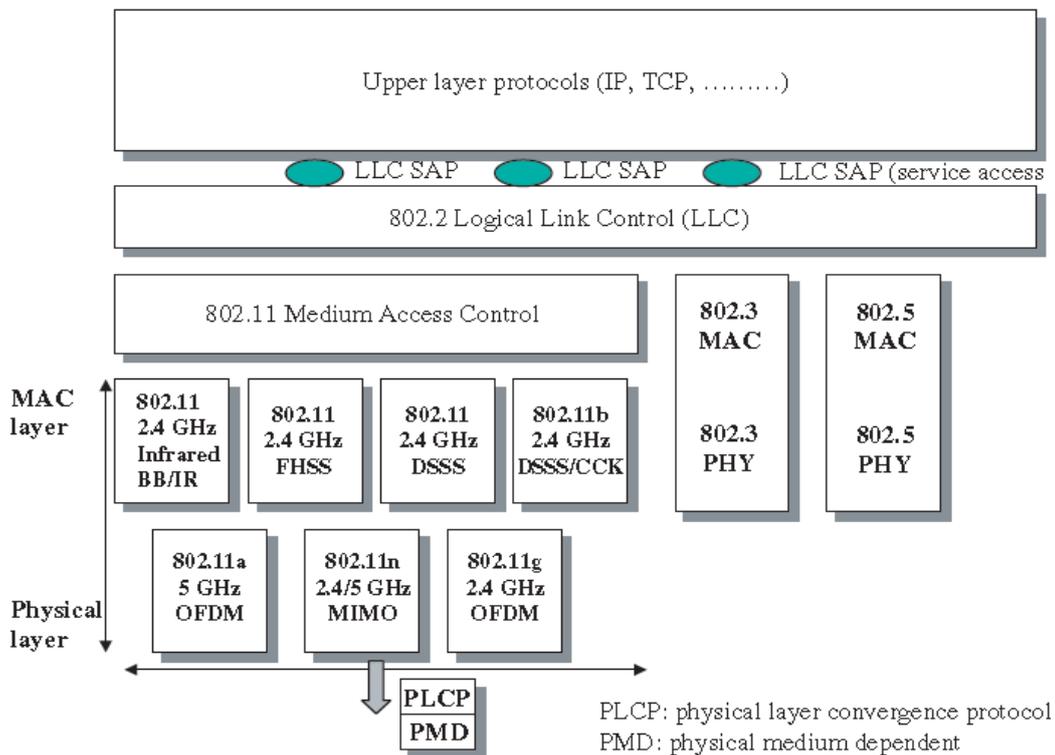


Figure II.4 : Le modèle de référence OSI, IEEE 802.11 et la famille 802

II.2.2.6 LA COUCHE PHYSIQUE

La norme IEEE 802.11 originale définit trois couches physiques différentes :

- ✓ Une couche physique radio utilisant la technique FHSS dans la bande 2,4 GHz
- ✓ Une couche physique radio utilisant la technique DSSS dans la bande 2,4 GHz
- ✓ Une couche physique pour la transmission infrarouge IR.

Les normes 802.11 établies par la suite ont modifié la couche physique en ajoutant des éléments tels que :

- ✓ Modulation CCK (Complementary Code Keying) ajouté à la technique DSSS, la couche physique devient ainsi HR/DSSS (High rate DSSS) dans la norme 802.11b.
- ✓ Modulation OFDM (Orthogonal Frequency Division Multiplexing) dans le cas de la norme IEEE 802.11a/g. La technique OFDM sera détaillée dans la partie des généralités sur le WiMAX.
- ✓ Les techniques MIMO (Multiple Input Multiple Output) utilisant des antennes multiples pour l'émission et la réception définie dans la norme IEEE 802.11n

La couche physique est divisée en deux sous-couches :

- ✓ La sous-couche de convergence ou PLCP (Physical Layer Convergence Sublayer Procedure) qui adapte les données au type de transmission choisi.
- ✓ La sous-couche dépendante du support et type de transmission PMD (Physical Medium Dependent).
- ✓ La transmission d'un paquet physique dépend de deux procédures : La détection de porteuse CA (Carrier Sens) et la détermination si le canal est libre CCA (Clear Channel assesment). La durée minimale d'une opération CS/CCA est égale à la durée d'un slot physique dont la valeur dépend du type de transmission sélectionné. Le tableau 2.2 suivant donne les débits résultants des différentes modulations utilisées actuellement.

Protocole	Fréquence	Débit Théorique Mb/s	Modulation	Portée (Intérieur)	Portée (Extérieur)
802.11a	5.15-5.35 /5.47-5.725 /5.725-5.875 GHz	6, 9, 12, 18, 24, 36, 48, 54	OFDM	~25 m	~75 m
802.11b	2.4-2.5 GHz	5.5, 11	DSSS	~35 m	~100 m
802.11g	2.4-2.5 GHz	6, 9, 12, 18, 24, 36, 48, 54	OFDM, DSSS	~25 m	~75 m
802.11n	2.4 GHz	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2	OFDM	~50 m	~125 m
802.11n	5 GHz	15, 30, 45, 60, 90, 120, 135, 150	OFDM	~50 m	~125 m

Tableau II-2 : les différents débits en fonction de la modulation

II.2.2.7 LA COUCHE LIAISON DE DONNEES

La Couche liaison de données est composée de deux sous-couches : la sous couche LLC (Logical link Control) et la sous couche MAC (Media Access Control). La sous couche LLC est définie par la norme IEEE 802.2 et permet de connecter un réseau local sans fil à tout réseau local de la famille IEEE 802.X. La couche MAC spécifique à la norme IEEE 802.11 définit les mécanismes d'accès au milieu de transmission comme suit.

II.2.2.8 LES MECANISMES D'ACCES AU MEDIA

Dans la norme originale, la sous couche MAC définit deux méthodes d'accès différentes :

- ✓ La méthode de transmission asynchrone à contention DCF (Distributed Coordination Function) pour l'accès aléatoire comme celui d'IEEE 802.3 avec des modifications spécifiques au Wifi.
- ✓ La méthode à contrainte temporelle PCF (Point Coordination Function) : pour un accès contrôlé et centralisé autour d'un coordinateur.
- ✓ La méthode DCF est destinée pour un service de type « meilleurs efforts » égalisant les chances des différentes stations pour l'accès au média.

Le PCF est basé sur une approche de polling (interrogation des stations tour à tour) et une transmission différenciée suivant les priorités des stations. Le PCF n'a jamais été réellement implémenté et a été remplacé plus tard par la méthode HCCA qui fait partie de la fonction HCF (Hybrid Coordination Function) introduit dans la norme IEEE802.11e pour prendre en compte la QoS des applications temps-réel. Les deux méthodes peuvent être utilisées simultanément où une trame balise (Beacon) définit la période basée contention CP et la période sans Contention CFP comme suit :

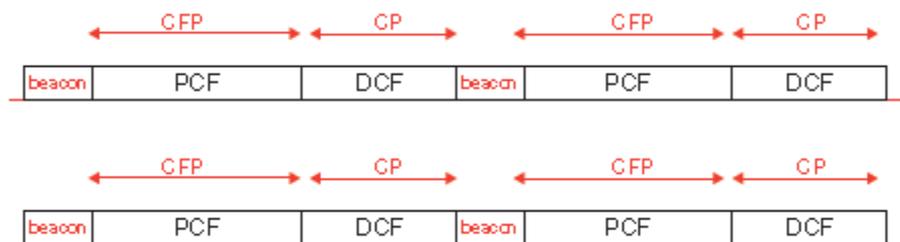


Figure II.5 : Méthodes d'accès au média 802.11.

II.2.2.9 LA METHODE DCF

Dans la méthode DCF originale le mécanisme d'accès au média est basé sur la méthode CSMA/CA (Carrier Sens Multiple Access/Collision Avoidance). Dans cette méthode si une station veut transmettre elle commence par écouter le canal de transmission. Si le média est occupé la transmission est différée durant un temps déterminé par un algorithme appelé « Backoff ». Dans le cas contraire et si le média est libre durant un délai appelé DIFS (Distributed Inter-Frame Space) alors la station peut émettre. La station (ou point d'accès) qui va recevoir les données vérifie leur intégrité et renvoie un acquittement. L'arrivée de l'acquiescement au niveau de la station émettrice signifie qu'il n'y a pas eu collision. Dans le cas contraire et après un délai d'attente, la station retransmet les données.

Le délai DIFS n'est pas le seul utilisé dans la sous couche MAC, où d'autres délais inter-frames sont définis comme suit :

- ✓ Le SIFS (Short Inter-Frame Space) qui est le délai le plus court, utilisé pour séparer les trames appartenant au même dialogue (ex. une trame donnée et la trame l'acquittant).
- ✓ Le PIFS (Priority Inter-Frame Space) qui est plus long que le SIFS mais moins long que le DIFS. Il est utilisé dans la méthode PCF/HCF pour permettre au coordinateur d'accéder au média avant toutes les autres stations.
- ✓ Le EIFS (Extended Inter-Frame Space) qui est le plus long et qui est utilisé par la station qui reçoit une trame erronée supposant ainsi qu'il y a eu collision. Ce délai permet d'éviter une collision supplémentaire.

La durée des délais est définie en fonction de la durée du SIFS et celle d'un slot physique. Ces derniers étant définis différemment selon le type de modulation utilisé (voir tableau ci-dessous). Ces différents délais donnent une granulation dans la priorité d'accès au média. En effet, plus le délai est court et plus la probabilité d'accès est grande.

L'algorithme « Backoff » d'attente d'une durée aléatoire est utilisé quand une station émettrice détecte que le média est occupé juste avant la transmission d'une nouvelle trame. Il est également utilisé dans le cas d'une collision (non réception d'acquiescement) et ceci après chaque retransmission. Il est même utilisé après une transmission réussie pour ne pas monopoliser le média. La durée d'attente BW (backoff window) est calculée comme suit :

$$BW = CW \times NA \times SlotTime \quad \text{Équation II-1}$$

Où :

- ✓ CW est la taille de la fenêtre de contention avec $CW \in [CW_{min}, CW_{max}]$. A la première exécution de l'algorithme « Backoff » CW est égale à CW_{min} et à chaque retransmission CW est doublé jusqu'à l'arrivé à la valeur CW_{max} .
- ✓ NA est un nombre aléatoire pris entre 0 et 1.
- ✓ Slot Time est la durée d'un slot physique.

Il faut noter que la décrémentation du compteur d'attente du Backoff se fait seulement quand le média est libre. Le tableau 2.3 suivant donne les valeurs du SIFS, slot physique, CW_{min} et CW_{max} pour les différents types de modulation.

Paramètre	Type de modulation	Durée
Slot Physique (micro seconde)	FHSS	50
	DSSS, HR/DSSS	20
	OFDM	9
	IR	8
SIFS (microseconde)	FHSS	28 ± 10%
	DSSS, HR/DSSS	10
	OFDM	16
	IR	10
CWmin	FHSS	15
	DSSS, HR/DSSS	31
	OFDM	15
	IR	63
CWmax	FHSS,DSSS, HR/DSSS,OFDM,IR	1023

Tableau II-3 : Les valeurs du SIFS, CW_{min} , CW_{max} en fonction de la modulation

La figure suivante décrit un mécanisme d'accès DCF élémentaire.

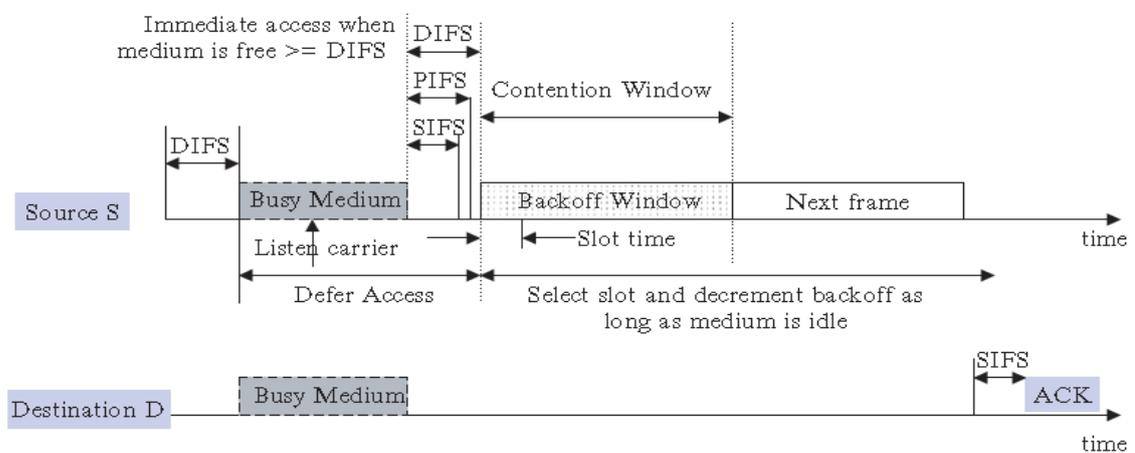


Figure II.6 La méthode d'accès CSMA/CA

La méthode originale CSMA/CA a été améliorée ensuite par l'introduction de la détection de porteuse virtuelle et les messages RTS (Request To Send) / CTS (Clear to Send). Ainsi dans la nouvelle version du DCF, si une station veut émettre elle commence par l'envoi d'une trame de contrôle appelée RTS. Cette dernière contient l'adresse source, l'adresse destination et la durée de transmission demandée (plus la durée de la trame d'acquiescement). La station de destination répond en envoyant une trame de contrôle appelée CTS qui contient la même information de durée. Toutes les autres stations qui détectent soit la trame RTS ou la trame CTS mettent à jours une variable interne appelée « indicateur de la porteuse virtuelle » NAV pour signaler que le média va être occupé durant la période indiquée dans les messages RTS/CTS. Cette modification rend la méthode DCF plus efficace dans le cas d'encombrement de cellules en diminuant la probabilité de collision. La figure suivante illustre la transmission par la méthode CSMA/CA avec RTS/CTS.

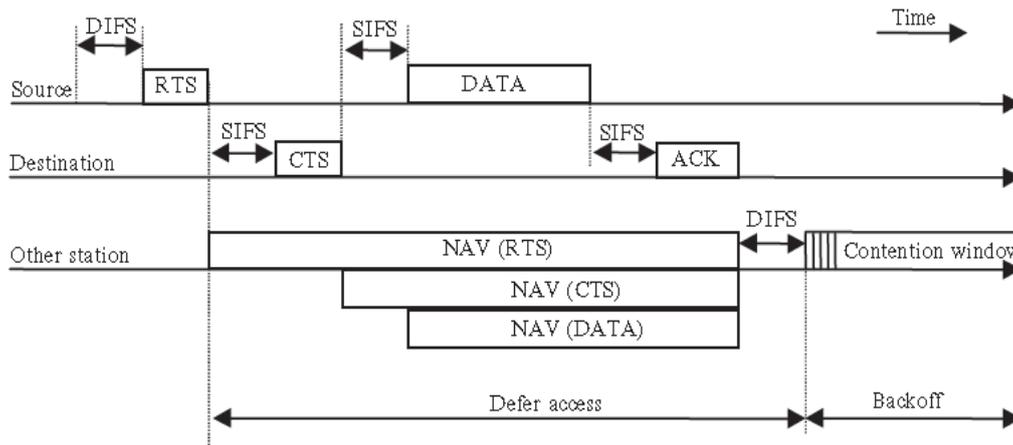


Figure II.7 La méthode d'accès CSMA/CA avec RTS/CTS

II.2.2.10 LA FONCTION HCF

La fonction HCF a été introduite pour supporter les applications ayant des exigences de QoS. Elle comporte deux mécanismes principaux :

- ✓ Une nouvelle méthode orientée contention appelée EDCA (Enhanced Distributed Channel Access) qui est l'extension de la méthode DCF.
- ✓ Une méthode d'accès sans contention appelée HCCA (HCF Controlled Channel Access).

Dans l'EDCA, la QoS est supportée en introduisant quatre catégories d'accès ACs (Access Category) ayant des priorités différentes. Chacune de ces ACs représente une fonction DCF avec un délai DIFS différent pour chaque AC d'où la nouvelle nomenclature AIFS (Arbitration Inter-Frame Space). Dans l'HCCA un coordinateur hybride HC alloue des opportunités de transmission TXOPs aux stations sur une base de polling. Ces TXOPs sont calculés selon les paramètres de la QoS exigés. La TXOP est un paramètre clé introduit par l'HCF, il représente l'intervalle temporel durant lequel une station peut tenter d'envoyer des trames. Une TXOP peut être obtenue en gagnant une contention EDCA ou peut être simplement attribué par un point d'accès après un polling HCCA.

Ainsi, l'HCF introduit au niveau de la couche MAC quatre files d'attente suivant l'AC et huit files d'attente suivant le flux de trafic TS (Trafic Stream). Quand une trame arrive à la couche MAC elle est marquée par un identificateur de priorité de trafic TID (de 0 à 15) selon ses paramètres de QoS.

Les trames avec un TID entre 0 et 7 sont classées dans les quatre files d'ACs suivant des règles EDCA préétablies (voir tableau 2.4). La QoS au niveau de ces quatre files est supportée sur une base de priorité, où le flux avec des paramètres de QoS plus strictes, est doté d'une priorité plus élevée. La différenciation des ACs est faite par l'attribution de différents délais (AIFS, CW_{min} et CW_{max}) aux différents ACs. Ainsi pour AC_i et AC_j avec $0 \leq i \leq j \leq 3$ alors on aura $CW_{min}[i] \geq CW_{min}[j]$, $CW_{max}[i] \geq CW_{max}[j]$ et $AIFS[i] \geq AIFS[j]$. Aussi, chaque niveau de priorité EDCA a une fonction Backoff différente, attribuant un délai plus court à la priorité la plus élevée. Ainsi le flux le plus prioritaire aura plus de chance pour accéder au média.

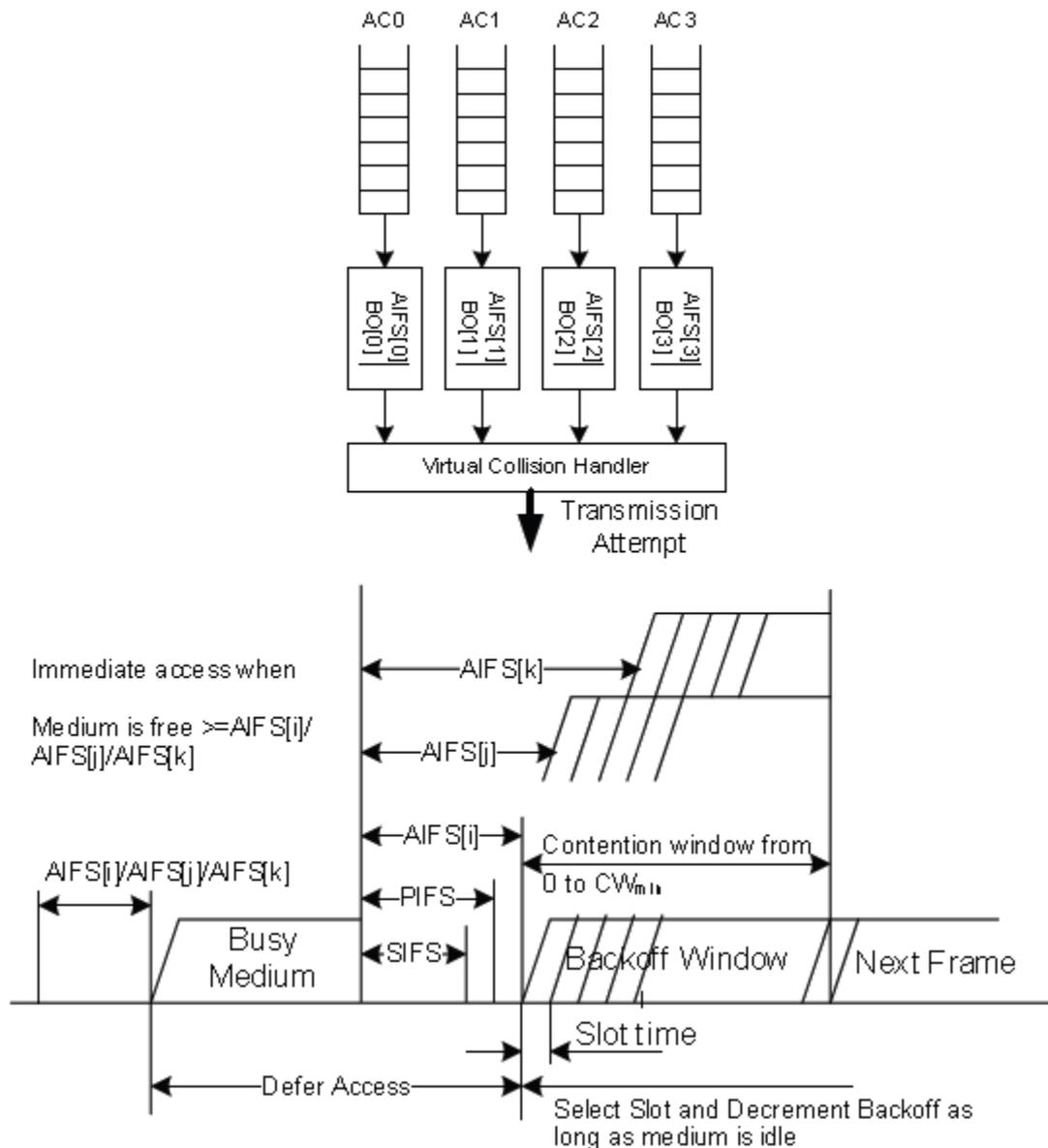


Figure II.8 : Exemple d'accès au média basé EDCA

Priorité	Catégorie d'accès AC	Type de trafic
7	3	Voie à 64 Kbps
6	3	Voie à 64 Kbps
5	2	Vidéo à 1.5 Kbps
4	2	Vidéo à 1.5 Kbps
3	1	Sondage Vidéo
2	0	Meilleur effort
1	0	Arrière plan
0	0	Arrière plan

Tableau II-4 : Exemple de classification de TID en AC

Les trames ayant un TID entre 8 et 15 sont classées dans huit files de TSs suivant des règles HCCA préétablies (Les paramètres TS à respecter). Ici la QoS est supportée sur une base de paramétrage stricte où les exigences de QoS d'un flux (Délai maximum, Jitter toléré, Perte de Paquets tolérée, ...etc.) sont tous respectés. Ceci est accompli en choisissant une stratégie adéquate d'attribution de TXOPs aux différentes stations. Ainsi un coordinateur hybride HC est défini au niveau d'un point d'accès de QoS QAP et utilise le délai PIFS pour gagner le contrôle du média et allouer des TXOPs aux stations de QoS, les QSTAs. Ces dernières sont interrogées par le HC pour fournir une requête de réservation de QoS avec un enregistrement des spécifications de trafic TSPEC dont les majors éléments sont :

- ✓ Débit nominal (ρ) exprimé en bps.
- ✓ Délai limite (D) pour un paquet exprimé en millisecondes.
- ✓ Intervalle maximum de service (SI_{max}) qui est le délai maximum entre deux allocations de TXOP successives exprimé en millisecondes.
- ✓ Taille nominal d'un paquet (L) en octet.
- ✓ Débit minimum (R)

II.2.3 LA GESTION DE QOS DANS LE WIFI

Les mécanismes d'accès au média présentent dans le IEEE 802.11 ne fournissent pas de garantie concernant la QoS. Par contre l'apparition de la norme IEEE 802.11e a proposé des mécanismes d'accès au média pour le support de la QoS. Cependant, ces derniers ne sont pas suffisants à eux seuls pour garantir la QoS d'où la nécessité de proposer des algorithmes complémentaires pour l'allocation de ressources et le contrôle d'admission de stations (ou flux de données) dans le réseau.

Toutefois, contrairement aux réseaux câblés, dans les réseaux sans fil une station n'a pas connaissance de la disponibilité des ressources réseau et ne peut donc pas prendre des décisions précises sur le fait d'admettre ou non un nouveau flux. En outre, avec le mécanisme d'accès au canal basé contention CSMA/CA, l'allocation de la bande passante est presque impossible, ce qui mène à une garantie de la QoS soft. En raison de ces deux difficultés majeures, le contrôle d'admission et la réservation de la bande passante dans les réseaux sans fil IEEE 802.11 est assez difficile [4].

En se basant sur la façon dont les problèmes de la gestion des ressources sont abordés, nous pouvons classer les modèles de gestion de ressources existants en trois catégories générales [6][7]:

- Les modèles basés DCF: Ces modèles essaient de fournir suffisamment de support de QoS sans changer les algorithmes d'ordonnancement originaux DCF / PCF.
- Les modèles basés EDCA: Ces modèles tentent d'améliorer la performance des flux en temps réel, la différenciation des services et les stratégies de la QoS offerte par l'algorithme d'ordonnancement EDCA.
- Les modèles basés HCCA : En contraste avec EDCA, Il n'ya pas beaucoup de recherches sur l'extension du HCCA du fait que cette méthode est déjà centralisée et peut naturellement renforcer le control d'admission.

II.2.4 LES MODELES DE GESTION DE LA QoS BASES DCF

Les caractéristiques communes de ces modèles est de concevoir des techniques supplémentaires pour estimer les ressources du réseau comme la bande passante et le retard et ensuite appliquer des stratégies de gestion de QoS tels que le contrôle d'admission et la réservation de débit pour assurer la performance des trafic voix et vidéo temps réel. Suivant les techniques appliquées, on peut classer ces approches en :

- ✓ Approches basées sur un modèle mathématique.
- ✓ Approches basées sur la condition du canal.
- ✓ Approches basées sur le test de paquet.
- ✓ Approches basées sur l'estimation de la bande passante.

II.2.4.1 LES APPROCHES BASEES SUR UN MODELE MATHEMATIQUE

Dans ces approches, un modèle mathématique modélisant le réseau est utilisé, pour pré-calculer les conséquences de l'admission d'un flux. Ainsi, le contrôle d'admission est basé sur les prédictions du modèle. La modélisation des performances de la méthode DCF a été intensivement étudiée. Les modèles utilisés sont souvent basés sur la supposition d'un état saturé où toutes les files d'attente d'émission des différentes stations sont toujours pleines. Cependant, un algorithme de contrôle d'admission doit au contraire éviter une telle saturation pour ne pas dégrader les performances des flux temps réel. Ainsi, Le model utilisé doit être basé sur la supposition de non saturation des files d'attente.

En général, il est très difficile d'obtenir des résultats précis avec une telle analyse de performance, ainsi ces approches ne sont pas largement utilisées pour la conception de la gestion de la QoS. Donc les approches basées sur les mesures directes sont favorisées parceque simples, dynamiques et pratiques.

II.2.4.2 LES APPROCHES BASEES SUR LA CONDITION DU CANAL.

Un exemple d'une telle approche est proposé dans la méthode ARME (Assured Rate MAC Extension) pour fournir un service assuré aux applications multimédia. L'idée de base de l'ARME est d'utiliser des indicateurs comme l'état de la file d'attente et le taux moyen de collision pour modifier la fenêtre de contention CW et réguler les flux de transmission. Par exemple, si la file d'attente des paquets est vide, le CW est légèrement augmenté. Par contre si la file est pleine, CW est diminué de telle sorte que la station ait un accès plus fréquent.

II.2.4.3 LES APPROCHES BASEES SUR LE TEST DE PAQUET.

Un exemple d'une telle approche est proposé dans l'algorithme VMAC (Virtual MAC) qui surveille passivement le canal en utilisant des paquets MAC virtuels comme sondes pour estimer les performances du service local. Etant donné que le délai dans les canaux sans fil atteint ses limites avant le débit, l'algorithme VMAC propose de prendre le délai des paquets virtuels comme indicateur des conditions de canal. Ainsi si ce délai est petit, l'admission d'un flux additionnelle est acceptable.

II.2.4.4 LES APPROCHES BASEES SUR L'ESTIMATION DE LA BANDE PASSANTE.

Ces approches sont conçues pour surveiller le canal sans fil, mesurer son utilisation et/ou utiliser le trafic existant pour estimer directement la bande passante disponible. Un exemple de ces approches serait la méthode PAC qui mesure les transmissions effectuées dans la zone de détection de porteuse, ceci en mesurant le temps durant lequel le canal est occupé, utilisé pour une transmission ou utilisé pour une réception. La bande passante disponible est ainsi mesurée en utilisant la formule :

$$B_{dispo} = (1-U) B_{max} \quad \text{Équation II-2}$$

Où U est le taux d'utilisation du canal estimé par la méthode précédente et B_{max} est une approximation de la bande passante maximale atteignable du canal. Ainsi, si la bande passante disponible estimée est assez large, l'admission de flux additionnel est acceptée.

II.2.5 MODELE DE GESTION DE LA QOS BASEE EDCA

De nombreux travaux de recherche se sont concentrés sur les questions de gestion des ressources en termes de l'amélioration dynamique du partage de la bande passante et la garantie de délais en ajustant les paramètres de configuration de l'EDCA. Suivant les techniques appliquées, on peut classer les approches de ces modèles en :

- ✓ Contrôle d'admission et réallocation des priorités.
- ✓ Approches basées sur l'ajustement des TXOPs et CWs.
- ✓ Approches basées sur le contrôle d'admission et de données.

II.2.5.1 LE CONTROLE D'ADMISSION ET LA REALLOCATION DES PRIORITES

Un exemple de cette approche est la méthode où les flux sont classifiés en deux classes, haute priorité (4 à 7) et basse priorité (1 à 3). Aussi une fonction appelée $flow_length(p)$ est définie pour une priorité p et est égale à la bande passante totale demandée par les flux ayant une priorité p . Quand un nouveau flux est initialisé avec une priorité initiale donnée, sa priorité est réaffectée suivant deux règles :

- ✓ On affecte au nouveau flux une priorité de la même classe ayant le plus petit $flow_length$
- ✓ Si plusieurs priorités satisfont la condition précédente, la priorité la plus proche de priorité originale est choisie.

Ce type d'algorithme donne des résultats intéressants dans les réseaux à grandes charges.

II.2.5.2 APPROCHES BASEES SUR L'AJUSTEMENT DES TXOPS ET CWS.

Un exemple de cette approche est la méthode où un distributeur de la bande passante du réseau sans fil est défini au niveau du point d'accès. Quand une nouvelle station S_{n+1} ayant des exigences de transmission noté R_{n+1} demande l'accès, le distributeur recalcule en premier lieu les CWS et les TXOPs optimaux pour toutes les stations de S_1 à S_{n+1} . Ensuite les performances résultantes projetées r_i sont calculées pour toutes les stations. La nouvelle station est acceptée seulement si pour toutes les stations S_i nous avons $r_i \geq R_{n+1}$.

II.2.5.3 LES APPROCHES BASEES SUR LE CONTROLE D'ADMISSION ET DE DONNEES

Dans ces approches, deux composantes majeures sont proposées :

- ✓ Contrôle de données : les paramètres tels que CW_{min} , CW_{max} et AIFS des flux de données de la classe « meilleur effort » sont ajustés dynamiquement pour protéger les performances des trafics voix et vidéo existants. essentiellement, ces paramètres doivent être incrémentés lorsqu'il y a trop de transmission de données et vice versa.
- ✓ Contrôle d'admission : le temps de transmission et le surplus de TXOP (temps libre disponible pour la transmission) sont calculés périodiquement au niveau du point d'accès. Les valeurs résultantes sont utilisées pour contrôler l'admission du nouveau flux et protéger les flux voix et vidéo existants. En effet, si ces derniers flux sont déjà admis dans le réseau, un nouveau flux n'est accepté que si le surplus est assez grand.

II.2.6 CONCLUSION

Donc on arrive au fait que pour gérer la QoS de manière efficace, il est primordial d'avoir un contrôle d'admission qui accepte ou refuse l'admission de nouveaux flux, pour ne pas dégrader la QoS de flux déjà acceptés dans le réseau. Chose qui a été démontrée par plusieurs travaux dans l'état de l'art.

II.3 GENERALITES SUR LE RESEAU WiMAX

Le réseau WiMAX est basé sur différentes versions de la norme 802.16 et existe dans deux configurations fixe et mobile. La configuration fixe est utilisée pour concurrencer les technologies d'accès DSL. La configuration mobile peut concurrencer les hotspots du wifi comme elle peut également concurrencer les réseaux cellulaires. Ainsi le WiMAX est considéré comme une technologie B3G (Katz et Fitzek 2009).

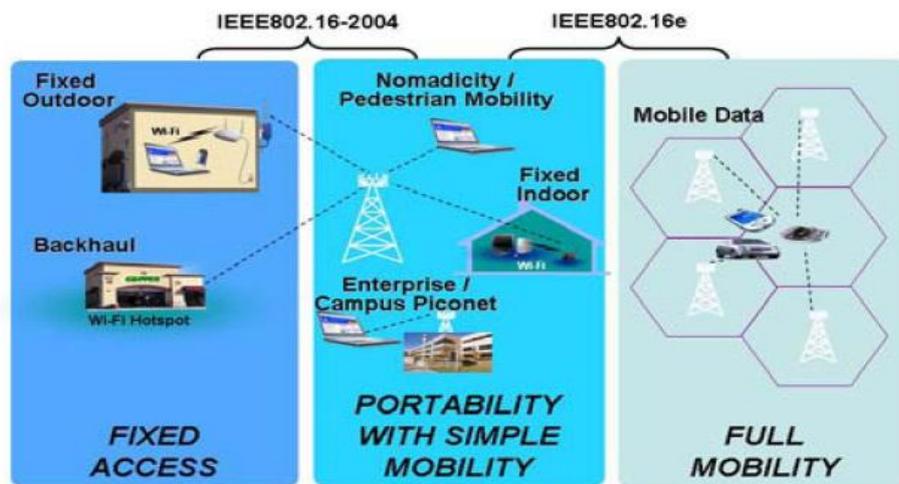


Figure II.9 : Les Différentes configuration du WiMAX

II.3.1 NORMALISATION ET LE STANDARD IEEE802.16

II.3.1.1 LA NORME IEEE 802.16 :

Les principales caractéristiques de la norme originale étaient:

- ✓ Une porteuse < 11 GHz.
- ✓ La technique de modulation utilisée est l'OFDM (Orthogonal Frequency Division Multiplex).
- ✓ Un débit minimum de l'ordre de 10 Mbit/s avec une espérance d'atteindre les 100 Mbit/s.
- ✓ Une distance de couverture jusqu'à 20 km.

Deux groupes de travail IEEE ont été créés au début et plusieurs documents ont été publiés (voir tableau ci-dessous).

Standard	Description	Publié	Statut
IEEE std 802.16-2001	définit des réseaux métropolitains sans fil utilisant des fréquences supérieures à 10 GHz (jusqu'à 66 GHz)	8 avril 2002	obsolètes
IEEE std 802.16c-2002	définit les options possibles pour les réseaux utilisant les fréquences entre 10 et 66 GHz.	15 janvier 2003	
IEEE std 802.16a-2003	amendement au standard 802.16 pour les fréquences entre 2 et 11 GHz.	1er avril 2003	
IEEE std 802.16-2004 (également désigné 802.16d)	il s'agit de l'actualisation (la révision) des standards de base 802.16, 802.16a et 802.16c.	1er octobre 2004	obsolète/ actifs
IEEE 802.16e (également désigné IEEE std 802.16e-2005)	apporte les possibilités d'utilisation en situation mobile du standard, jusqu'à 122 km/h.	7 décembre 2005	actifs
IEEE 802.16f	Spécifie la MIB (Management Information Base), pour les couches MAC (Media Access Control) et PHY (Physical)	22 janvier 2006	
IEEE 802.16m	Débits en nomade ou stationnaire jusqu'à 1 Gbit/s et 100 Mbits/s en mobile grande vitesse. Convergence des technologies WiMAX, Wi-Fi et 4G	-	En cours

Tableau II-5 : différentes normes du WiMAX harmonisez les titres

La version 802.16-2004 était certainement très utile, remplaçant un ensemble de documents tous décrivant différentes parties de la même technologie. Cependant, après sa publication, il y a eu toujours besoin d'une mise à niveau, principalement pour l'ajout des dispositifs mobiles ainsi que la correction de quelques erreurs. Ceci a mené à la norme 802.16e approuvée décembre 2005 et publiée en février 2006. Les différences principales de 802.16e par rapport au 802,16-2004 sont les suivantes (liste non exhaustive):

- ✓ Introduction du terme MS (Station Mobile) au lieu du terme SS (Subscriber Station) utilisé auparavant pour désigner une station d'abonné.
- ✓ Introduction du Handover au niveau de la couche MAC.
- ✓ Introduction et gestion de la mise en veille (gestion de la puissance) pour les stations mobiles.
- ✓ Reconstruction complète du mode OFDMA pour la couche physique, introduisant ainsi le SOFDMA (Scalable OFDMA).
- ✓ La mise à jour de la sous couche sécurité.
- ✓ Amélioration de l'implémentation MIMO et ASS (Techniques avancées d'antennes).
- ✓ Une cinquième classe de qualité de service QoS a été introduite dénommée ertPS.

II.3.1.2 LE FORUM WiMAX



Les normes IEEE 802 fournissent seulement la technologie. Il est nécessaire alors d'avoir d'autres organismes pour la certification de la conformité et la vérification de l'interopérabilité des interfaces industrialisées. Le forum WiMAX (interopérabilité mondiale pour l'accès micro-onde) a été créé en juin 2001 avec l'objectif de fournir la certification de la conformité, de la compatibilité et de l'interopérabilité des produits IEEE 802.16. Après une période de décadence, le forum WiMAX a été réactivé en avril 2003. Les membres du forum WiMAX sont des constructeurs de circuit à semi-conducteurs, d'autres sont de constructeurs de matériel de téléphonie et de réseau. Intel et Nokia, avec d'autres, ont joué un rôle principal dans la création du forum. Une liste non exhaustive des membres du Forum WiMAX est donnée ci-dessous:

- ✓ Constructeurs: Airspan, Alcatel, Alvarion, Broadcom, Cisco, Ericsson, Fujitsu, Huawei, Intel, LG, Lucent, Motorola, Navini, Nokia, Nortel, NEC Proxim, Sagem, Samsung, Sequans, Siemens, ZTE, etc.
- ✓ Fournisseurs de service: British Telecom, France Telecom, KT (Korea Telecom), PCCW, Sprint Nextel, Telmex, etc.

II.3.2 LA STRUCTURE WiMAX

Comme avec toutes les normes IEEE 802.X, les protocoles définis concernent les deux couches les plus basses du modèle OSI (couche physique et Liaison de donnée). La couche liaison de données étant divisée en deux sous couches : MAC (Media Access Control) et LLC (Logical Link Control).

L'architecture de couches et de protocoles définie dans le WiMAX/802.16 est montrée dans la figure 2.9 suivante. On peut voir que la norme 802.16 définit seulement la couche physique et la couche MAC, avec la couche LLC appliquant très souvent la norme IEEE 802.2. La couche MAC elle-même divisée en trois sous couches : CS (sous couche de convergence), CPS (sous couche des parties communes du MAC) et SS (sous couche de sécurité).

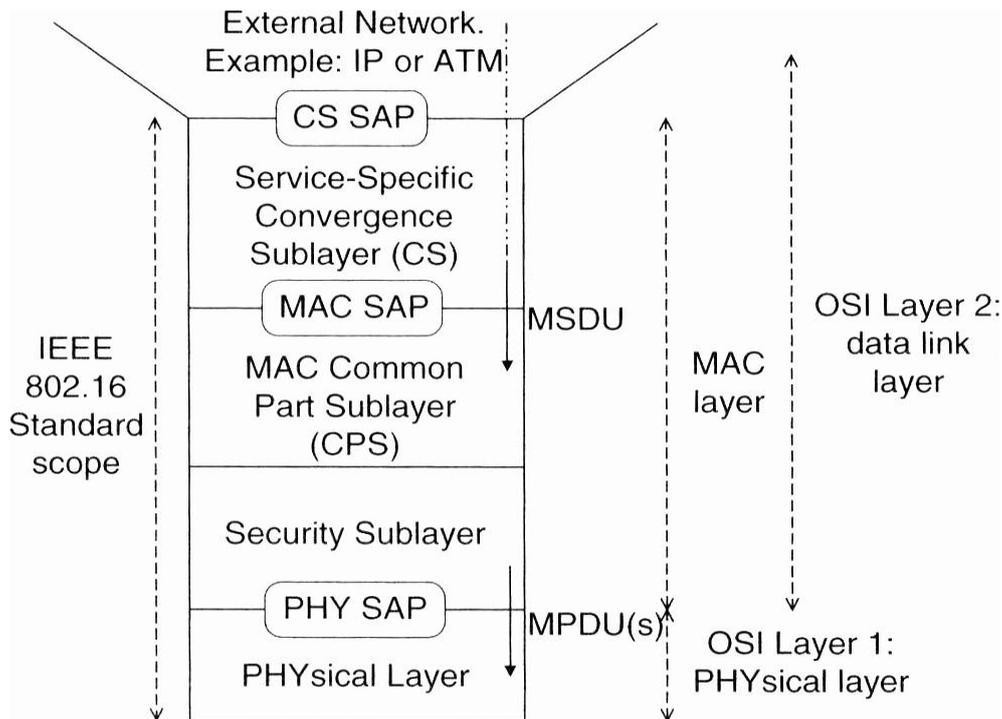


Figure II.10 : Structure des couches WiMAX

II.3.2.1 LA SOUS-COUCHE (CS)

La sous couche de convergence (CS), effectue les tâches suivantes :

- ✓ Elle accepte les PDUs qui proviennent de la couche supérieure. Actuellement deux types de couche supérieure sont supportés par le WiMAX, l'ATM et les protocoles IP (IPv4, IPv6).
- ✓ Elle classe et dirige les MSDUs (Mac Service Data Unit) vers les CIDs (Identificateur de connexion) correspondants suivants le QoS demandée.
- ✓ Elle effectue une opération optionnelle appelée PHS (Payload Header Suppression), qui consiste à supprimer les entêtes répétitifs des PDUs de la couche supérieure, qui seront restitués à la réception.

La classification effectuée par la couche CS est basée sur deux concepts fondamentaux:

Une connexion: désigne une connexion unidirectionnelle au niveau MAC entre un BS et un SS (MS). Une connexion est identifiée par un CID (Connection Identifier sur 16bits) et concerne un seul type de QoS (la voix et le mail ne peuvent pas partager le même CID).

Un flux de service (SF) : est un service de transport unidirectionnel de paquet. Il est identifié par un SFID (Service Flow Identifier 32bits). Le SF définit les paramètres QoS pour les PDUs du MAC qui seront envoyés durant la connexion. Un SF passe par trois états (Provision, Admission, Activation), Seul l'état actif peut générer une connexion (CID).

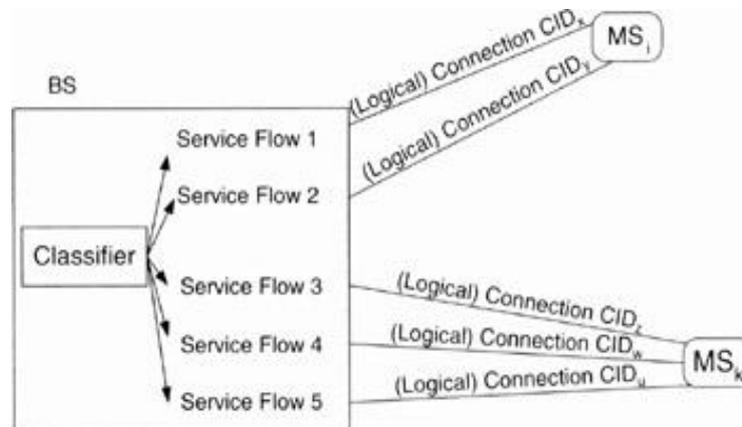


Figure II.11 Connexion et flux de service dans le WiMAX

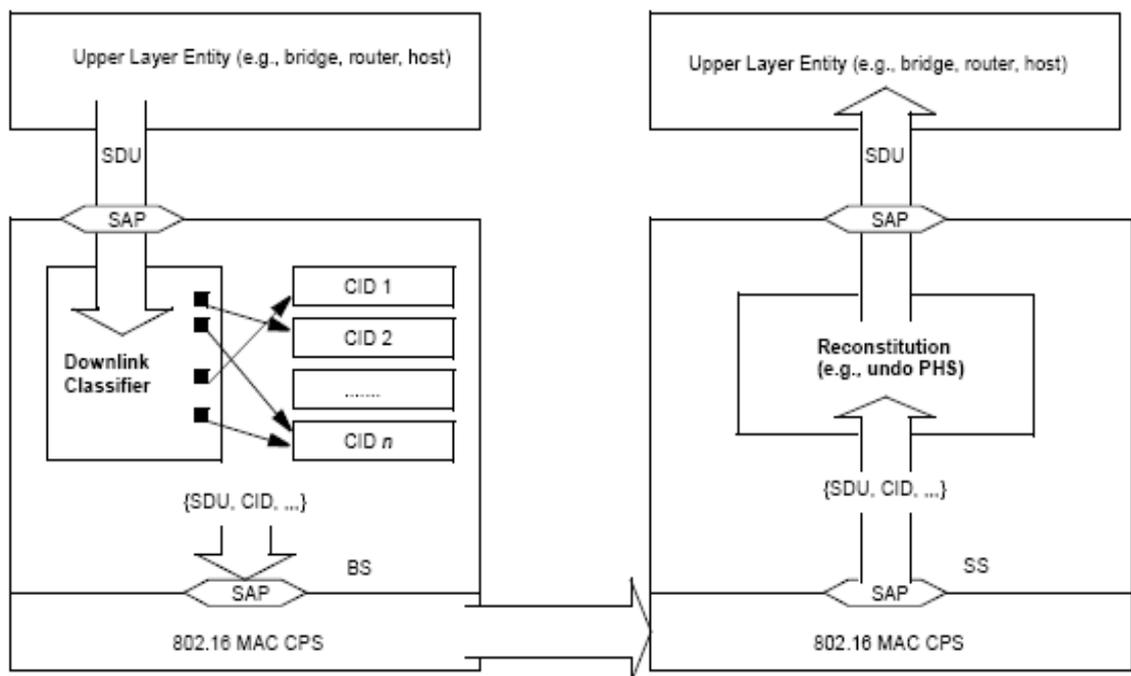


Figure II.12 Flux de données dans les différentes couches MAC

Durant la création d'un SF, la couche CS définit les spécifications QoS suivant le type de protocole de la couche supérieure. Les paramètres QoS décrivent plusieurs options MAC parmi lesquelles le type de service de planification, ou classe QoS. Le standard 802.16-2004 définit quatre classes (BE, nrPS, rtPS et UGS), la version 802.16e a ajouté le ertPS. Le tableau suivant décrit les détails associés à chaque classe :

Classes	Paramètres
Unsolicited Grant Service (UGS)	Débit maximum, Délais maximum toléré, Jitter(variation dans le délai) maximum toléré
Real-Time Polling Service (rtPS)	Débit Maximum Débit Minimum Taille garantie du Burst Délais maximum toléré, etc
Extended Real-Time Polling Service (ertPS)	Combine UGS et rtPS.
Non-real-time Polling Service (nrtPS)	Débit Garanti, Débit Maximum.
Best Effort (BE)	Débit maximum

Tableau II-6 : Les différentes classes du QoS dans le WiMAX

Unsolicited Grant Service (UGS): Cette classe de service est utilisée pour transmettre des flux temps réels. La transmission doit s'effectuer avec des trames de taille fixe à intervalle régulier. Comme exemple d'application, on peut citer la transmission de voix sans suppression de silence et l'émulation de liaison spécialisée T1/E1 (liaison commutation de circuit à 2Mbps).

Real-time Polling Services (rtPS) : Cette deuxième classe de service permet la transmission de flux temps réels avec des paquets de taille variable et à intervalle régulier. Son utilisation convient très bien pour la transmission de vidéo MPEG.

Extended real-time variable rate Polling Service (ertPS) : Cette classe de service est utilisée pour les applications temps réels ayant des exigences entre UGS et rtPS comme la transmission de voix avec suppression de silence.

Non-Real-time Polling Services : Cette classe de service permet la transmission de flux qui tolèrent des délais. De plus ces flux contiennent des trames de taille variable. Seul le taux de transfert minimum est garanti. Ce type de qualité de service convient très bien aux transferts de fichiers : FTP.

Best effort : Ce service ne donne aucune garantie sur l'acheminement des flux de données. Il convient tout de même à certaines utilisations, par exemple la navigation sur internet.

II.3.2.2 LA SOUS-COUCHE MAC COMMUNE (MAC CPS)

La sous-couche CPS représente le corps et le cœur de la couche MAC du WiMAX et est responsable de:

- ✓ L'allocation de bande passante.
- ✓ Etablissement de connexions.
- ✓ Maintenance de la connexion.

Ces fonctions sont gérées par l'utilisation de connexions de management (CM). Il existe trois types de CM.

Le premier est le « Basic Management Connection ». Il est généralement utilisé par la station de base ou la station réceptrice pour envoyer des messages de management MAC court ou urgent.

Le deuxième type correspond aux messages de management de connexion primaire « Primary Management Connection ». Ils sont utilisés par la station de base et la station réceptrice afin de transmettre des informations plus longues et plus tolérantes en termes de délai.

Le « Secondary Management Connection » est utilisé par la station de base et la station réceptrice afin de transmettre des messages standards. Ces messages incluent des messages de type : DHCP, SNMP...

Les trames MAC sont gérées principalement par la sous couche CPS. La trame se compose de trois parties. La première partie correspond à l'en-tête de la trame donnant les informations principales sur le transfert des informations. Le payload correspond aux données transportées par la trame. Et enfin, le CRC, celui-ci est facultatif, il permet au récepteur de la trame de vérifier l'intégrité de la trame.



Figure II.13 : Une trame MAC WiMAX

Il existe deux types d'en-têtes pour les trames WiMAX :

HT	EC	TYPE	RS	CI	EKS	RS	LEN	CID	HCS
1	1	6	V 1	1	2	V 1	11	16	8

HT	EC	TYPE	BR	CID	HCS
1	1	6	19	16	8

HT : Header Type = 1

EC : Encryption Control = 0

RSV : Reserved= 0 ou 1

CI : CRC indicator= 0 ou 1

EKS : EncryptionKeySequence

LEN : Lenght (en-tête+payload+CRC)

CID : Connection Identifier

HSC : Header Check Sequence

BR : Bandwith Request

Figure II.14 : Les deux types d'entête MAC du WiMAX

Comme on peut le voir, plusieurs champs sont communs. HT nous donne le type d'en-tête, on peut ainsi définir l'entête d'une trame générique (HT = 0) et l'entête d'une trame sans Payload. EC informe sur le cryptage de la trame, si le champ est nul alors la trame n'est pas cryptée. RSV définit une réservation. LEN donne la longueur du paquet y compris l'en-tête et le CRC s'il est présent. EKS est l'index de la clé de cryptage. Le payload constitue la partie centrale de la trame WiMAX. Le CID donne l'identifiant de connexion. Le paramètre BR contient les informations de requête de bande passante pour le lien montant.

Les trames MAC peuvent subir les opérations de mise en forme suivantes :

- ✓ Fragmentation : Une trame MAC est fragmentée en plusieurs trames pour adapter l'émission à un mauvais canal.

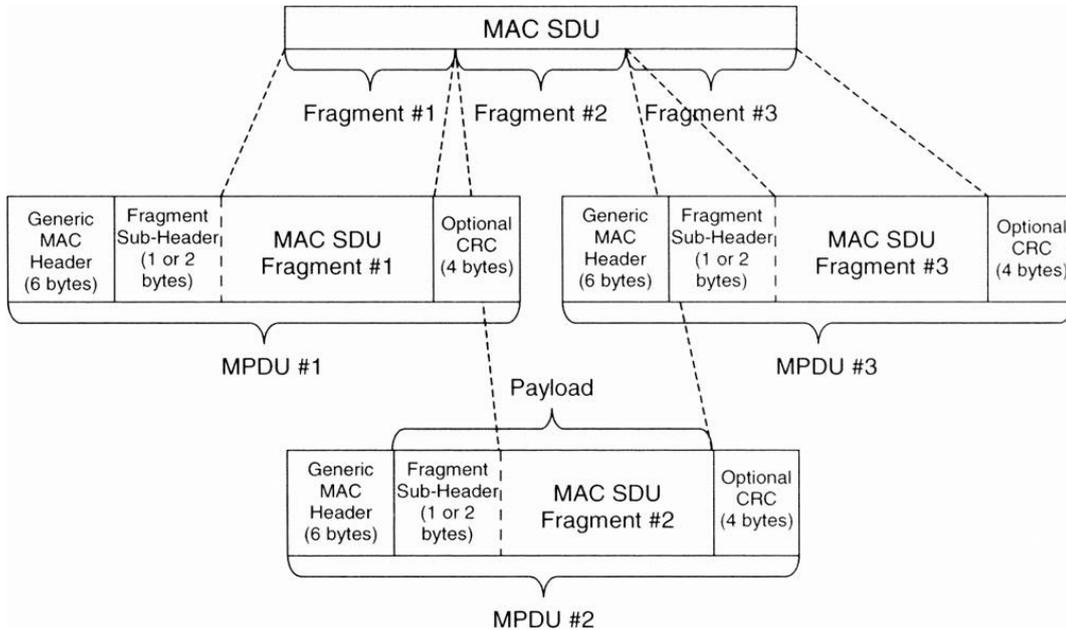
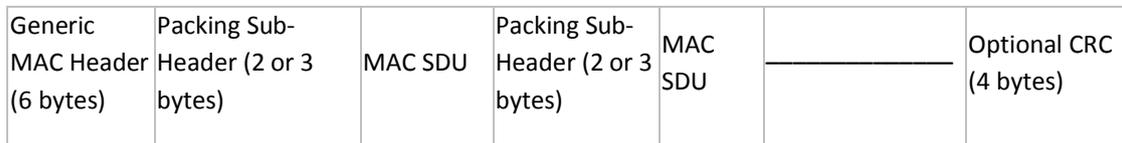


Figure II.15 : La Fragmentation de la trame MAC du WiMAX

- ✓ Paquetage : Plusieurs trames MAC sont paquetées dans une trame unique pour bénéficier d'un bon canal radio.



- ✓ Concaténation : C'est similaire au paquetage, seulement les trames concaténées n'appartiennent pas aux mêmes CIDs, ainsi ils ne sont pas mélangés dans une seule trame.

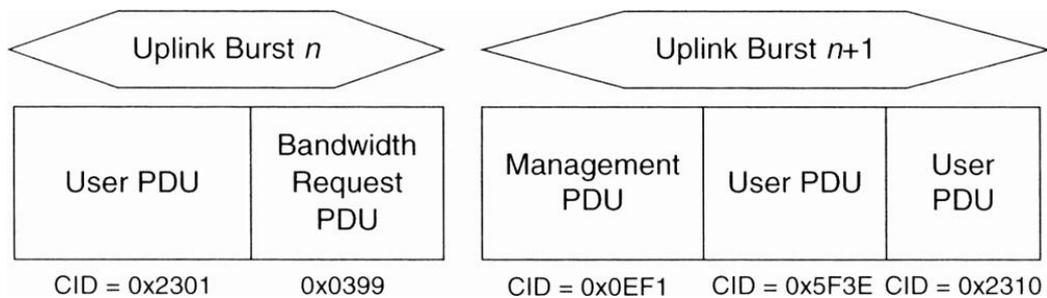


Figure II.16 : La Concaténation de la trame MAC du WiMAX

II.3.2.3 LA SOUS-COUCHE DE SECURITE

Dans un premier temps la station cliente (SS) envoie une demande de connexion en envoyant ses certificats ainsi que ses paramètres de cryptage. Si le client a les droits nécessaires, la BS autorise l'accès au réseau et envoie un acquittement crypté par la clé publique du client. La BS s'authentifie alors devant le client en envoyant une association de sécurité et ses paramètres de chiffrement. Le client vérifie l'identité de la BS puis s'enregistre sur le réseau, il reçoit par la suite un acquittement crypté, la connexion est maintenant établie et sécurisée.

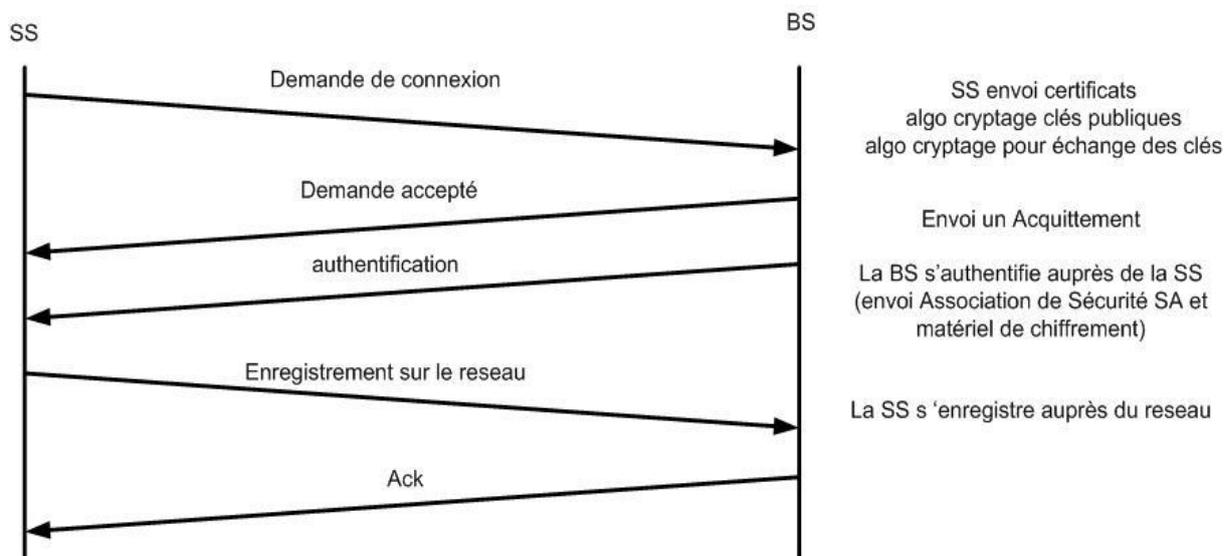


Figure II.17 : Etablissement de connexion sécurisée WiMAX entre une BS et une station cliente

II.3.2.4 LA COUCHE PHYSIQUE

Comme toutes les technologies utilisant la voie hertzienne le WiMAX est soumis aux interférences du milieu dans lequel les ondes se propagent. Plusieurs techniques permettent d'atténuer la perturbation du signal par le bruit environnant. Ces techniques ont évolué avec les versions du WiMAX. On a ainsi plusieurs types de couche physique, résumé dans le tableau suivant.

Désignation	Fréquence	LoS/NLoS	Options	Duplexage
WirelessMAN-SC	10-66Ghz	LoS		TDD,FDD
WirelessMAN-SCa	2,5-11Ghz	NLoS	AAS, ARQ, STC	TDD,FDD
WirelessMAN-OFDM	2,5-11Ghz	NLoS	AAS, ARQ, STC, mesh	TDD,FDD
WirelessMAN-OFDMA	2,5-11Ghz	NLoS	AAS, ARQ, STC	TDD,FDD
Wireless-HUMAN	2,5-11Ghz	NLoS	ARQ, STC, Mesh	TDD

Les options :

- AAS : Antennes adaptives
- ARQ : AutomaticRetransmission Request
- STC : Schéma de diversité
- Mesh: Topologies Mesh

Le duplexage :

- TDD : Time Division Duplex
- FDD : FrequencyDivision Duplex

Tableau II-7 : Les différents types de la couche physique dans le WiMAX.

Comme on peut le constater, différents types de modulation numérique sont possibles :

- ✓ SC : (Simple Carrier) Modulation classique à porteuse unique
- ✓ OFDM : Le principe de l'OFDM (Orthogonal Frequency Division Multiplexing) est la transmission sur plusieurs fréquences orthogonales. Ce principe va diminuer la perception des interférences notamment en Nlos (non line of sight). La modulation OFDM permet de moduler en fréquence jusqu'à 256 sous porteuses, l'accès multiple dans cette couche physique s'effectue en TDMA.

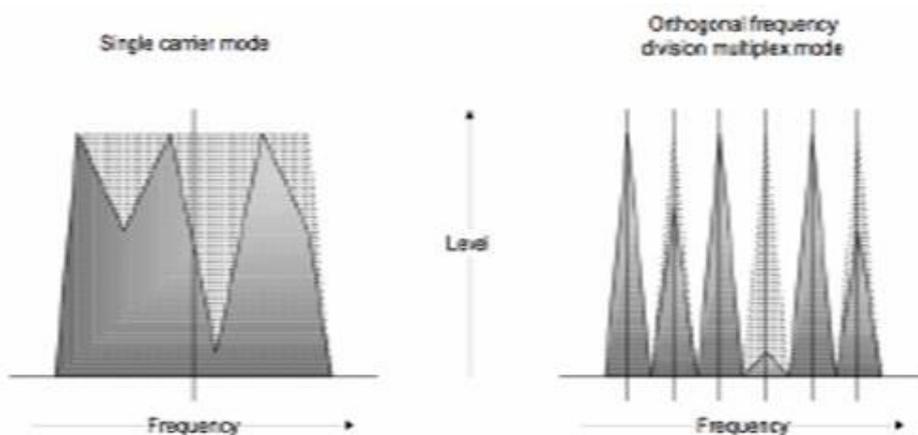


Figure II.18 : Modulation simple porteuse (à gauche) Modulation OFDM (à droite)

- ✓ OFDMA : L'OFDMA (Orthogonal Frequency Division Multiples Accés) reprend le principe de l'OFDM. Dans cette version il est désormais possible d'utiliser jusqu'à 2048 sous porteuses qui cette fois ci, à la différence de l'OFDM, sont attribuées de manière dynamique. En effet si un utilisateur a besoin de trois fois plus de bande passante qu'un autre utilisateur alors la modulation lui allouera 3 emplacements alors que l'utilisateur normal n'en aura qu'un seul.

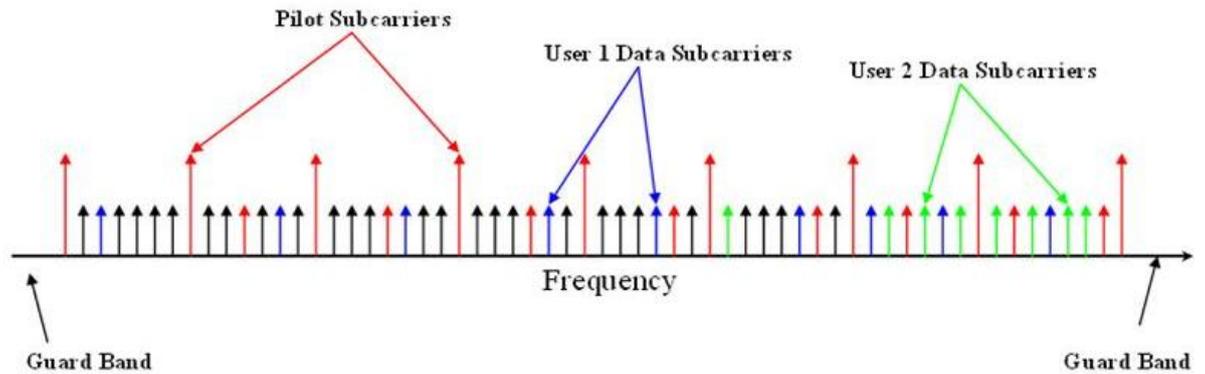


Figure II.19 : La Modulation OFDMA

Le duplexage, ou comment diviser la transmission entre le débit montant et le débit descendant, peut être réalisé soit temporellement soit fréquentiellement :

Le duplexage FDD (Frequency Division Duplex) : Le FDD est un duplexage fréquentiel. La voie montante et descendante utilisent les mêmes intervalles de temps mais avec des fréquences séparées. Le WiMAX utilise le HFDD (half-duplex FDD). Le HFDD mixe les liens pour offrir tour à tour du full et du half duplex.

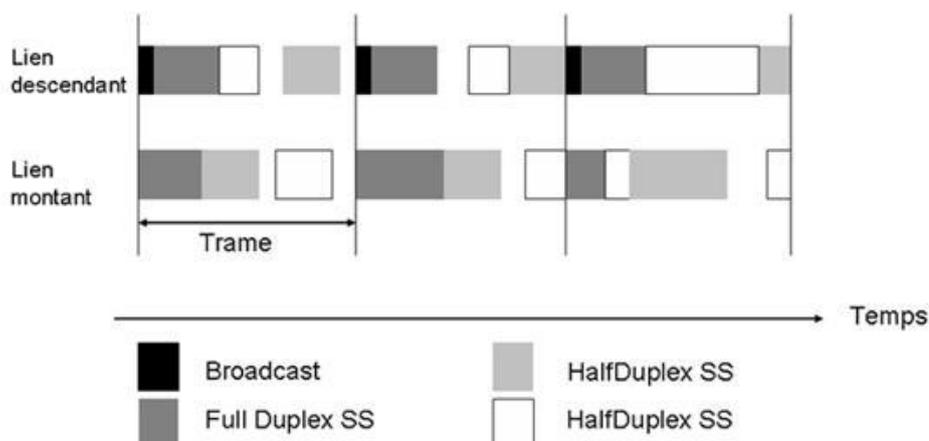


Figure II.20 : Le duplexage FDD

Duplexage TDD (Time Division Duplex) : Multiplexage temporel dans les deux sens de transmission sur une seule fréquence. Les voies montantes et descendantes utilisent à tour de rôle la même fréquence.

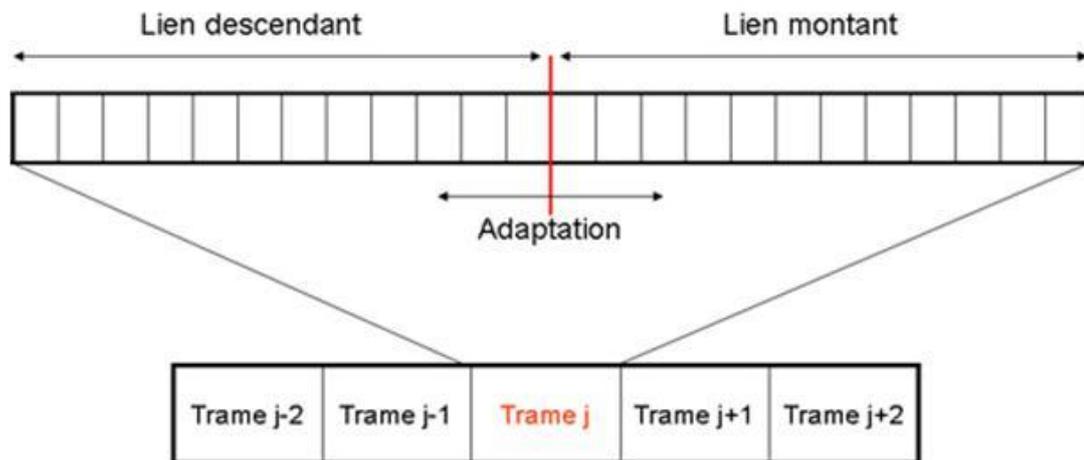


Figure II.21 : Le duplexage TDD

Afin d'avoir des transmissions multiutilisateurs, un système à accès multiple, tel que Time Division Multiple Access (TDMA) ou Frequency Division Multiple Access (FDMA) doit être associé à l'OFDM. En fait, un signal OFDM peut être construit à partir de plusieurs signaux utilisateurs constituant ainsi l'accès multiple OFDMA.

L'accès multiple a une nouvelle dimension avec OFDMA. L'utilisateur de lien descendant ou ascendant aura une allocation temporelle et fréquentielle pour chacune de ses communications. Cependant, les sous-porteuses disponibles peuvent être divisées en plusieurs groupes de sous-porteuses appelée sous-canaux. Ces derniers peuvent être constitués soit en utilisant des sous-porteuses contigus ou distribués pseudo-aléatoirement à travers la bande passante. Les sous-canaux formés en utilisant des sous-porteuses distribuées fournissent une plus grande diversité fréquentielle. Cette permutation peut être représentée par les modes d'utilisation partielle de sous-porteuses (PUSC) et utilisation complète de sous-porteuses (FUSC).

Les transmissions descendante et ascendante cohabitent selon l'un des deux modes : Duplexage temporel (TDD) ou Duplexage fréquentiel (FDD). Cependant, le mode TDD est favorisé par la majorité des implémentations du forum WiMAX. La trame est divisée en deux sous-trames: une sous-trame descendante suivie d'une sous-trame ascendante séparée par petit intervalle de garde. La figure 2.1 montre une trame OFDMA lors du fonctionnement en mode TDD.

La sous-trame descendante commence par un préambule qui est utilisé pour les procédures de la couche physique, tels que la synchronisation temporelle et fréquentielle et l'estimation initial du canal. Le préambule est suivi par une entête de contrôle de trame FCH (Frame Control Header) qui fournit des informations de configuration de la trame, tels que la longueur des messages MAP (Media Access Protocol), type de modulation et de codage et les sous-porteuses utilisables. Des zones de données sont allouées aux différents utilisateurs dans la trame et ces allocations sont spécifiées dans les messages MAP descendants et ascendants (DL-MAP and UL-MAP) qui sont broadcastés après le FCH. Les messages MAP incluent le profile d'envoi pour chaque MS qui définit le type de modulation et de codage utilisé pour chaque lien.

La sous trame ascendante est constituée de plusieurs émissions ascendantes initiée par différents utilisateurs. Une partie de la sous trame ascendante est laissée à part pour un accès basé contention qui est utilisé pour des buts variables. La sous trame ascendante inclut un canal d'indication de qualité du canal utilisé par le MS pour envoyer des informations de feedback à la BS, elle inclut également un canal d'acquiescement utilisé par les MS pour acquiescer les paquets descendants.

Une fois que les données de la couche supérieure ont été classés dans des SFs et planifiées par la couche MAC, un allocateur de slots leurs alloue les slots nécessaires. Un slot étant l'unité de ressource de base dans la structure de slot OFDMA et qui constitué d'une intersection entre une division fréquentielle et une division temporelle du symbole OFDMA.

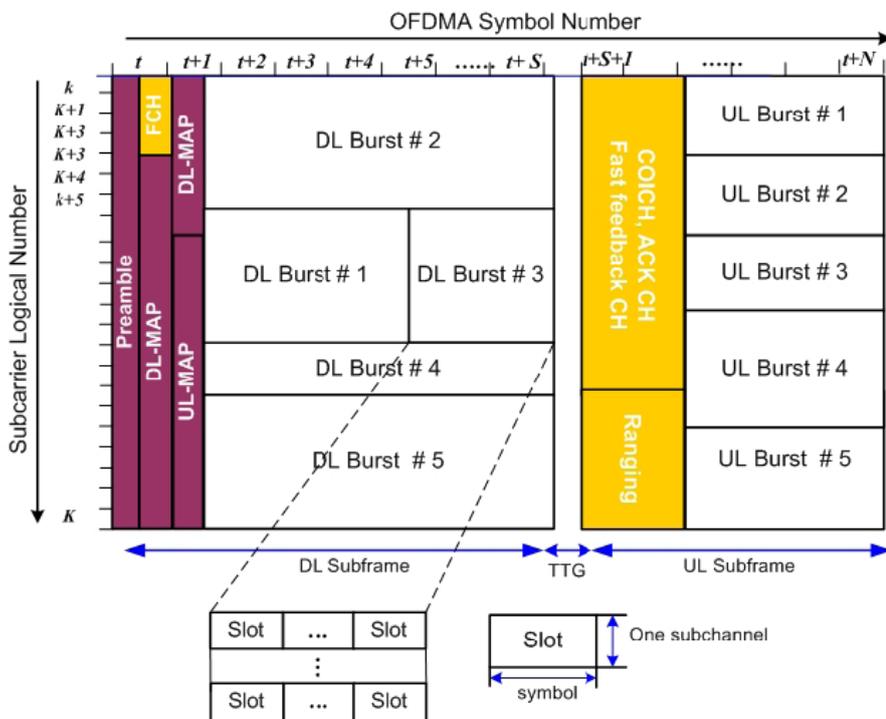


Figure II.22 : Structure d'une trame OFDMA WiMAX

II.3.2.5 TOPOLOGIES WIMAX

La norme d'IEEE 802,16 définit deux topologies possibles de réseau :

PMP (Point-à-Multipoint)

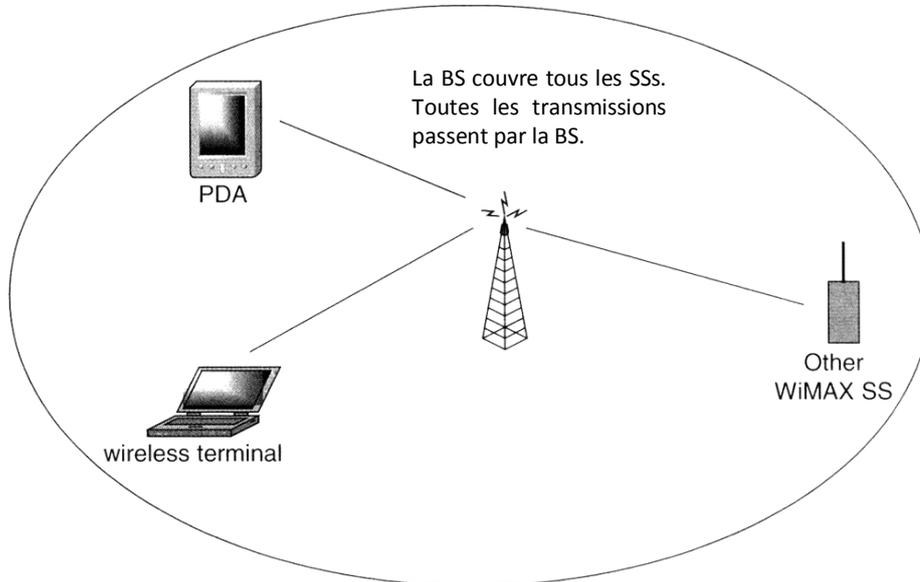


Figure II.23 : Topologie PMP

Mesh (Maille)

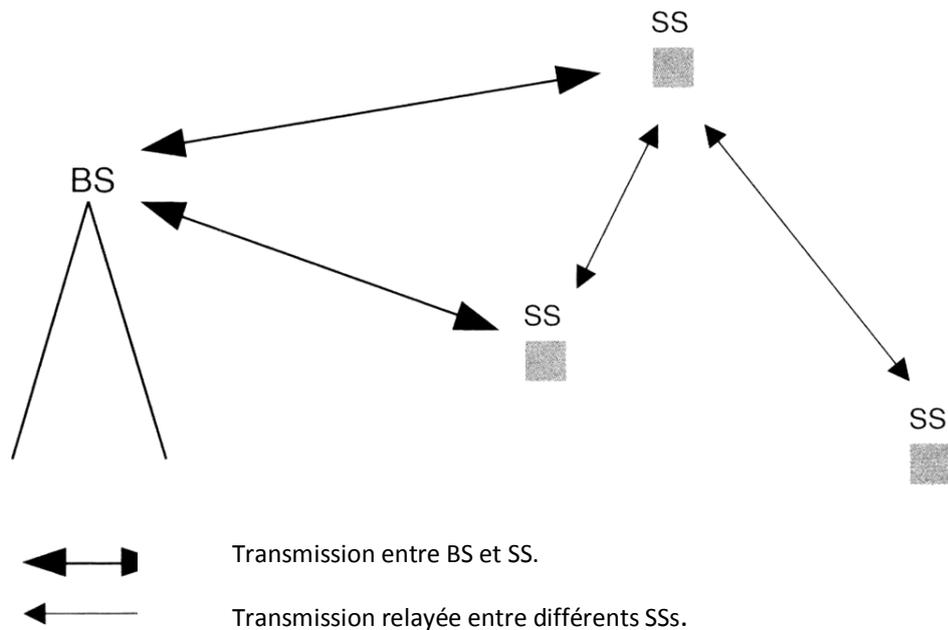


Figure II.24 : Topologie Mesh

La différence principale entre les deux modes est que dans le mode PMP, l'échange de trame ne peut se faire qu'entre une BS et un SS(MS) ; alors que dans le mode Mesh l'échange est possible entre deux SS et même entre un BS et un SS à travers un autre SS. Les éléments d'un réseau en mode Mesh sont appelés Nodes. On peut voir le mode mesh comme l'évolution du mode Adhoc du wifi.

II.3.2.6 ETABLISSEMENT D'UNE CONNEXION

Lorsqu'une station réceptrice souhaite s'initialiser ou lorsqu'elle vient de perdre le signal, elle doit acquérir un canal descendant.

Pour ceci, elle va dans un premier temps vérifier qu'elle ne possède pas de paramètres qu'elle pourrait utiliser pour acquérir un canal. Si elle possède de telles informations, elle teste alors l'acquisition des canaux qu'elle a utilisé précédemment.

Si pour une raison quelconque l'acquisition d'un canal ne peut pas s'effectuer, la station réceptrice va continuer à scanner le réseau. Elle s'arrêtera uniquement lorsqu'elle aura trouvé un canal valide.

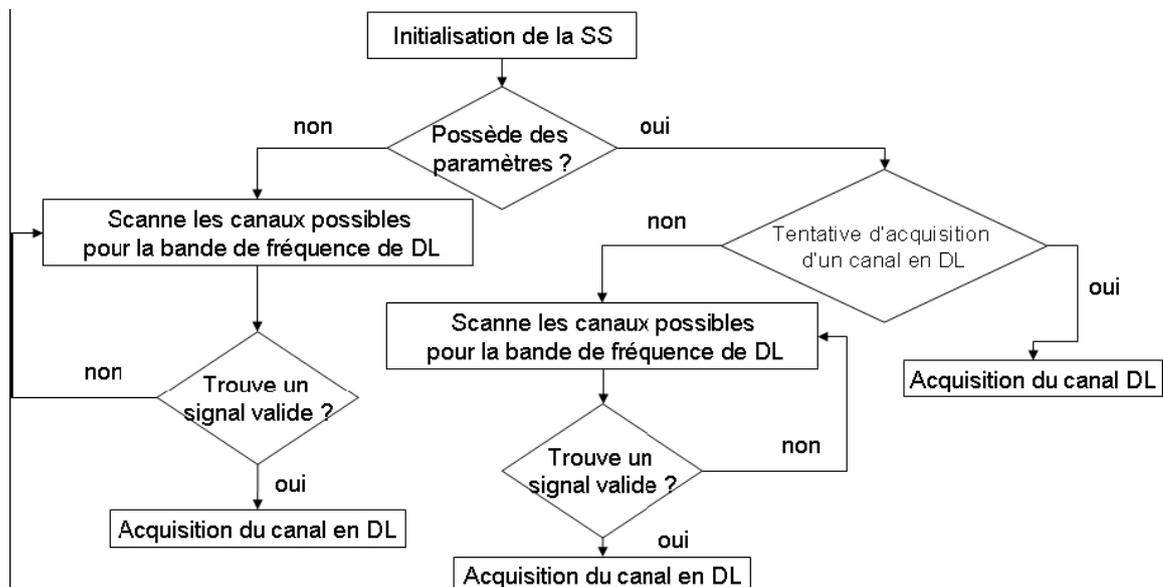


Figure II.25 : Algorithme pour l'établissement d'une connexion

Cet algorithme reprend les différentes étapes évoquées précédemment. Ainsi, nous pouvons voir l'enchaînement des différentes actions à effectuer pour permettre l'initialisation de la station réceptrice.

Rappelons que l'initialisation de la station réceptrice consiste à acquérir un canal descendant.

La synchronisation entre la station de base et la station réceptrice s'effectue grâce à l'acquisition du préambule. Celui-ci permettra par la suite la lecture du FCH (Frame Control Header) qui fournit les informations nécessaires pour décoder le profil du lien descendant.

La synchronisation est effectuée dès que la station réceptrice reçoit son premier profil du lien descendant. Par la suite, elle sera maintenue aussi longtemps que la station réceptrice reçoit des profils du lien descendant. Pour ceci, la station réceptrice observe le broadcast périodique (DCD Downlink Channel Descriptor) émis par la station de base.

La station réceptrice doit trouver un canal montant utilisable. Pour ceci, elle attend un message UCD. Ce type de message est transmis périodiquement par la station de base. Le message UCD contient tous les canaux d'uplink disponibles.

Si la station réceptrice n'a pas trouvé de canal disponible durant un certain temps, elle devra scanner le réseau afin de trouver un autre canal descendant.

Cependant si la station réceptrice a reçu un UCD, elle va en extraire les paramètres d'uplink. Elle va ensuite attendre qu'un profil lui soit fourni pour ce canal. Elle pourra ensuite l'utiliser.

Comme dans la plupart des réseaux, les capacités d'émission et de réception diffèrent entre les équipements. Les équipements doivent pouvoir s'adapter.

Les paramètres transmis dans la transmission initiale sont contenus dans le profil du lien montant. Par la suite, ces informations seront également présentes dans chaque trame.

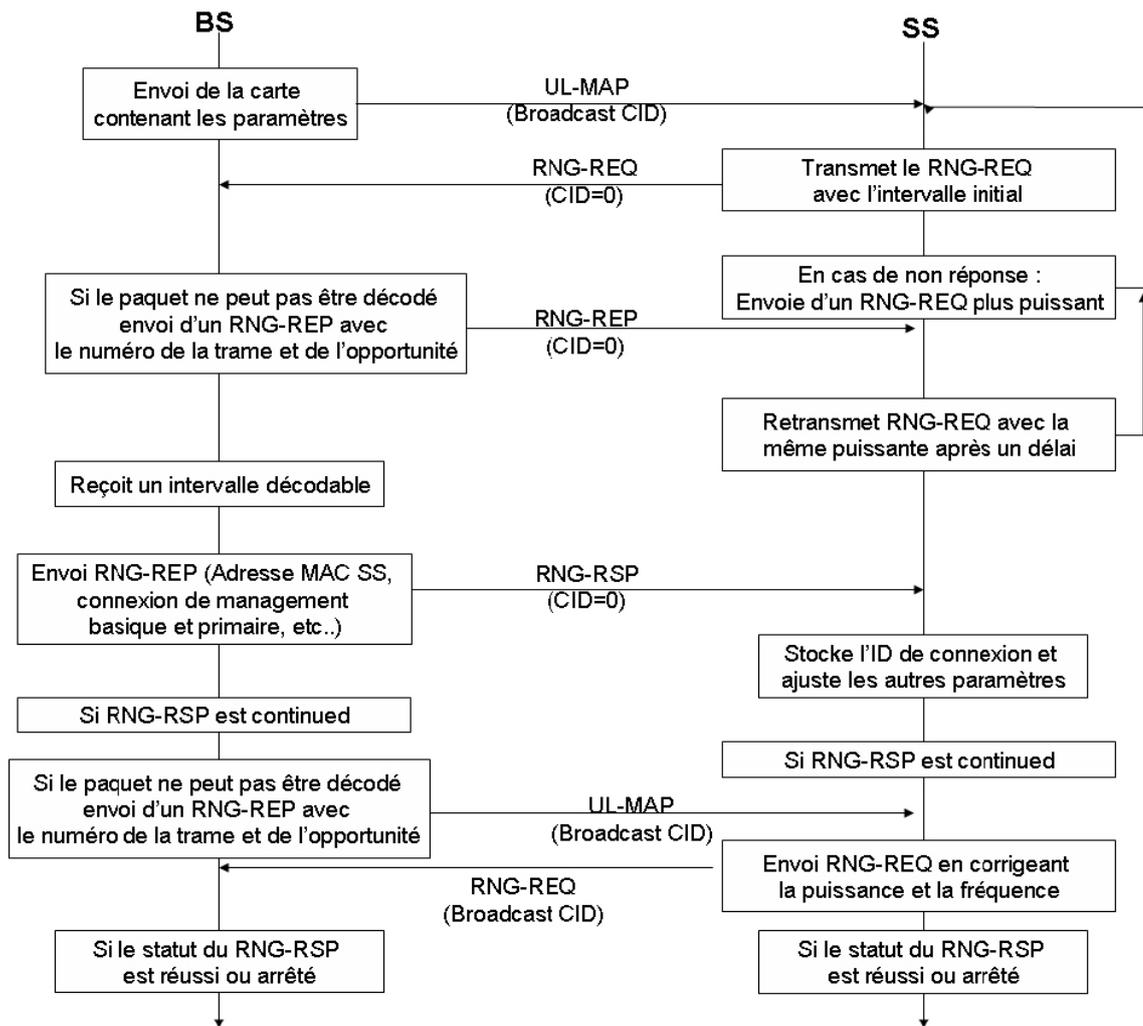


Figure II.26 : Algorithme d'adaptation

La station émettrice va choisir une puissance d'émission. Si elle n'obtient pas de réponse, elle va augmenter cette puissance jusqu'à obtenir une réponse de la station de base.

Il est possible que la station de base ne soit pas en mesure de décoder certaines informations. Dans ce cas, elle prévient la station réceptrice qui va retransmettre le message sans modifier sa puissance d'émission. Cette retransmission est possible puisque la station de base précise l'identifiant de la trame qu'elle n'a pas réussi à décoder.

Lorsque les échanges se sont bien passés, la station de base transmet les différents paramètres à la station réceptrice. Celle-ci pourra ainsi ajuster les siens.

II.3.3 LA GESTION DE LA QoS DANS LE RESEAU WIMAX

Le but ici est de fournir un résumé pour les questions les plus importantes liées à la Qualité de Service (QoS) dans le WiMAX. En effet, malgré le fait que le WiMAX prend en compte le support de la QoS, le contrôle d'admission de communication CAC et la planification de ressource est encore un sujet ouvert. La figure 2.24 suivante représente l'architecture de la gestion de QoS dans le WiMAX, avec les parties en pointillés représentant les fonctions non précisées dans la norme IEEE802.16 (laissées aux constructeurs).

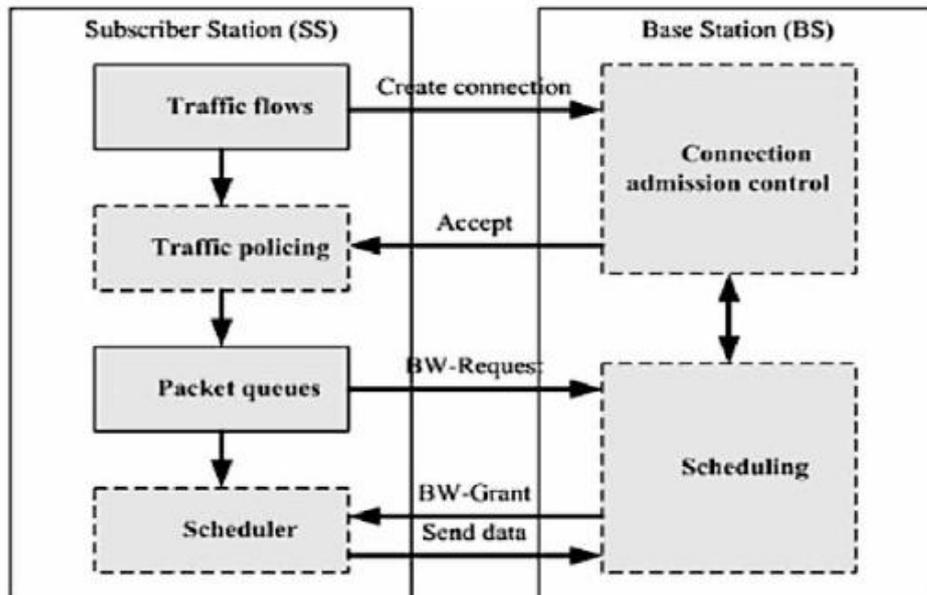


Figure II.27 : Architecture de la gestion de la QoS dans le WiMAX.

Plusieurs solutions au problème de la gestion de QoS dans le WiMAX sont proposées dans la littérature [8] [9] [10][11]. La plupart d'entre eux se concentrent sur la planification de ressource alors que peu d'entre eux traitent le problème d'admission CAC. Ceci est considéré quelque part comme étant irréaliste, du fait que les deux mécanismes sont nécessaires pour une quelconque gestion de la QoS digne de ce nom.

En général, les méthodes de planification de ressources sont composées d'une hiérarchie de planificateurs classiques bien connue, comme le planificateur « la plus proche échéance en premier » ou EDF (Earliest Deadlin First) et le planificateur « File d'attente pondérée équitable » WFQ (Weighted Fair Queuing). Dans ces méthodes, les solutions les plus simples sont favorisées pour pouvoir implémenter la gestion de QoS en temps réel (trame par trame).

La planification de ressource est constituée de deux procédures :

- ✓ L'allocation de ressources au niveau de la couche physique.
- ✓ L'ordonnancement des trames dans les files d'attente au niveau de la couche MAC.

Ces deux procédures peuvent être envisagées séparément comme ça a été souvent le cas, mais ils peuvent être pris en considération simultanément dans des solutions dites « inter-couche » (Cross layer). Ces derniers semblent prendre plus d'intérêt dans les travaux récents.

La gestion du CAC utilise généralement des approches analytiques, où souvent la bande passante disponible est estimée pour pouvoir décider si un nouveau SF peut être admis ou pas. De telles méthodes peuvent garantir une bande passante non saturée par contre ils ne peuvent pas garantir un délai satisfaisant.

Une approche indirecte pour l'amélioration de la gestion de la QoS serait l'augmentation de l'efficacité spectrale de l'interface sans fil, car ceci implique l'augmentation de la capacité totale du système de transmission. Ceci est faisable en exploitant la diversité multiutilisateur et en utilisant la stratégie de codage AMC (Adaptive Modulation and Coding) qui s'adapte aux conditions du canal. Mais souvent cette solution n'est pas efficace, car si on augmente la bande passante, le nombre d'utilisateurs augmentent et on retombe sur le même problème, d'où la nécessité d'un CAC.

Généralement, il est impossible d'avoir une solution optimale pour l'efficacité spectrale, la QoS et l'équité simultanément, car ces objectifs sont souvent contradictoires [12]. Ainsi un compromis effectif est nécessaire pour fixer les objectifs d'une stratégie de gestion de ressource.

Nous pouvons formuler l'allocation des ressources en OFDMA comme étant un problème d'optimisation à contraintes qui peut être classifié comme suit :

Soit en minimisant la puissance de transmission totale avec une contrainte sur le taux de données utilisateur.

Soit en maximisant le débit de données total avec une contrainte sur la puissance de transmission totale.

Le premier objectif est approprié pour les applications à débit fixe, telle que la voix, tandis que la seconde est plus appropriée pour des applications impulsive telles que les données et autres applications IP.

Dans un environnement multiutilisateur, un bon modèle pour l'allocation des ressources s'appuie sur la diversité multiutilisateur et les conditions du canal.

L'architecture d'un ordonnanceur type de données en lien descendant est représentée sur la figure suivante :

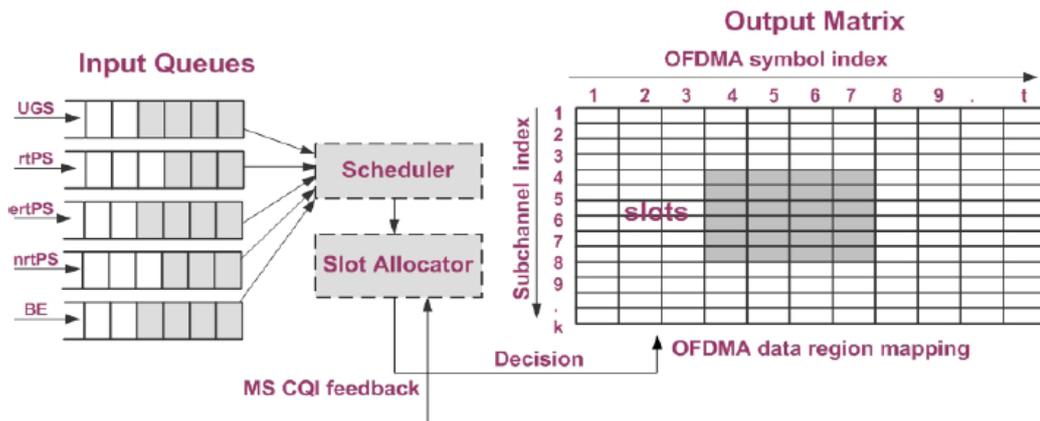


Figure II.28 : Un modèle de planification de ressource dans le WiMAX

Chaque MS peut lui être alloué un ou plusieurs slots conformément aux exigences en QoS de ses applications. Un des avantages de ce modèle est qu'une large étendue de débits peut être prise en charge et il est donc très approprié pour le système WiMAX.

Nous supposons que l'indicateur de qualité de canal CQI (Channel Quality Indicator) de toute la trame et pour chaque utilisateur est parfaitement connu par la BS grâce au message de retour CQICH (Channel Quality Indicator Channel). Le planificateur avec l'aide de l'allocation de slots au niveau de la BS, peut ordonnancer les SFs, attribuer les slots et répartir la puissance sur les slots OFDMA de manière efficace et ceci en exploitant la connaissance des conditions de canal sans fil ainsi que les caractéristiques des SFs.

On peut clairement remarquer que les mécanismes de la gestion de la QoS dans le WiMAX sont plus complets et complexes que ceux du Wifi. Ceci nous pousse donc à stipuler que le passage d'un dispositif mobile d'un réseau Wifi vers un réseau WiMAX est très susceptible de se réaliser sans dégradation de la QoS [13] [14], étant donné bien sur le respect des conditions des limites de saturation du réseau WiMAX. Toutefois, une correspondance entre les critères de gestion de la QoS des deux technologies doit être établit proprement.

Le tableau suivant propose une règle de passage des catégories de QoS Wifi vers des classes de QoS WiMAX :

802.11e (Catégorie d'accès)	802.16e (Flux de Service)
AC_VO (3)	UGS
AC_VI (2)	rtPS
AC_BE (1)	BE
AC_BK (1)	BE

Tableau II-8 : Une correspondance entre les principales classes de QoS dans les réseaux Wifi et WiMAX

Il faut noter que les conditions sur les critères acceptables pour chaque type de flux restent inchangées, par exemple le délai maximum acceptable pour une application vidéo ne va pas changer à cause du passage d'un réseau à un autre.

II.4 CONCLUSION

On s'aperçoit que la gestion de la QoS dans le réseau WiMAX est très bien prise en charge par les standards existants, mais il faudrait juste y intégrer un algorithme de contrôle d'admission pour ne pas dégrader les paramètres de QoS de trafics déjà admis dans le réseau et si en plus c'est des utilisateurs ayant la possibilité de basculer entre plusieurs réseaux hétérogènes sans fil comme le Wifi et le WiMAX, il est tout simplement vital d'intégrer dans les deux réseaux des algorithmes CAC pour surveiller la garantie de la QoS déjà allouée et savoir et décider si le basculement d'un utilisateur est souhaitable ou pas. Les paramètres de QoS d'un flux Wifi seront translatés en paramètres WiMAX selon le tableau 2.8, donc cet aspect est tout à fait faisable comme l'a démontré T. Ali Yahiya dans ses travaux de recherche. Dans le chapitre suivant, on va s'évertuer à vérifier ce point en particulier, c.à.d. la garantie des paramètres de QoS et les méthodes qui peuvent nous aider à optimiser la QoS globale dans une architecture hétérogène Wifi-Wimax.

[

III EXTENSION DE LA QoS WIFI AU WIMAX

III.1 INTRODUCTION

Comme nous l'avons cité dans le premier chapitre, les réseaux sans fil du futur visent à fournir une couverture universelle omniprésente à travers des technologies radio différentes pour des nœuds mobiles multi-accès. Ceci est désormais possible dans une architecture qui intègre de multiples technologies radio B3G interagissant de façon transparente et flexible. Toutefois une telle démarche posera plusieurs défis dont :

- ✓ La sélection de l'interface radio à utiliser.
- ✓ Les mécanismes de Handover transparent.
- ✓ La configuration coordonnée des mécanismes de la QoS.

Une exigence future additionnelle serait le fait que chaque technologie mobile participant à une telle configuration doit être capable de s'auto-adapter aux changements de l'environnement ou du réseau d'accès d'où la nécessité d'une architecture intelligente, traduite par un cadre de travail conçu d'une manière adéquate.

Un premier pas vers ce type de réseau composite est la convergence entre des couples de technologies d'accès. Dans notre cas, le défi qu'on s'est posé est de résoudre et concevoir la convergence efficace et optimale Wifi-WiMAX.

Dans ce chapitre, on va exposer la manière dont on a solutionné le défi et le problème traité dans cette thèse, en nous inspirant de certains travaux de l'état de l'art.

III.2 GENERALITE SUR LA CONVERGENCE WIFI-WIMAX

Divers travaux de recherche en cours traitent le sujet de la convergence. Deux tendances proposées dans la littérature récente ont attiré notre attention. Le premier a été proposé dans [15] et se base principalement sur l'architecture du standard MIH (Media Independent Handover) IEEE 802.21 en ajoutant des éléments supplémentaires. Le deuxième a été proposé dans [16] où deux technologies réseaux ont été couplées au niveau du gestionnaire de l'admission réseau.

Dans la proposition traitée dans [15], le standard IEEE802.21 joue un rôle important. Ce standard qui est en cours d'évolution, définit une structure destinée à améliorer les décisions du Handover des nœuds mobiles en se basant principalement sur les informations des couches inférieures collectés parallèlement des nœuds mobiles et de l'infrastructure d'accès. Ces informations sont définies dans une couche d'abstraction appelé MIH (Media Independent Handover) qui est utilisé pour optimiser le Handover entre des réseaux hétérogènes.

Le cœur de l'architecture IEEE802.21 est le MIHF (Media Independent Handover Functions) qui fournit des services abstraits aux couches supérieures par le biais d'une interface unifiée composée de primitives appelés SAP (Service Access Point) indépendantes de la technologie d'accès. La figure suivante illustre le principe de fonctionnement du MIHF.

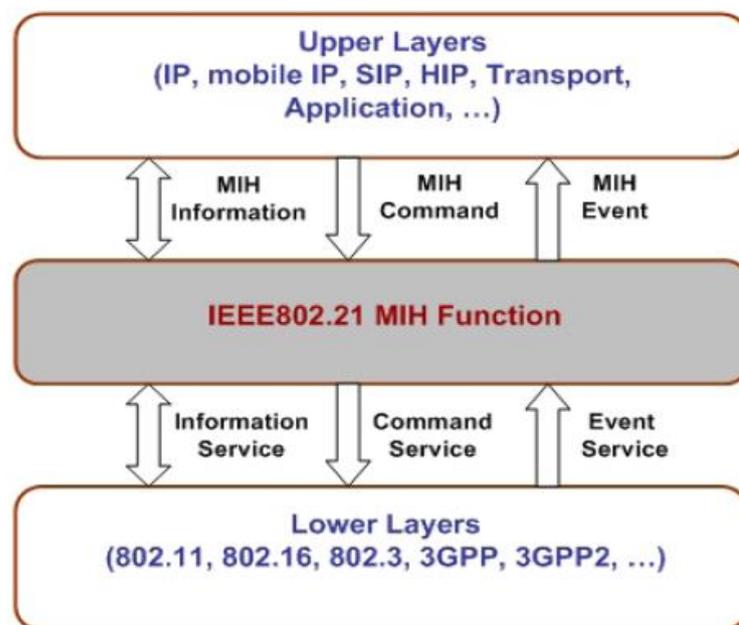


Figure III.1 : Infrastructure IEEE802.21

Les fonctionnalités principales fournies par le MIHF sont:

- ✓ Le MIES (Media Independent Event Service) qui fournit des alertes d'évènements concernant l'état du lien, le déclenchement de Handover, connexion/déconnexion et plusieurs autres changements d'états liés au fonctionnement de la couche MAC.

- ✓ Le MICS (Media Independent Command Service) qui fournit des interfaces de contrôle permettant aux couches supérieures d'exécuter des décisions de mobilité ou de connectivité en influençant directement le fonctionnement des couches inférieures. Ce contrôle peut être local au niveau d'un nœud ou distant entre deux nœuds différents.
- ✓ Le MIIS (Media Independent Information Service) qui fournit des mécanismes de stockage et de distribution de divers types d'informations en utilisant une structure de base de données de gestion matérielle. Ces données sont très utiles lorsqu'un nœud qui est sur le point de changer de réseau a besoin de découvrir les réseaux avoisinants candidats à le recevoir.

En plus de la structure déjà mentionné du MIH, l'idée proposée dans [15] ajoute des fonctionnalités additionnelles comme le montre la figure suivante.

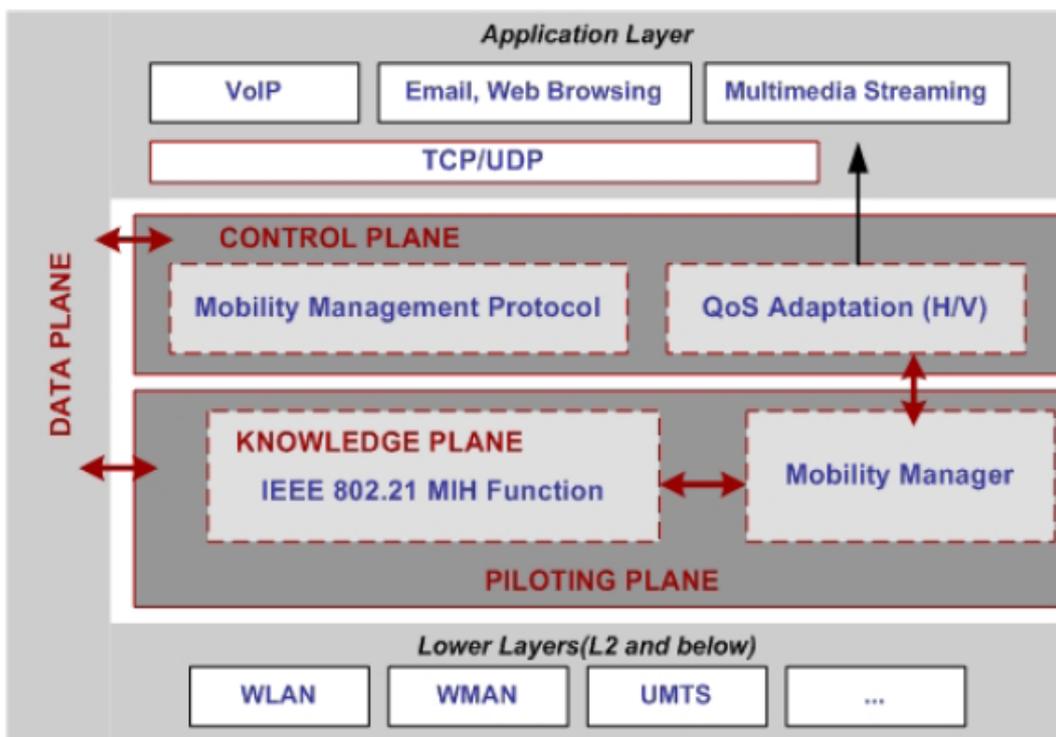


Figure III.2 : Le fonctionnement de l'idée proposée dans [15]

Les éléments ajoutés visent à compléter l'autonomie de l'infrastructure en ajoutant des éléments gérant la collecte des informations nécessaires, la décision de Handover et l'exécution de ce dernier au moment opportun. Dans [15] l'auteur applique son modèle au cas de la convergence WiMAX-Wifi où des procédures de gestion de Handover transparent du WiMAX au Wifi sont proposés et évalués par la simulation. Les critères de décision de Handover sont toutefois laissés ouverts. Pour plus de détail voir [15].

La deuxième tendance dans l'état de l'art [16] intègre un point d'accès dans une infrastructure UMTS réalisant ainsi un couplage étroit entre les deux technologies radios UMTS-Wifi. Sans entrer dans le détail de la technologie UMTS, la figure III.3 illustre l'infrastructure proposée dans [16].

En contraste avec la première tendance, la deuxième se concentre sur la proposition d'une stratégie optimale de décision de Handover sans trop de rajout de modules et d'innovation dans l'infrastructure composite (protocole MIH manquant).

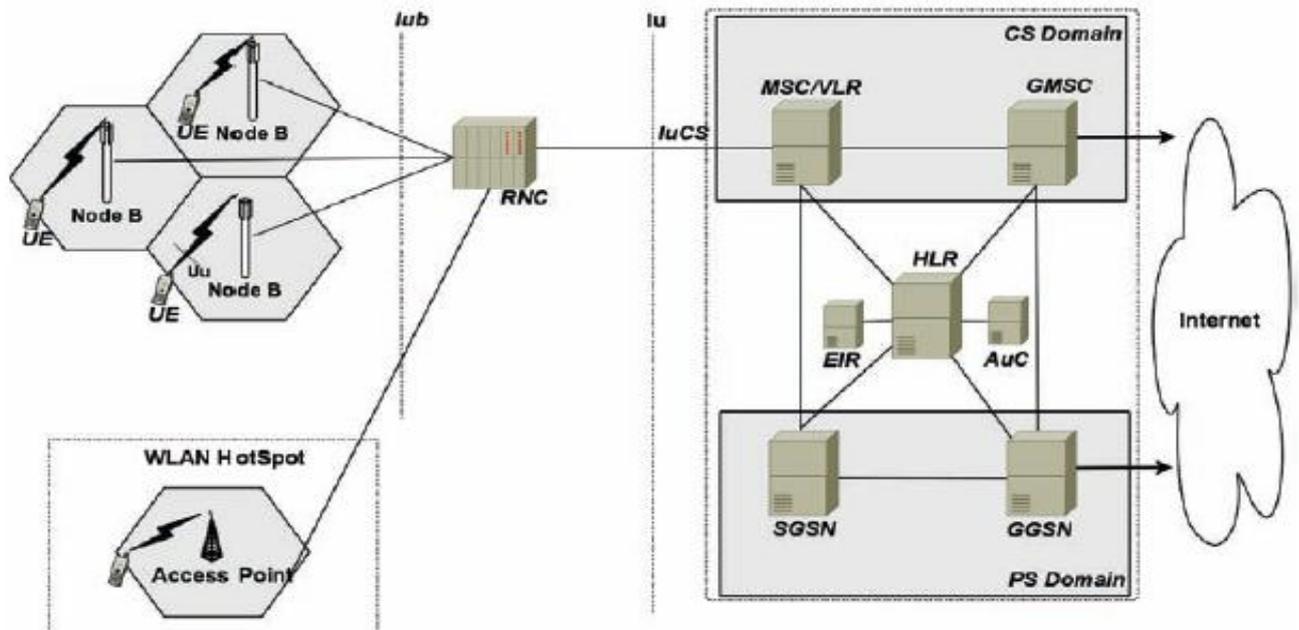


Figure III.3 : Couplage étroit Wifi-UMTS

Le couplage entre les deux technologies a été réalisé au niveau du contrôle d'admission de flux UMTS. Ainsi une station mobile qui intègre les deux interfaces est dirigée vers un accès wifi de préférence. Si le point d'accès devient chargé, le contrôleur d'accès commence à diriger les connexions (décision de Handover) vers la technologie UMTS.

Bien que notre but soit similaire au travail présenté dans [15], c'est-à-dire la convergence Wifi-WiMAX, cependant nous n'utiliserons point la tendance proposée que nous jugeons trop compliquée, donc nous avons opté pour l'utilisation de la deuxième solution plus simple et facilement implémentable sur l'infrastructure déjà existante. Évaluer la stratégie de décision de Handover dans le but spécifique d'étendre la QoS du Wifi au WiMAX, est selon nous beaucoup plus simple à déployer d'un point de vue aussi bien technique que coût. Ainsi, nous optons pour la solution dans laquelle les deux technologies convergent de façon complémentaire dans le but d'améliorer la QoS du réseau global composite résultant.

III.3 NOTRE PROPOSITION

L'environnement de travail pour lequel nous avons opté est basé sur une architecture de convergence étroite. Ainsi nous avons choisi une infrastructure réseau comme suit :

- ✓ 3 groupes de serveurs d'application réseaux filaires Ethernet, qui vont converser avec d'autres clients mobiles utilisant soit le wifi ou le wimax, en générant les trois types de flux temps réel et données (vidéo, voix et données respectivement).
- ✓ Un nœud jouant le rôle d'une passerelle internet. Généralement, dans les réseaux existant, les différents flux convergent tous vers un même point de sortie comme un routeur ou un firewall, d'où le choix d'intégrer cette passerelle dans la topologie.
- ✓ Un nœud (bs-ap) jouant le rôle d'un routeur d'accès composite c.à.d. comportant à la fois une interface de station de base WiMAX et une interface de point d'accès Wifi.
- ✓ Des nœuds jouant le rôle de stations mobiles WiMAX SSs
- ✓ Des nœuds jouant le rôle de stations mobiles Wifi Stas

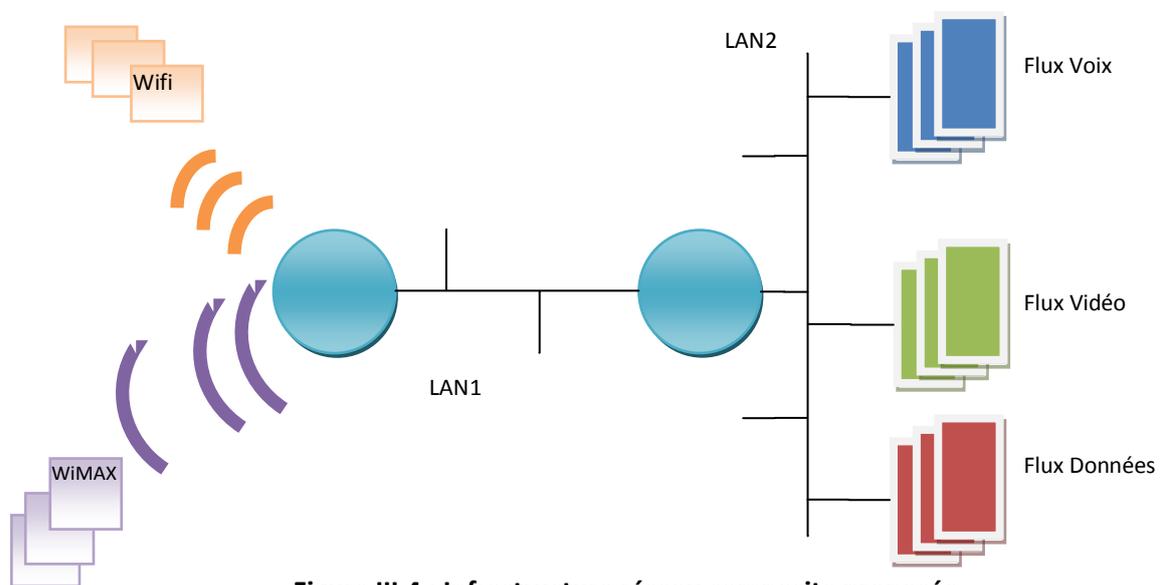


Figure III.4 : Infrastructure réseau composite proposée

La passerelle est connectée avec le nœud bs-ap par un réseau Ethernet LAN1. Les différentes sources de flux sont connectées à la passerelle par un autre réseau LAN Ethernet LAN2. Le routeur bs-ap constitue une cellule WiMAX avec les stations SSs et un réseau infrastructure Wifi avec les stations Wifi.

Notre but étant de gérer la QoS dans un réseau composite Wifi-WiMAX où la gestion de la QoS wifi est étendue lors du passage du réseau Wifi au réseau WiMAX comme une alternative pour éviter la saturation du réseau Wifi ou tout simplement car la station client mobile s'éloigne de l'AP, perd ou risque de perdre sa QoS . Malgré l'aspect général de cette approche, les performances dépendent étroitement du choix du critère de décision de renvoi vers le réseau WiMAX.

L'exploration de la gestion de la QoS afin de fixer les critères de passage Wifi-WiMAX ou WiMAX-Wifi va être réalisée en 3 étapes comme suit :

- ✓ Nous allons commencer par étudier la QoS dans le réseau Wifi uniquement. Le but est d'étudier les performances de gestion de la QoS du point d'accès en fonction du nombre de flux vidéo, voix et données créées et aussi en fonction de la distance entre les stations et le point d'accès.
- ✓ En deuxième lieu, nous allons répéter la démarche de l'étape précédente pour le réseau WiMAX uniquement.
- ✓ En troisième lieu, nous allons coupler la gestion de la QoS Wifi avec la gestion de qualité de service WiMAX comme suit :
 - En fonction du résultat des deux premières étapes, nous allons poser une distance limite D qui va séparer deux zones, une zone proche du routeur bs-ap dénommée $Z1$ et une zone loin du routeur bs-ap dénommée $Z2$.
 - Les stations de la zone $Z1$ sont admises de préférence dans le réseau wifi. Si le réseau Wifi se voit saturé, la station est admise dans le réseau WiMAX. Si le réseau WiMAX est aussi saturé, la station n'est admise ni dans le Wifi ni dans le WiMAX, jusqu'à une décongestion de l'un ou l'autre.
 - Les stations dans la zone $Z2$ sont admises de préférence dans le réseau WiMAX. Si le réseau WiMAX est saturé, les stations sont admises dans Le Wifi si ce dernier est toujours atteignable et non saturé.

La différence principale entre notre idée et celle référencée dans [16] réside dans le fait que la délimitation et définition des zones de prédilection pour telle ou telle technologie, était absente et le réseau Wifi était favorisé au réseau UMTS dans l'absolu. Il faut noter que l'utilisation de la distance comme paramètre indirect de la gestion de la QoS dans le réseau Wifi est inspiré du travail référencé dans [17].

La construction de cette méthode peut sembler triviale mais la définition d'une distance limite qui donne des performances optimales n'en est pas aussi évidente. La proposition de cette distance limite repose sur l'expérience qui a démontré que le réseau Wifi en mode infrastructure permet d'excellents débits et des temps de transmission optimaux, quant les stations sont proches de l'AP (D'ailleurs nous-mêmes en tant qu'utilisateurs du Wifi pour l'accès à internet remarquant bien que la connexion est meilleure quand on est bien disposé, c.à.d. près de l'AP que quand on en est éloigné). En plus du fait que les constructeurs eux-mêmes ont prévus des ajustements de débit (un basculement automatique sur différents débits disponibles se fait en fonction de la portée de l'AP) selon la position de la carte wifi par rapport à l'AP. Le WiMAX étant plus robuste dans les distances lointaines grâce à la combinaison de la méthode OFDMA et MIMO, est donc préféré dans la zone de délimitation deux.

Cette distance limite est un critère simplifié qui est en réalité représentatif de la qualité de la liaison Wifi sensible aux changements des conditions des canaux radios utilisés et aussi aux obstacles présents. Dans ce travail, nous allons étudier seulement le réseau dans des conditions de transmission radios statiques où l'influence de la distance sur la qualité de transmission est statique et stationnaire. Le cas contraire est réservé pour des travaux futurs où une implémentation réelle va être apportée. Pour le moment nous avons opté pour une implémentation et modélisation par la simulation. Cette dernière va nous permettre de révéler ce contraste entre le réseau Wifi et WiMAX et va aussi nous aider à déduire LA distance limite optimale.

Les paramètres de QoS utilisés sont ceux exposés au chapitre deux avec la correspondance qui y est fournie entre les paramètres du wifi avec ceux du wimax.

III.4 CHOIX DU SIMULATEUR

La plupart des logiciels disponibles de simulation des réseaux sont basés sur la simulation à événements discrets DES (Discrete event-based Simulation), où les nœuds du réseau simulé déclenchent des événements. Le simulateur maintient une file d'événements triés suivant le moment planifié pour l'exécution de l'événement. La simulation est elle-même effectuée par le traitement successif d'événements dans la file d'attente. L'approche DES a commencé à être appliquée à la simulation des réseaux informatiques durant les deux dernières décennies. NS-2 est le fruit des premières implémentations et depuis, il est devenu pratiquement la référence pour la simulation des réseaux, ceci peut être attribué au fait que de nombreux modèles de composants réseaux ont été implémentés dans NS-2.

Cependant, une lacune majeure de NS-2 est sa limite d'évolutivité en termes d'utilisation de la mémoire et du temps d'exécution de la simulation. Ceci est particulièrement un problème en ce qui concerne les nouveaux domaines de recherche dans les réseaux informatiques, tels que les réseaux de capteurs sans fil, les réseaux peer-to-peer ou des architectures maillées qui exigent une simulation de réseaux très larges. Pour faire face à ces défis, quelques améliorations de NS-2 ont été proposées, par exemple l'incorporation du parallélisme. Toutefois, NS-2 est actuellement l'objet d'une re-conception majeure. Un des principaux objectifs de développement de son successeur NS-3, est l'amélioration des performances de simulation.

Outre NS-2, plus d'une douzaine de simulateurs des réseaux sont actuellement utilisés dans les universités et l'industrie[18]. Les simulateurs les plus connus comprennent ceux libres de droits comme OMNeT ++, JIST et NS-3, d'autres qui sont commerciaux tels que OpNet ou Omnest et certains simulateurs spécialisés tels que le simulateur de réseaux de capteurs sans-fil TOSSIM. Le simulateur NS3 comme son prédécesseur s'appuie sur le langage C++ pour l'implémentation des modèles de simulation. Toutefois, NS-3 n'utilise pas des scripts OTcl pour contrôler la simulation, abandonnant ainsi les problèmes qui ont été introduits par la combinaison de C++ et OTcl dans NS-2. Ainsi, les simulations des réseaux en NS-3 peuvent être implémentées en pure C++. Par ailleurs, NS-3 intègre des concepts d'architecture et le code d'un simulateur évolutif. Ces décisions de conception ont été faites au détriment de la compatibilité avec NS-2.

Des travaux de comparaison entre différents simulateurs ont été réalisés dans [19] et ont montré que NS-3, OMNeT++ et JIST sont capables d'exécuter des simulations des réseaux à grande échelle d'une manière efficace. JIST était de loin le simulateur le plus rapide, cependant la consommation excessive de la mémoire, limite son applicabilité à quelques scénarios de simulation. Dans cette comparaison de performances, le simulateur NS-3 a démontré les meilleures performances en général. Malgré le fait qu'il est surpassé par JIST en termes de temps d'exécution de simulation, il montre des demandes en mémoire et en temps de calcul plus économiques. Les figures suivantes illustrent quelques résultats de la comparaison réalisée dans [19].

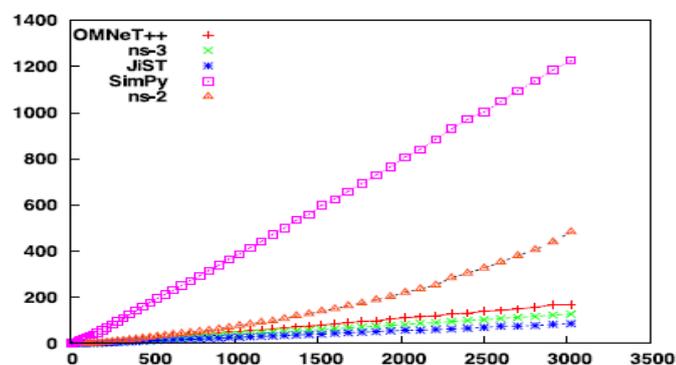


Figure III.5 : Comparaison du temps de calcul en fonction de la taille du réseau.

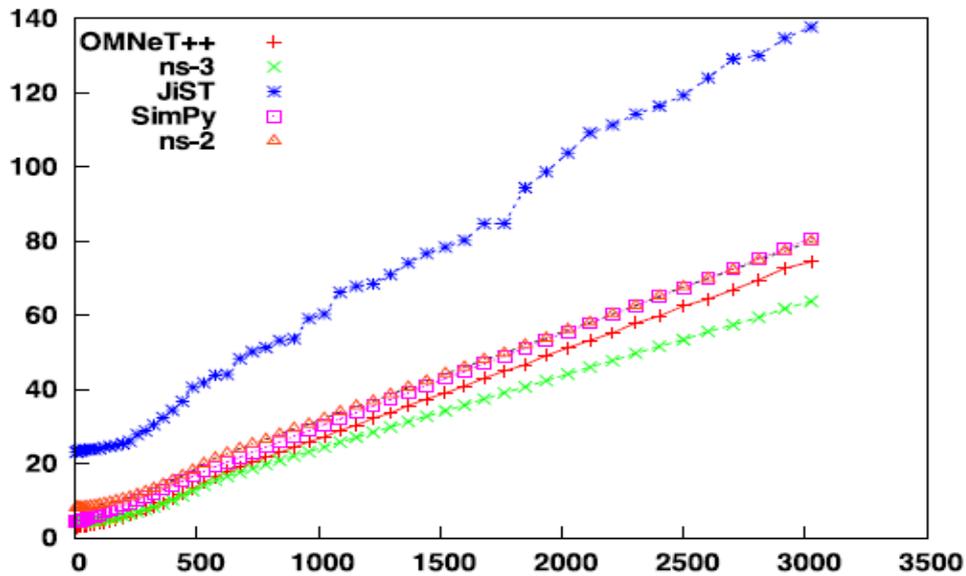


Figure III.6 : Comparaison de l'utilisation de la mémoire en fonction de la taille du réseau.

A nos jours, NS-3 est toujours en cours d'évolution où certains modèles sont encore en cours de développement. Toutefois, le simulateur NS-3 reste un parmi les choix les plus intelligents comme JiST et OMNeT++ le sont.

Ainsi nous avons choisi NS-3 du fait de son potentiel d'expansion qui commence à se matérialiser par le nombre important et croissant de code source contribué par la communauté informatique de NS-3 chaque jour. Une autre raison majeure est le fait de vouloir renforcer l'esprit du logiciel libre et open source dans la communauté scientifique.

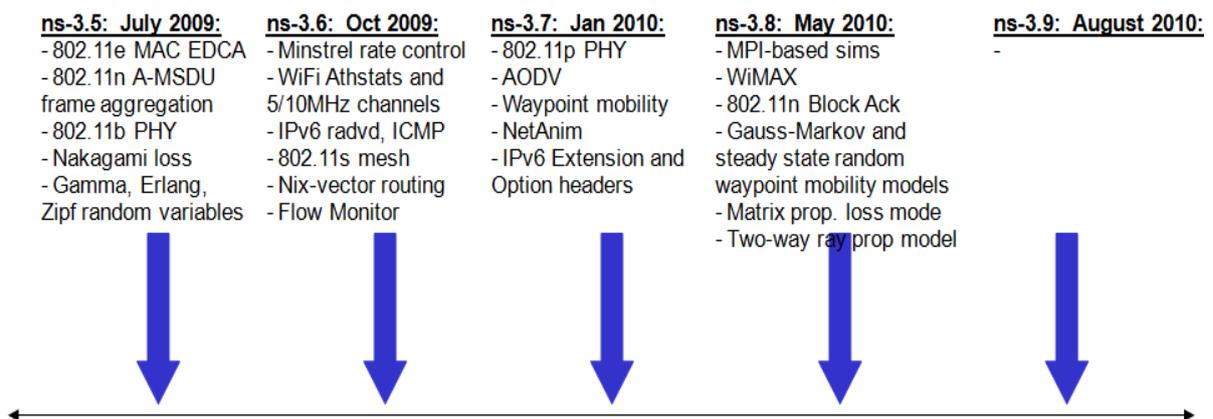


Figure III.7 : Les additions principales dans NS-3 durant la dernière année.

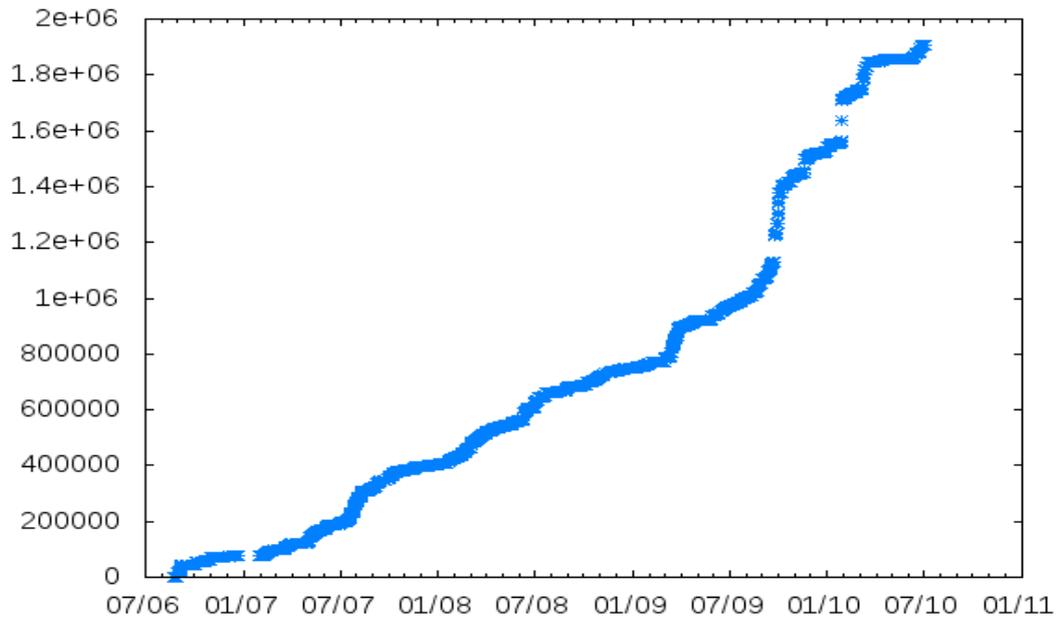


Figure III.8 : La croissance du cumule de nombre de ligne de code C++ (axe y) contribué dans NS-3 dans le temps (axe x).

III.5 SIMULATION NS-3

Le nouveau simulateur de réseau NS-3 est, comme son nom l'indique, le successeur désigné du simulateur populaire NS-2. Son développement a d'abord commencé en 2006, la première version majeure publique et stable a été publiée en Juin 2008[20]. La maintenance et le développement initial sont financés par l'organisme américain NSF (National science Foundation) sous forme d'une subvention de quatre ans.

Initialement les développeurs de NS-3 ont décidé que l'architecture de simulation devait être remaniée complètement, en partant de zéro. Dans cette conception, l'expérience tirée de NS-2 doit être associée avec les progrès des langages de programmation et du génie logiciel. L'idée de la rétrocompatibilité avec NS-2 a été abandonnée dès le départ. Cela libère NS-3 des contraintes héritées de NS-2 et permet la construction d'un simulateur qui est «bien conçu» depuis le début.

Pour permettre une intégration facile de nouveaux modèles dans NS-3, son architecture réseau est inspirée du monde réel sur le plan matériel et logiciel. NS-3 peut également s'interfacer avec le monde réel par le biais de l'émulation. En effet, les paquets créés et utilisés par la simulation sont indiscernables des vrais paquets, ils peuvent ainsi être échangés avec les réseaux réels.

Les composants de la figure 3.9 forment une architecture logicielle en couches. Les concepts de base sont fournis par les modules *Core*, *Common* et *Simulator*. Le module *Helper* contient des interfaces logicielles (API) inter-couche améliorées.

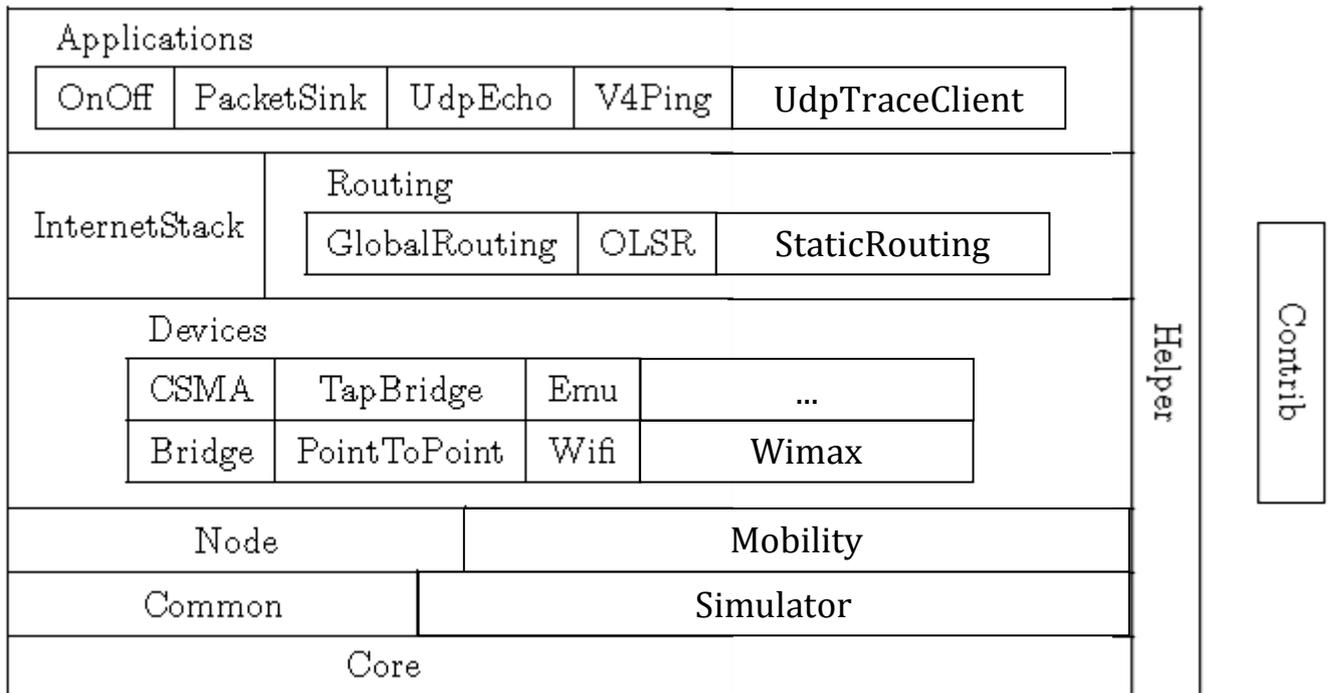


Figure III.9 : Les principaux modules de NS-3

Core: Ce module forme la base de tous les programmes NS-3 par la création d'une structure de classes hiérarchiques dérivant de la classe *Object* de base. Cette hiérarchie d'objets possède de nombreuses fonctionnalités conçues spécialement pour répondre aux besoins de simulation, parmi lesquels sont: Les pointeurs intelligents « PTRs », l'agrégation d'objets, l'enregistrement actif « Typeld » avec les attributs publiques, des interfaces de gestion d'évènement de suivie « Callbacks et TracedCallbacks ».

Common: Une grande partie de ce groupe est centrée autour de la classe *Packet* utilisée pour la simulation de réseau au niveau paquet. L'utilisation de cette dernière classe est conçu de façon à optimiser la gestion de la mémoire en utilisant la méthode « copy on write » et en donnant la possibilité de manipuler des paquets vides et dont la taille est le paramètre suffisant et nécessaire pour la simulation.

Simulator: ce module permet la simulation à événements discrets. Le pilier le plus important du simulateur est la classe *Time* qui représente le temps simulé à haute résolution en utilisant un entier de 128 bits. Par défaut, la graduation de l'échelle temporelle dans NS-3 est de 1 nanoseconde.

Node: Le module *Node* simule un nœud réseau contenant des piles de protocoles. Cette disposition ressemble à celle des systèmes d'exploitation réels. La classe *Node* peut contenir plusieurs *NetDevices*. Chaque *NetDevice* est attaché à un *Channel*, par lequel il envoie et reçoit des paquets. La figure 3.10 illustre cette architecture pour deux nœuds. Le *NetDevice* et le *Channel* demeurent des classes abstraites dans le module *Node*; des implémentations spécifiques se trouvent dans le module *devices*. Un nœud peut contenir plusieurs gestionnaires de protocole, qui acceptent les paquets reçus par le *NetDevices*. Pour lancer la transmission des paquets, chaque nœud peut également contenir une liste d'applications. Encore une fois l'application demeure une classe abstraite à ce niveau.

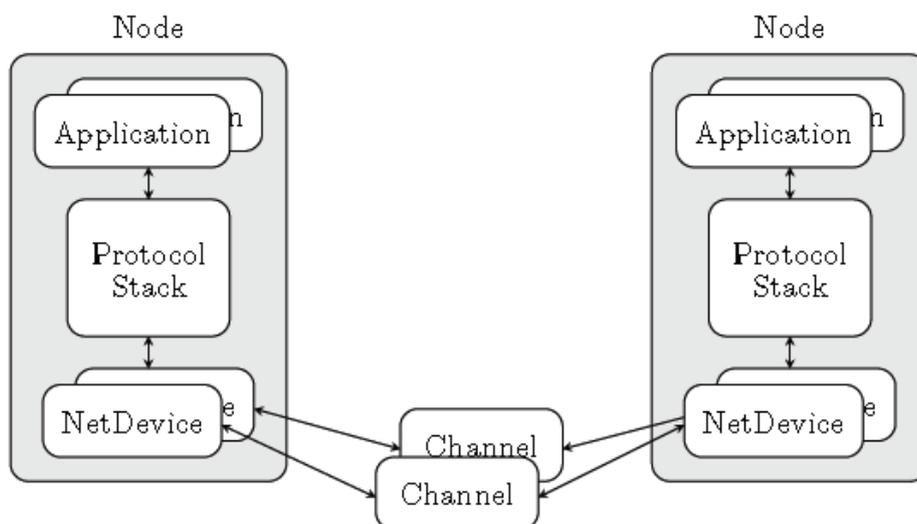


Figure III.10 : L'architecture de nœud NS-3

Mobility: Pour permettre aux nœuds d'être associés à une position, ce module définit une géométrie générale. Pour chaque nœud dans la simulation, un *MobilityModel* peut lui être agrégé, associant une position et un modèle de mouvement au nœud. Actuellement NS-3 contient sept modèles de mobilité.

Devices: Les composants de ce type représente des périphériques réseaux et transmettent des paquets via un canal virtuel à d'autres instances de la même classe *Netdevice*. Comme exemple, on peut citer la classe *CsmaNetDevice* qui implémente un bus CSMA ressemblant à celui d'Ethernet IEEE 802.3. Les paquets sont transmis via un *CsmaChannel* auxquels d'autres *CsmaNetDevices* sont attachés. D'autres exemples de *NetDevices* sont le *WifiNetDevice* et le *WiMAXNetDevice* dont l'implémentation sera détaillée par la suite.

InternetStack: Les classes de ce module définissent les protocoles TCP/IP des couches réseaux trois et quatre (TCP/IP).

Routing: Deux algorithmes de routage sont disponibles dans NS-3. Le premier, appelé *GlobalRouter*, utilise des routes statiques pré calculées en utilisant les calculs de *Dijkstra* SPF à partir des informations recueillies au niveau du lien. Le deuxième met en œuvre le protocole OLSR pour les réseaux dynamiques ad-hoc.

Applications: plusieurs modèles d'applications sont disponibles dans NS-3, Ils sont installés dans les nœuds et peuvent être démarrées/arrêtées à des moments précis dans la simulation.

Helper: L'intérêt des classes du module *Helper* est de rendre plus facile la construction des scénarios complexes de simulation. Comme ce module ne définit pas une couche réseau spécifique, il se tient à l'écart des autres modules.

III.5.1 IMPLEMENTATION DU WIFI DANS NS-3:

Le *WifiNetDevice* modélise la carte réseau sans fil basée sur la norme standard IEEE 802.11. NS-3 fournit des modèles pour les aspects suivant de la norme 802.11 [21]:

- ✓ La fonction DCF 802.11 de base avec les modes infrastructure et ad hoc
- ✓ Les couches physiques 802.11a et 802.11b
- ✓ La fonction EDCA basée QoS
- ✓ Des modèles de propagation de perte divers, y compris Nakagami, Rayleigh, Friis, LogDistance, FixedRss, Random.
- ✓ Deux modèles à délai de propagation, un modèle aléatoire et un modèle basé sur la distance.
- ✓ Des algorithmes de contrôle de débit, y compris AARF, Arf, Cara, Onoe, RRAA, ConstantRate, et Minstrel.

L'ensemble des modèles Wifi NS-3 tentent de fournir une implémentation précise des spécifications MAC 802.11 et un modèle assez rapide au niveau PHY. L'implémentation est modulaire et fournit pratiquement quatre groupes de modèles:

- ✓ les modèles de la couche PHY.
- ✓ les modèles MAC niveau bas qui implémentent DCF et EDCA.
- ✓ les modèles MAC niveau haut qui implémentent la génération de balise MAC, et le sondage.
- ✓ un ensemble d'algorithmes d'auto-adaptation de débit utilisés par les modèles MAC de niveau bas.

Il existe actuellement six modèles MAC de haut niveau, trois pour le MAC sans gestion de la QoS et trois avec gestion de la QoS. Dans chaque groupe (avec et sans QoS) on trouve un modèle *adhocWifiMac*, un modèle *staWifiMac* et un modèle *apWifiMac*.

Avec les modèles MAC QoS, il est possible de travailler avec un trafic appartenant à quatre classes d'accès différentes: AC_VO pour le trafic voix, AC_VI pour le trafic vidéo, AC_BE pour le trafic best-effort et AC_BK pour le trafic d'arrière plan (background). Afin de déterminer la classe d'accès, chaque paquet provenant d'un MAC autre que le Wifi et transmis vers un MAC Wifi doit être marqué au moyen d'un objet *QosTag* afin d'associer un TID (trafic id) pour ce paquet. Dans le cas où un *QosTag* est absent le paquet sera considéré comme appartenant à la classe d'accès AC_BE.

Dans NS-3, les nœuds peuvent avoir de multiples *WifiNetDevices* sur des canaux séparés, et le *WifiNetDevice* peut coexister avec d'autres type de *NetDevice*; ce qui supprime une limitation architecturale trouvé dans NS-2.

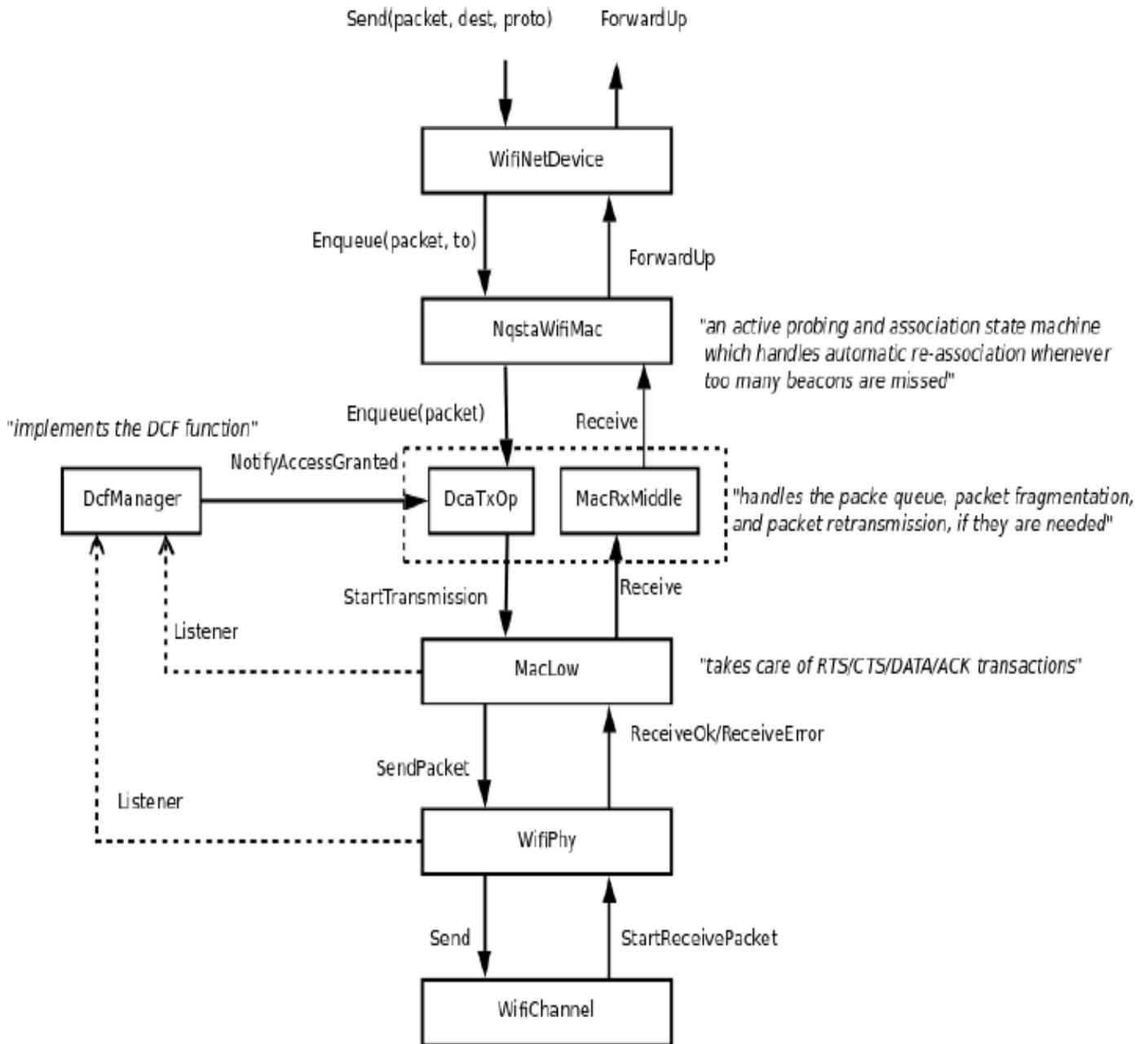


Figure III.11 : L'architecture du *WifiNetDevice* dans NS-3.

La sous-classe *WifiChannel* peut être utilisée pour relier un ensemble d'interfaces réseau *WifiNetDevice*. La classe *WifiPhy* est la partie dans le *WifiNetDevice* qui reçoit les bits du canal. Le *WifiPhy* modélise un canal 802.11 en termes de fréquence, de modulation, de débit binaire et interagit avec le *PropagationLossModel* et *PropagationDelayModel* trouvés dans le canal. La couche PHY peut être dans l'un des trois états: TX (transmission), RX(Réception) ou IDLE(neutre).

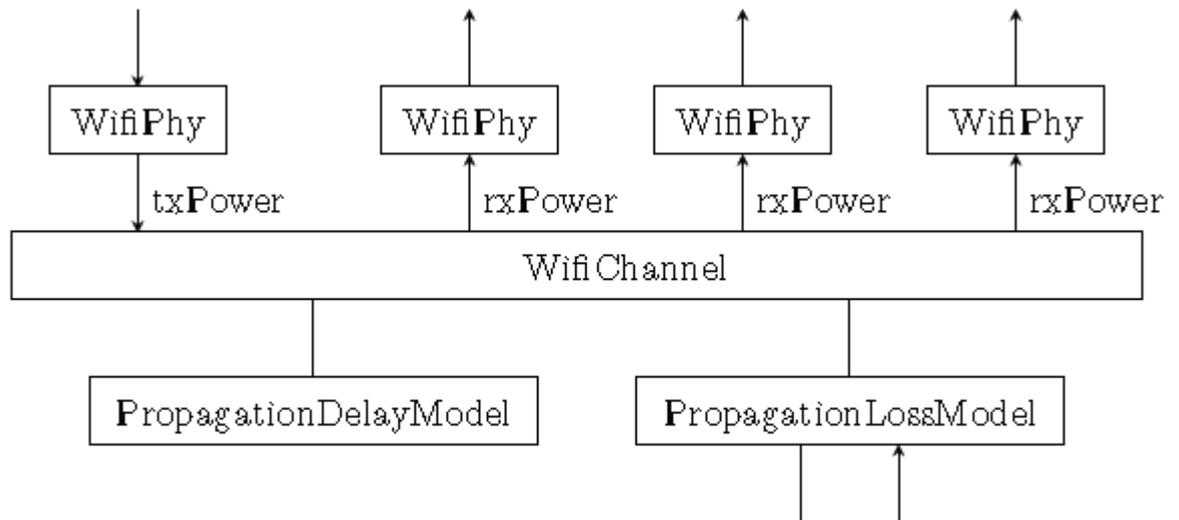


Figure III.12 : Modèle NS-3 de propagation dans le Wifi

Lorsque le premier bit d'un nouveau paquet est reçu alors que le PHY n'est pas à l'état IDLE le paquet est éliminé. Sinon, la puissance du premier bit reçu est calculée et comparée avec un seuil de détection de puissance. Si la puissance est supérieure à un certain seuil le PHY passe à l'état RX et planifie l'événement correspondant à la réception du dernier bit du paquet. Sinon, le PHY demeure dans l'état IDLE et élimine le paquet. Lorsque le dernier bit du paquet est reçu la probabilité d'erreur sur le paquet reçu est calculée, la valeur de cette dernière va permettre de décider si oui ou pas le paquet pourrait être bien reçu ou non.

III.5.2 IMPLEMENTATION DU WiMAX DANS NS-3:

Le module WiMAX de NS-3 est assez récent, son intégration officielle dans NS-3 date de Avril 2010. Sa création qui date de bien avant a fait l'objet de deux articles référencés par [22] [23].

Le module WiMAX de NS-3 est caractérisé par:

- ✓ Un modèle de canal et de couche physique réaliste et extensible.
- ✓ Un classificateur de paquet pour la sous couche de convergence IP.
- ✓ Des planificateurs efficaces pour les liens ascendants et descendants.
- ✓ Un support pour les services Broadcast et multicast
- ✓ Des fonctionnalités pour le suivi des paquets.

Du point de vu de la couche MAC, le module implémente deux modes opératoires de base, un mode pour la station d'abonné SS (Subscriber Station) et le mode pour la station de base BS (Base Station). Suivant la méthodologie NS-3, le module implémente également une classe pour la couche physique et une classe pour le canal de propagation. Le modèle implémente 3 planificateurs de lien:

- ✓ SIMPLE: un planificateur a simple priorité basé sur l'ordre FIFO.
- ✓ RTPS: a real-time polling service : un planificateur orienté service rtPS.
- ✓ MBQOS: un planificateur du lien ascendant basé migration.

Les aspects additionnels suivants sont également implémentés dans le module:

- ✓ La possibilité d'utiliser des modèles de propagation sans fil (perte et délai) et de la mobilité déjà existant dans NS-3 (proposé et associé avec le module Wifi).
- ✓ Le mode point multi-point et WirelessMAN-OFDM pour la couche physique.
- ✓ Certaines fonctions de la sous couche commune: adaptation initiale de la puissance d'émission, initialisation du flux de service, gestion de connexions et l'initialisation de transport.
- ✓ Les catégories de flux de service UGS, rtPS, nrtPS, et BE.

Certains aspects ne sont pas encore modélisés pour l'instant et le seront dans un futur proche (ex. couche physique OFDMA, auto adaptation de débit, topologie mesh, la catégorie de flux de service ertPS...). Le modèle 802.16 fournit dans NS-3 essaye de donner une implémentation physique et MAC la plus précise possible. La figure suivante montre l'architecture interne du module WiMAX NS-3:

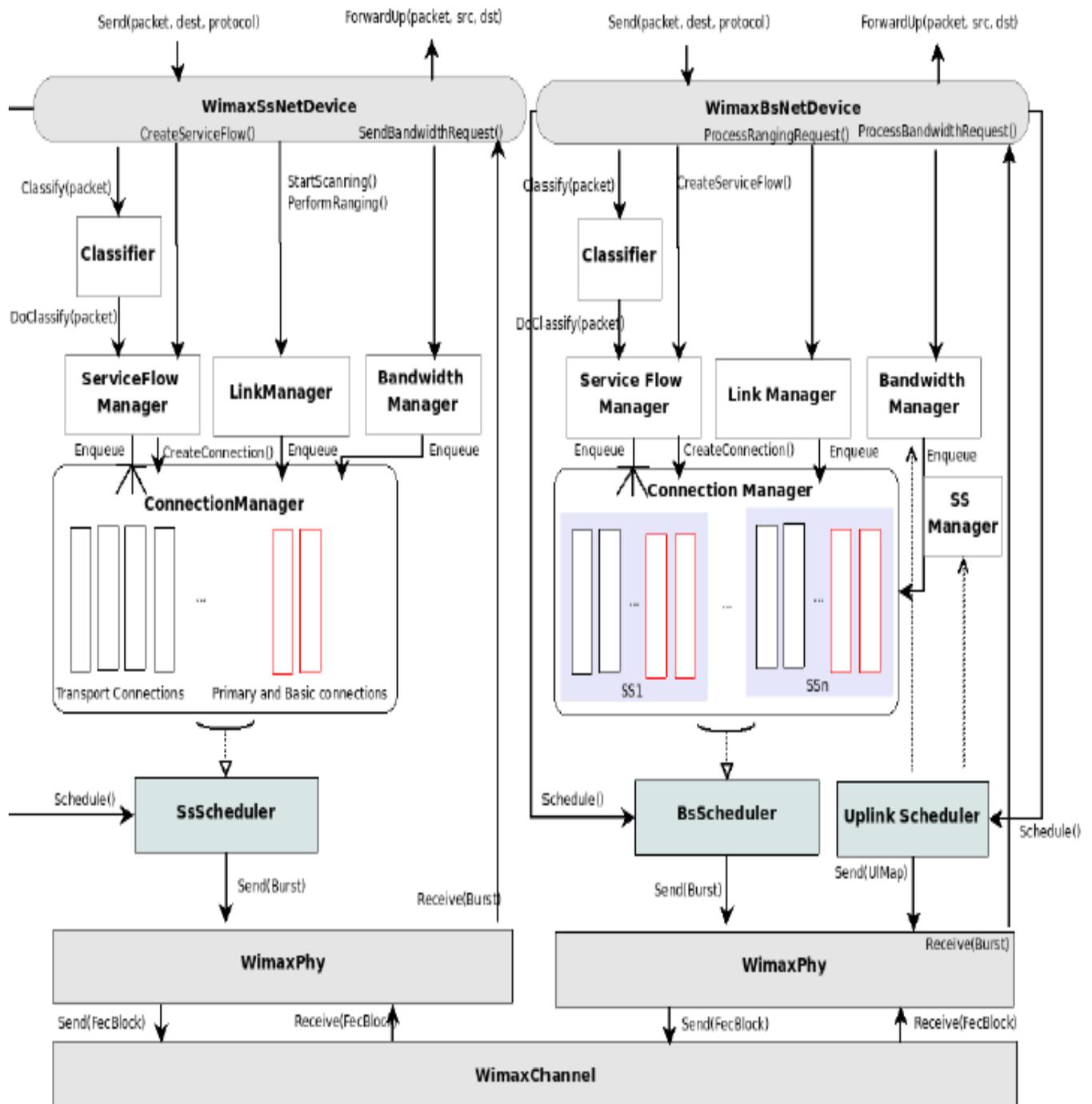


Figure III.13 : L'architecture NS-3 du WiMAX

Le module implémente la spécification PHY sans fil MAN-OFDM car c'est le schéma choisi par le Forum WiMAX. Cette spécification est conçue pour une liaison NLOS. Le modèle proposé utilise une FFT d'une largeur de 256 avec 192 sous-porteuses de données.

Le modèle de canal est implémenté par la classe *SimpleOFDMWiMAXChannel*, qui gère tous les *WiMAXNetDevices* qui lui sont connectés. La couche physique effectue deux opérations principales:

- ✓ Il reçoit un burst du canal et le transmet à la couche MAC.
- ✓ Il reçoit un burst de la couche MAC et le transmet sur le canal.

Afin de réduire la complexité de simulation de la couche physique WiMAX, le concepteur du module a choisi de modéliser séparément une partie de la couche physique. Les conditions de la couche physique sont pré-calculées pour plusieurs distances à l'aide d'une bibliothèque spécialisée (IT++) et sauvegardé dans un fichier de référence. Dans une simulation WiMAX NS-3, au lieu d'effectuer un calcul complexe à chaque fois, le résultat est pris directement à partir du fichier de référence.

III.5.3 LES DETAILS DE L'IMPLEMENTATION DE NOTRE PROPOSITION DANS NS-3

L'implémentation de la simulation est centrée autour de deux scripts NS-3. Dans le premier script *extensionWifiWiMAX_separe.cc*, la gestion de la QoS du réseau Wifi et WiMAX est séparé, ce script est utilisé pour évaluer les performances de la gestion de QoS des réseaux Wifi et WiMAX individuellement. Le deuxième script *extensionWifiWiMAX_mixte.cc* est utilisé pour évaluer les performances de la gestion de QoS couplée. Le code source des deux scripts est listé dans l'annexe.

Dans NS-3, un script de simulation est lancé à l'aide de l'utilitaire «waf» qui prend en charge la compilation, l'édition de lien des bibliothèques et l'exécution de programme C++. Cet utilitaire permet de paramétrer l'exécution du script en permettant à l'utilisateur (celui qui exécute le script) de fixer les valeurs de certaines variables de simulation à partir de la ligne de commande.

Dans le premier script le nombre de nœud Wifi (*nbSta*) et WiMAX (*nbSS*) sont des variables de simulation. Pour ce qui est de la mobilité des nœuds, initialement les noeuds sont distribués uniformément dans deux disques centrés autour du nœud *bs-ap* et dont les rayons (*rayonWifi* pour les Stas et *rayonWiMAX* pour les SSs) sont des variables de simulation. Par la suite, le modèle de mobilité *RandomWaypointMobilityModel* est attribué à chaque nœud (Sta et SS) avec une restriction de la mobilité autour de la position initiale dans une distance maximale égale à un centième du rayon des disques précédemment cités.

Dans le deuxième script, les nœuds font aussi bien du Wifi que du WiMAX, du fait que chaque station est dotée des deux interfaces simultanément. Le nombre global de nœuds mixtes est fixé à l'aide d'une variable de simulation *nbNoeuds*. La mobilité est configurée d'une façon similaire au premier script avec un disque unique ayant un rayon *rayonMixte*.

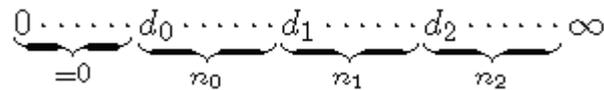
Nous avons choisi la configuration Wifi 802.11a avec comme module de délai de propagation *ConstantSpeedPropagationDelayModel*. La perte de propagation est simulée par la superposition de deux modules : *ThreeLogDistancePropagationLossModel* et *NakagamiPropagationLossModel*.

Le premier module *ThreeLogDistancePropagationLossModel* est une variante du modèle de perte à logarithme de la distance dans lequel la perte L est estimée comme suit :

$$L = L_0 + 10 \cdot n \cdot \log_{10} \left(\frac{d}{d_0} \right) \quad \text{Équation III-1}$$

Où L est exprimée en décibel, n est l'exposant de la distance, d la distance entre les antennes, L_0 est la perte de référence mesuré à la distance de référence d_0 .

Dans le modèle *ThreeLogDistancePropagationLossModel* quatre régions sont définies à l'aide de trois distances d_0 , d_1 et d_2 comme suite :



La perte dans la première région ($d < d_0$) est supposée être nulle, dans les trois autres régions la perte est estimée d'une manière similaire au modèle de logarithme de la distance avec un exposant propre à chaque région. Le calcul de perte L se fait comme suite :

$$L = \begin{cases} 0 & d < d_0 \\ L_0 + 10 \cdot n_0 \log_{10} \left(\frac{d}{d_0} \right) & d_0 \leq d < d_1 \\ L_0 + 10 \cdot n_0 \log_{10} \left(\frac{d_1}{d_0} \right) + 10 \cdot n_1 \log_{10} \left(\frac{d}{d_1} \right) & d_1 \leq d < d_2 \\ L_0 + 10 \cdot n_0 \log_{10} \left(\frac{d_1}{d_0} \right) + 10 \cdot n_1 \log_{10} \left(\frac{d_2}{d_1} \right) + 10 \cdot n_2 \log_{10} \left(\frac{d}{d_2} \right) & d_2 \leq d \end{cases}$$

Équation III-2

Les valeurs par défaut choisies dans NS-3 sont $d_0=1m$, $d_1=200m$, $d_2=500m$, $n_0=1.9$, $n_1=n_2=3.8$ et $L_0=46.67dB$. Dans notre simulation nous avons gardé ces valeurs.

Le deuxième module *NakagamiPropagationLossModel* est basé sur le modèle d'évanouissement de Nakagami qui est un modèle statistique décrivant l'effet de la superposition de rayons multiples de propagation radio arrivant au récepteur par différents chemins.

Dans les deux scripts, le nombre d'applications voix, vidéo et données sont des variables de simulation à partir desquelles le nombre de nœuds du réseau Lan2 est déduit (un nœud pour chaque application). En plus des variables de simulation déjà citées, on trouve une variable *duration* pour fixer la période de temps durant laquelle se déroule la simulation.

Dans les deux scripts, le protocole TCP/IPv4 est installé dans tous les nœuds. Le plan d'adressage adopté pour les différents réseaux est comme suite:

- ✓ Le 12.1.1.0/24 pour le réseau LAN2
- ✓ Le 11.1.1.0/24 pour le réseau LAN1
- ✓ Le 10.1.0.0/24 pour le réseau WiMAX
- ✓ Le 9.1.0.0/24 pour le réseau Wifi

La méthode de routage configurée dans le routeur et le nœud bs-ap est le routage statique implémenté à l'aide de la classe *GlobalRouter*.

Nous avons choisi la classe d'application *UdpTraceClient* pour simuler les différents types d'application. Un objet de cette classe envoie continuellement des paquets dont la taille et le délai sont pris à partir d'un fichier appelé fichier trace. Par défaut la classe utilise un fichier trace déjà existant et qui correspond à un flux vidéo. Le fichier et le détail du flux vidéo peuvent être trouvés dans le site web <http://www.tkn.tu-berlin.de/research/trace/ltvt.html>

Le flux des applications voix est généré à partir d'un fichier trace d'une conversation de téléphonie sur IP pré-capturée. Ce fichier est le résultat de traitement par le logiciel Wireshark d'un fichier de capture de paquet (pcap) téléchargé à partir du lien web (<http://www.techtraces.com/files/sip-rtp-g711.cap>). D'une manière similaire le flux des applications données est généré à partir d'un fichier trace correspondant à la navigation web dans un forum de discussion.

Nous avons personnalisé l'implémentation de la classe *UdpTraceClient* afin d'y intégrer la différenciation de QoS de trafic Wifi en attachant à chaque paquet envoyé un *QosTag* adéquat. Au moment où chaque paquet est envoyé par un objet *UdpTraceClient* un message est affiché contenant la taille du paquet et l'adresse ip destination. Dans notre personnalisation, ce message est enrichi en ajoutant le numéro de séquence du paquet et l'adresse ip source ceci dans le but de faciliter le calcul du nombre de paquets perdus.

NS-3 incorpore un système interne de suivi de simulation qui est appelé « Logging ». Ce système affiche des messages différents pendant l'exécution d'un script. Le système de Logging est divisé en plusieurs niveaux suivant la nature du message, ainsi nous avons les messages informatifs dans le niveau LOG_INFO, les messages d'erreur dans le niveau LOG_ERROR, les messages d'alerte dans le niveau LOG_WARN et les messages de débogage dans le niveau LOG_DEBUG qui concernent le fonctionnement interne bas niveau de NS-3.

Dans les deux scripts, des applications de type *UdpServer* sont installées au niveau des nœuds Wifi et WiMAX. Une telle application est couplée avec un *UdpTraceClient* pour recevoir les paquets envoyés par ce dernier. A la réception de chaque paquet, *UdpServer* affiche un message informatif spécifiant le numéro de séquence du paquet reçu, sa taille, son adresse ip d'origine et le délai entre son émission et sa réception.

D'autres messages informatifs sont aussi affichés par les deux scripts, ce pourquoi il faut effectuer un traitement textuelle (filtrage) sur les sortie des scripts pour en tirer les informations qui nous intéressent (délai et paquets perdus). L'extraction de ces dernières n'est pas suffisante car nous avons souvent besoin de les transformer et les afficher sous forme de graphe.

Afin de simplifier ce processus de traitement et aussi dans le but d'automatiser la répétition de la simulation pour diverses valeurs des paramètres, nous avons choisi de piloter les scripts dans Scilab. Ce dernier est un environnement de calcul scientifique similaire à Matlab[®] mais qui est open source et libre de droit.

Le pilotage des scripts sous Scilab se fait par l'utilisation de la fonction Scilab `host()` pour invoquer l'exécution d'une simulation NS3, le résultat étant redirigé vers un fichier. Par la suite la fonction `mgetl()` est utilisée pour charger le fichier dans Scilab. Finalement, un script Scilab que nous avons créé, effectue le passage (filtrage des données) pour extraire les informations intéressantes, les visualiser et les exploiter.

III.6 RESULTATS ET DISCUSSION

Le premier script `extensionWifiWiMAX_séparé.cc` a été paramétré pour simuler 10 secondes de fonctionnement de réseau et pour un nombre d'application allant de 1 à 13 avec un pas de 2. et pour les rayons de distribution des nœuds Wifi et WiMAX variant entre 1m et 581m avec un pas de 20 m.

Il est à noter que le PC utilisé pour la simulation a 3Gigat de Ram et on a installé NS3 sous linux.

Après une première exécution qui a duré plusieurs semaines, les résultats se sont avérés suspects concernant le nombre de paquet perdu. En effet il fallait programmer le temps d'arrêt des applications réceptrices de quelques millisecondes après l'arrêt des applications émettrices pour permettre aux derniers paquets en route d'être réceptionné.

Les résultats de la deuxième exécution, après modification du premier script qui a également durée plusieurs semaines, ont été évalués dans Scilab pour calculer le délai moyen, le délai maximum, le nombre de paquets perdus moyen et le nombre de paquets perdus maximum. Les résultats pour le délai moyen et le nombre de paquets perdus moyen sont illustrés dans les figures suivantes.

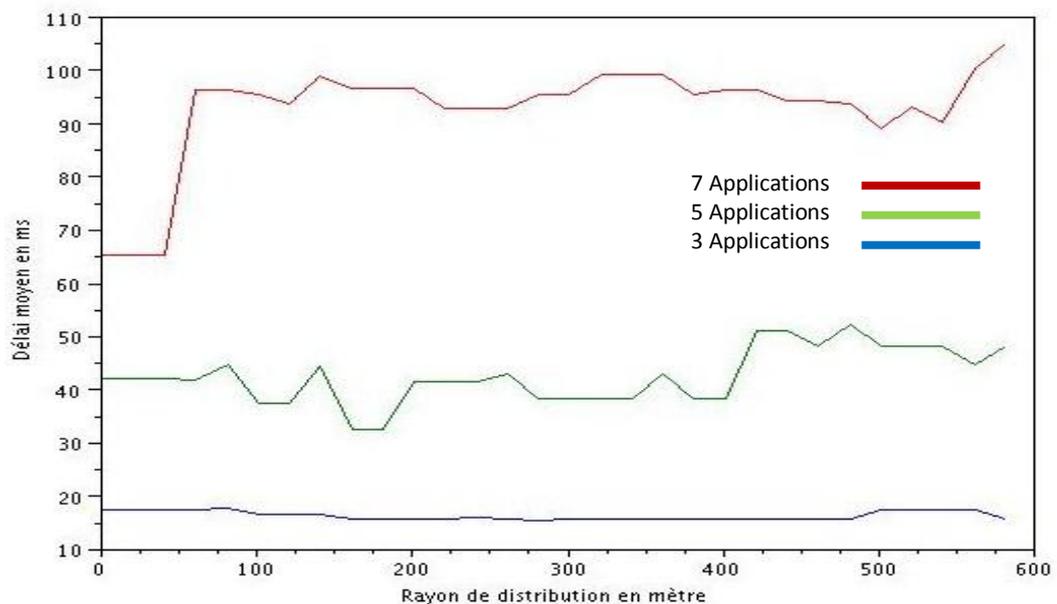


Figure III.14: Variation du délai moyen pour les flux Vidéo Wifi en fonction du rayon de distribution des nœuds.

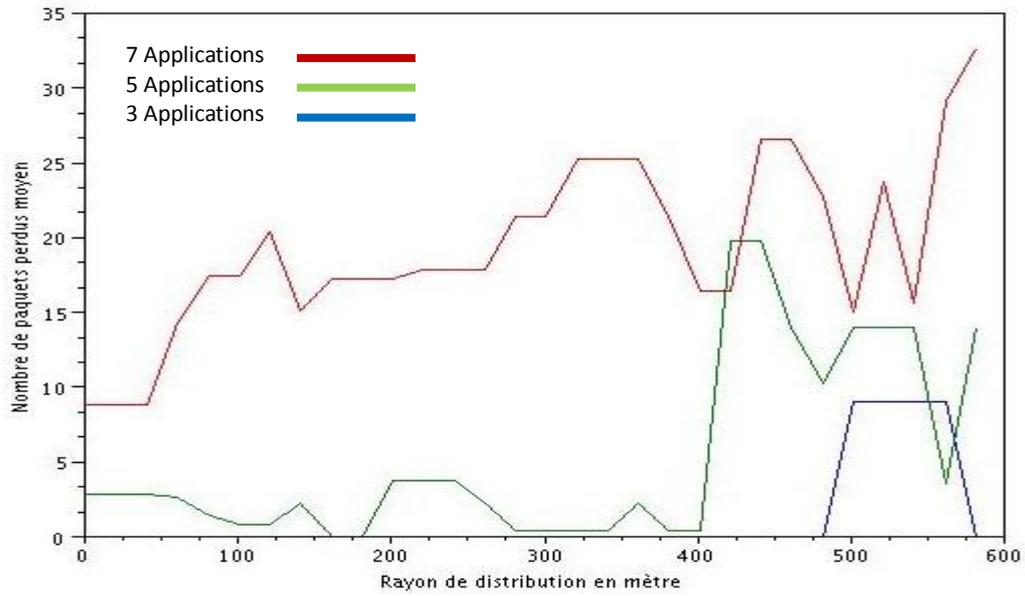


Figure III.15 : Variation du nombre de paquets perdus moyen pour les flux Vidéo Wifi en fonction du rayon de distribution des nœuds.

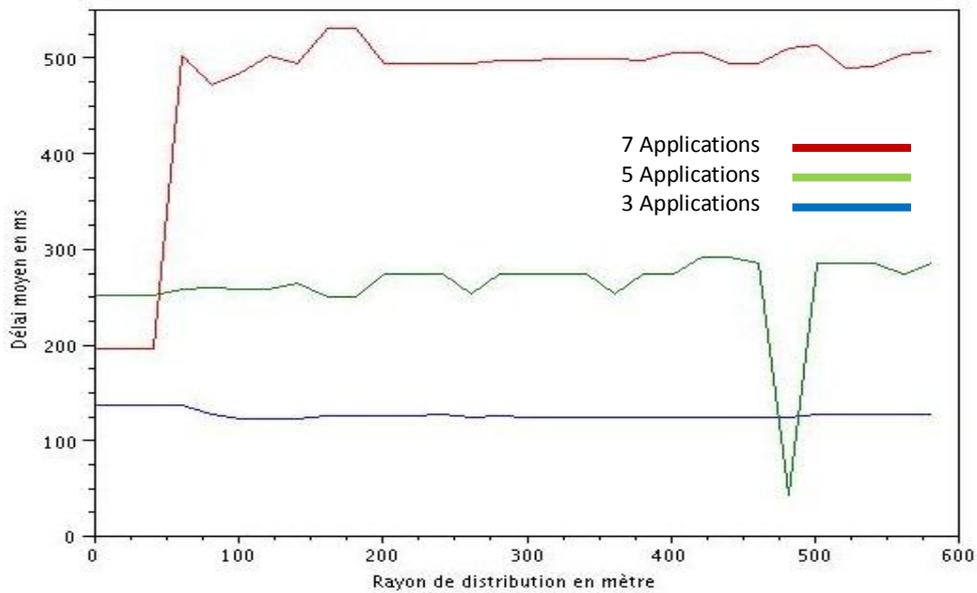


Figure III.16 : Variation du délai moyen pour les flux Voix Wifi en fonction du rayon de distribution des nœuds.

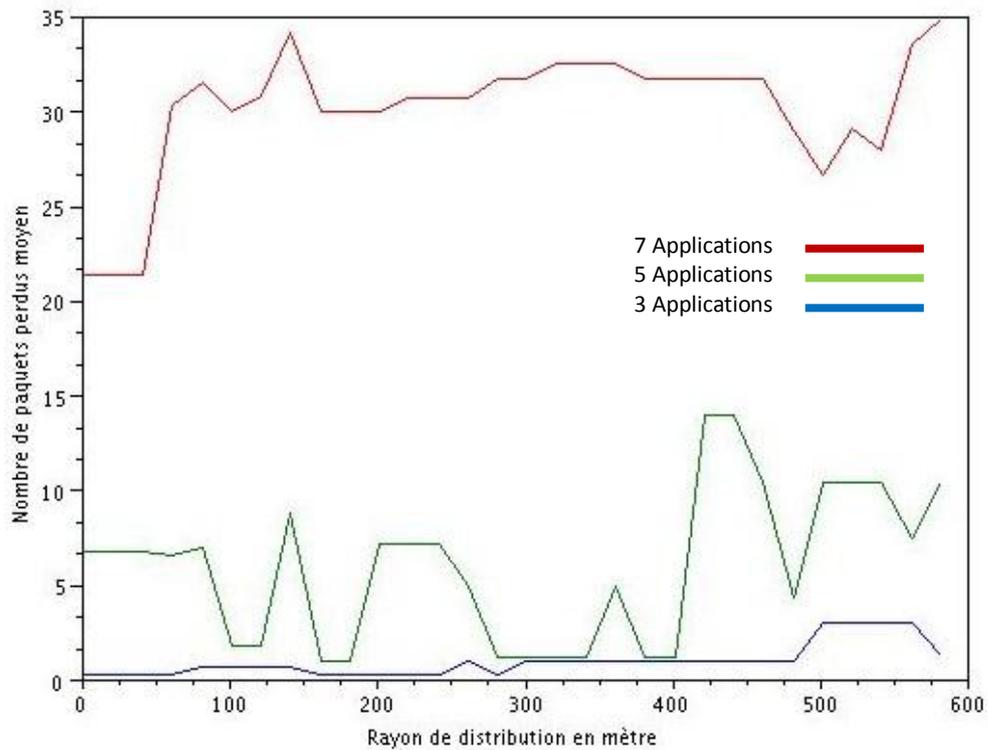


Figure III.17 : Variation du nombre de paquets perdus moyen pour les flux Voix Wifi en fonction du rayon de distribution des nœuds.

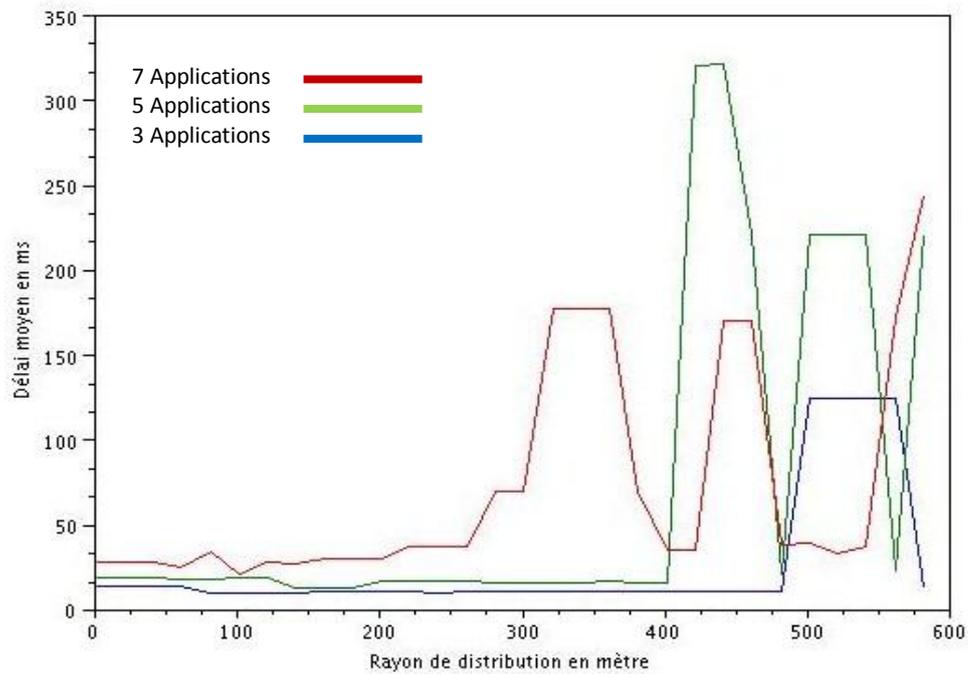


Figure III.18 : Variation du délai moyen pour les flux de données Wifi en fonction du rayon de distribution des nœuds.

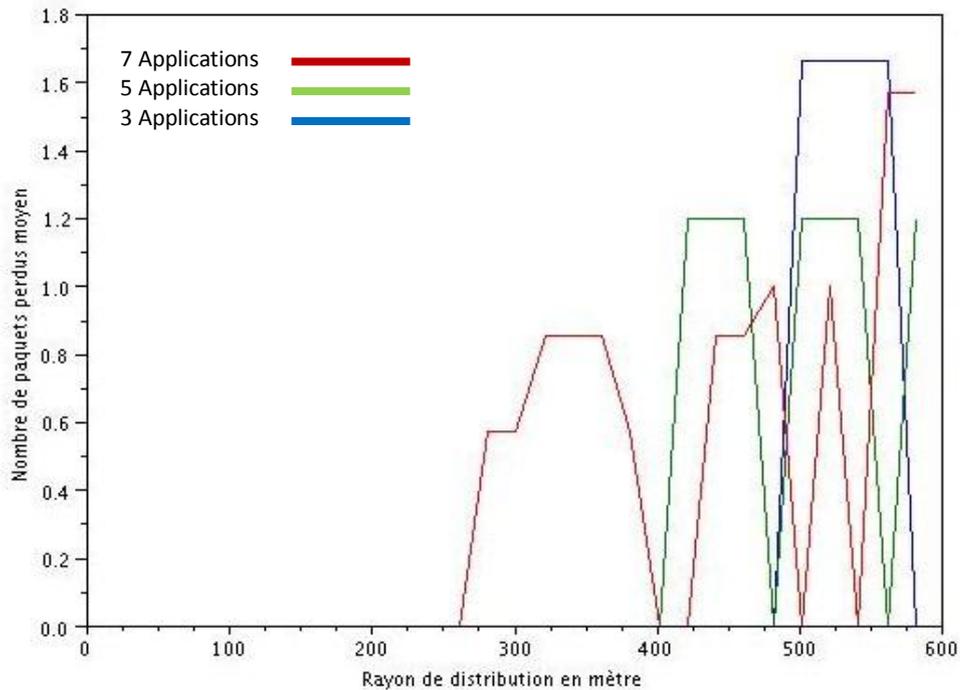


Figure III.19 : Variation du nombre de paquets perdus moyen pour les flux de données Wifi en fonction du rayon de distribution des nœuds.

Dans la figure 14, on remarque que tant que le nombre de stations du réseau wifi émettant un flux vidéo chacune et qui nécessite de la QoS, est bas inférieur à 3, le délai moyen de transmission reste inférieur à 20ms, même si les stations s'éloignent de l'AP. Mais dès que le nombre de stations augmente de 5 puis à 7, le délai augmente de façon considérable surtout pour les stations positionnées à plus de 100 m, ceci dit reste encore majoré par 100ms, valeur tout à fait acceptable pour des flux vidéo. La figure 15 confirme la tendance précédente, c.à.d. que le nombre moyen de paquets perdus augmente avec l'augmentation des stations surtout après un rayon de distribution supérieur à 400m.

La figure 16 montre que tant que le nombre de stations est petit (3, 5), le délai moyen des flux voix, qui sont très sensibles à ce paramètre de QoS en particulier, reste acceptable mais s'envole dès que le rayon dépasse les 100m et la QoS n'est plus garantie. Chose confirmée avec la figure17 qui donne le nombre moyen de paquets perdus pour des applications voix, qui s'envole vers des valeurs inacceptables surtout à partir d'un rayon de distribution de 400m. Pour des flux data en figure 18 et 19, après 400m de distance, le délai aussi bien que le nombre moyen de paquets perdus, deviennent instables et fluctuants surtout pour un nombre de stations supérieur à 5.

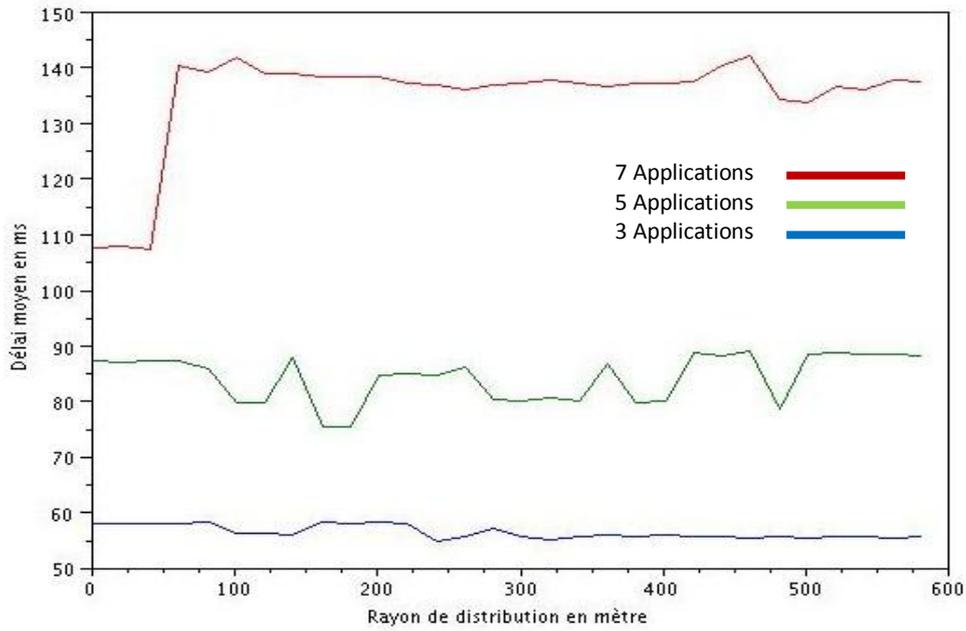


Figure III.20 : Variation du délai moyen pour les flux Vidéo WiMAX en fonction du rayon de distribution des nœuds.

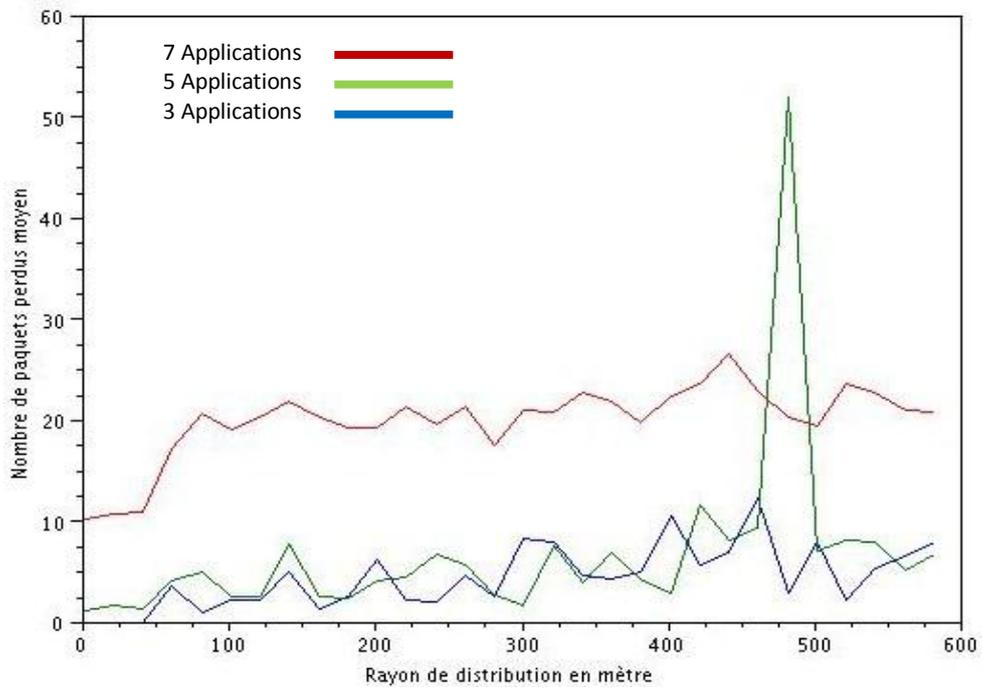


Figure III.21 : Variation du nombre de paquets perdus moyen pour les flux Vidéo WiMAX en fonction du rayon de distribution des nœuds.

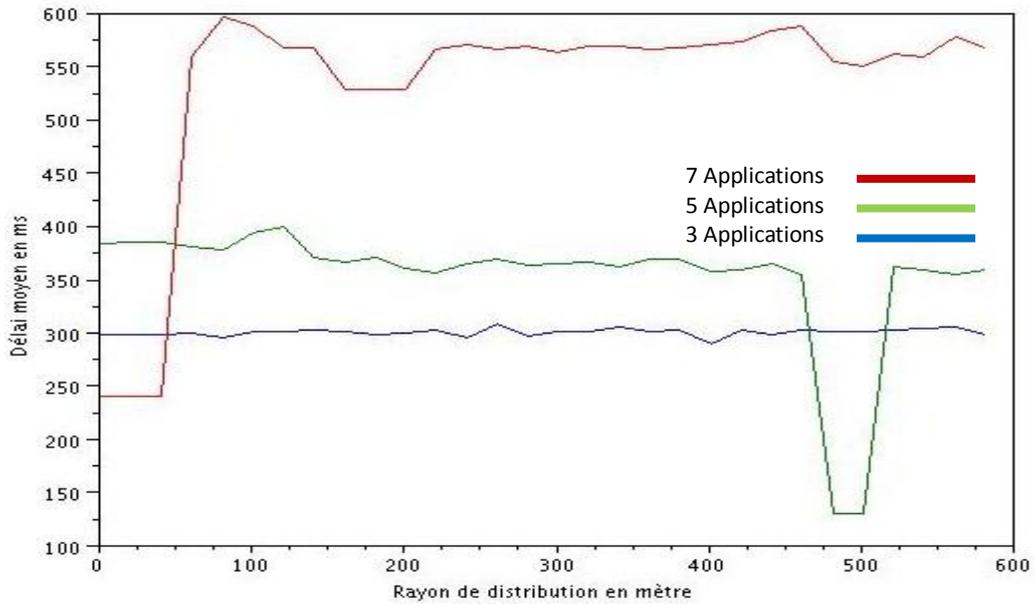


Figure III.22 : Variation du délai moyen pour les flux Voix WiMAX en fonction du rayon de distribution des nœuds.

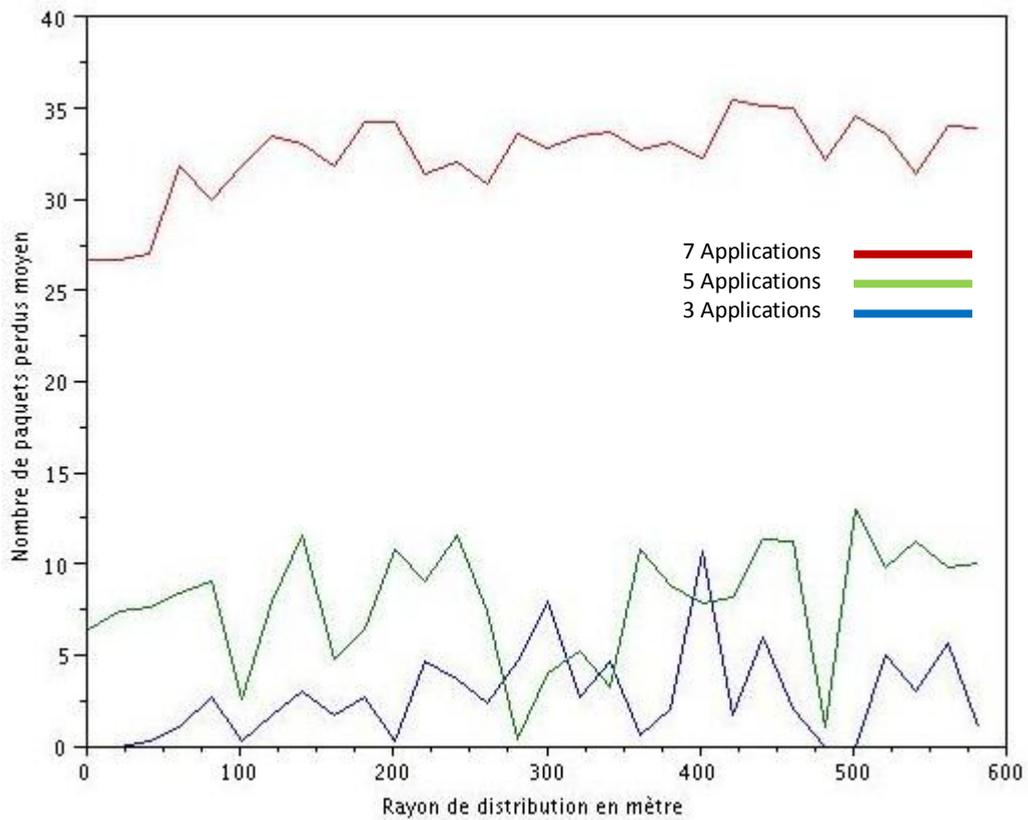


Figure III.23 : Variation du nombre de paquets perdus moyen pour les flux Voix WiMAX en fonction du rayon de distribution des nœuds.

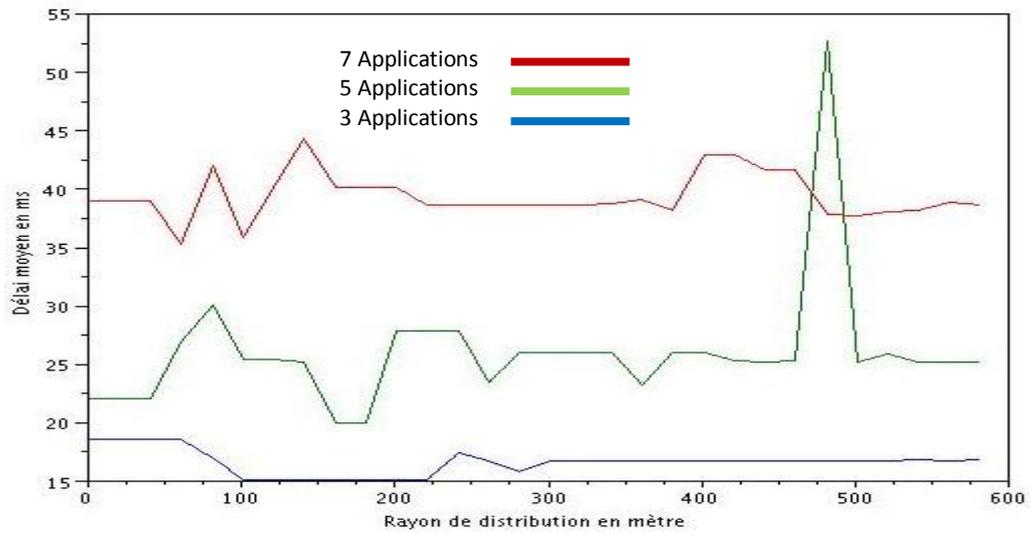


Figure III.24 : Variation du délai moyen pour les flux de données WiMAX en fonction du rayon de distribution des nœuds.

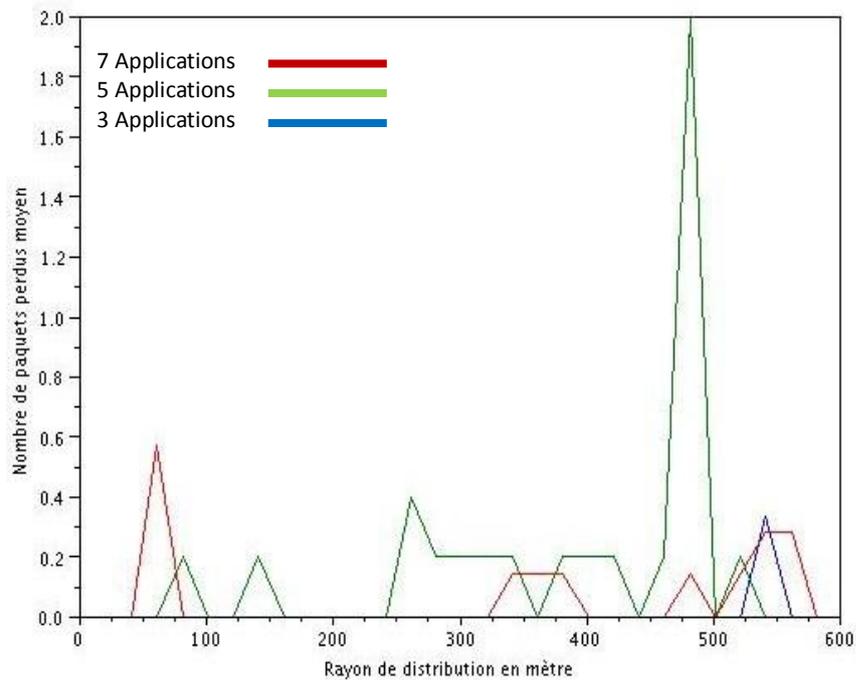


Figure III.25 : Variation du nombre de paquets perdus moyen pour les flux de données WiMAX en fonction du rayon de distribution des nœuds.

Pour le réseau Wimax, les mêmes constatations peuvent être confirmées que pour le réseau wifi avec quelques différences notables comme le rayon de distribution, ou commence la dégradation des paramètres de QoS, est plus important, supérieur à 600m, que pour le cas

du réseau Wifi. Cependant les paramètres de QoS ne sont pas aussi stricts que pour le Wifi, comme par exemple le délai moyen pour les flux vidéo sous wimax est de 60ms minimum et 150ms maximum, et 20ms minimum, 100ms maximum pour wifi.

Comme on peut s'en rendre compte par les figures 20-25, aussi bien pour les flux vidéo, voix que data, et pour les deux paramètres étudiés ici le délai moyen de transmission et le nombre moyen de paquets perdus.

Dans le deuxième script *extensionWifiWiMAX_mixte.cc*, nous avons étudié et évalué les performances de notre algorithme pour les rayons allant de 301 m à 601 m avec un pas de 20 m et pour un nombre de nœuds mixtes Wifi/WiMAX variant de 5 à 8. Par la suite et comme avec le premier script nous avons calculé le délai moyen, le délai maximum, le nombre de paquets perdus moyen et le nombre de paquets perdus maximum. Les résultats sont illustrés dans les figures suivantes :

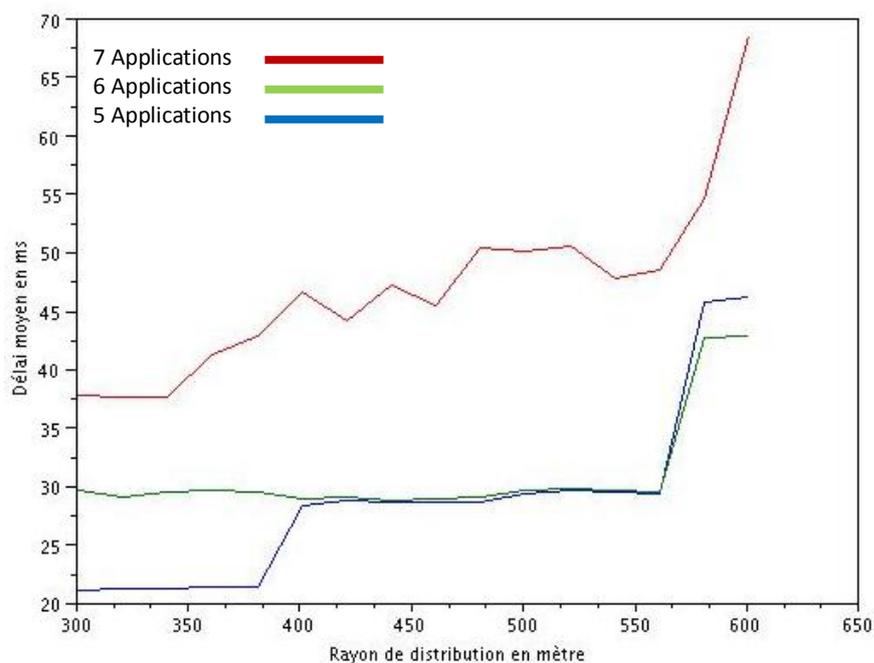


Figure III.26 : Variation du délai moyen pour les flux Vidéo Mixte en fonction du rayon de distribution des nœuds.

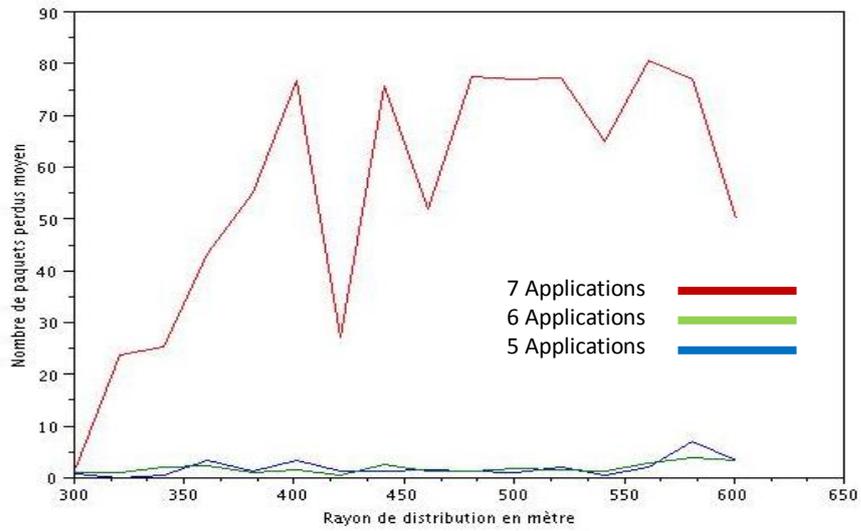


Figure III.27 : Variation du nombre de paquets perdus moyen pour les flux Vidéo Mixte en fonction du rayon de distribution des nœuds

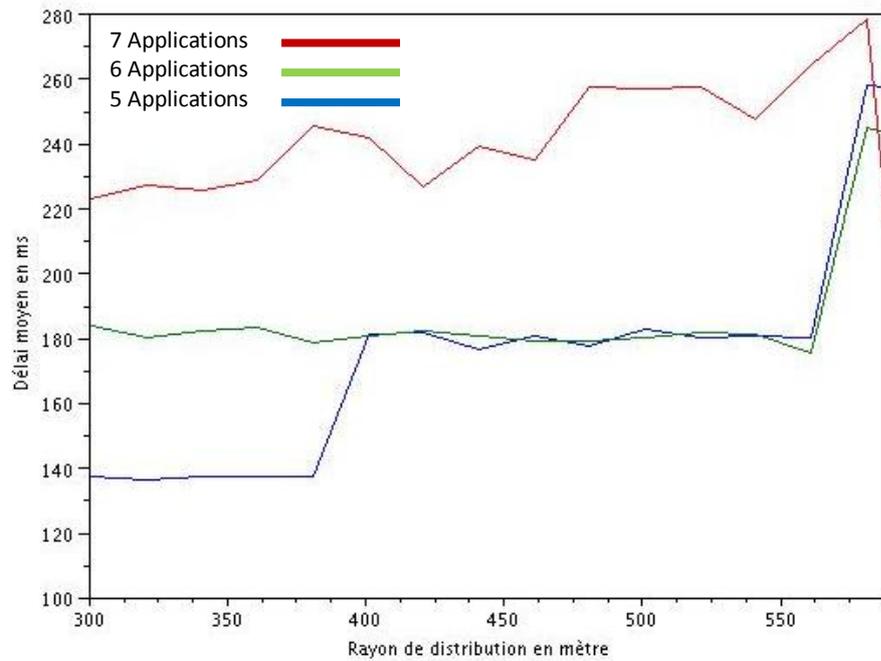


Figure III.28 : Variation du délai moyen pour les flux Voix Mixte en fonction du rayon de distribution des nœuds.

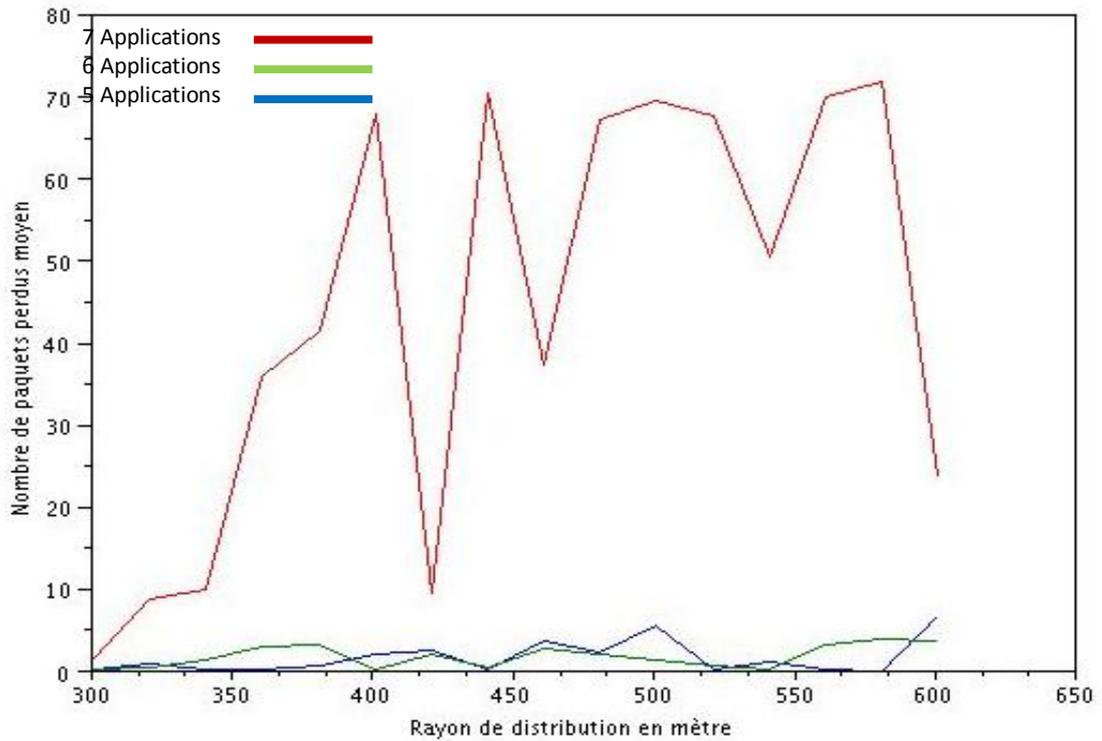


Figure III.29 : Variation du nombre de paquets perdus moyen pour les flux Voix Mixte en fonction du rayon de distribution des nœuds.

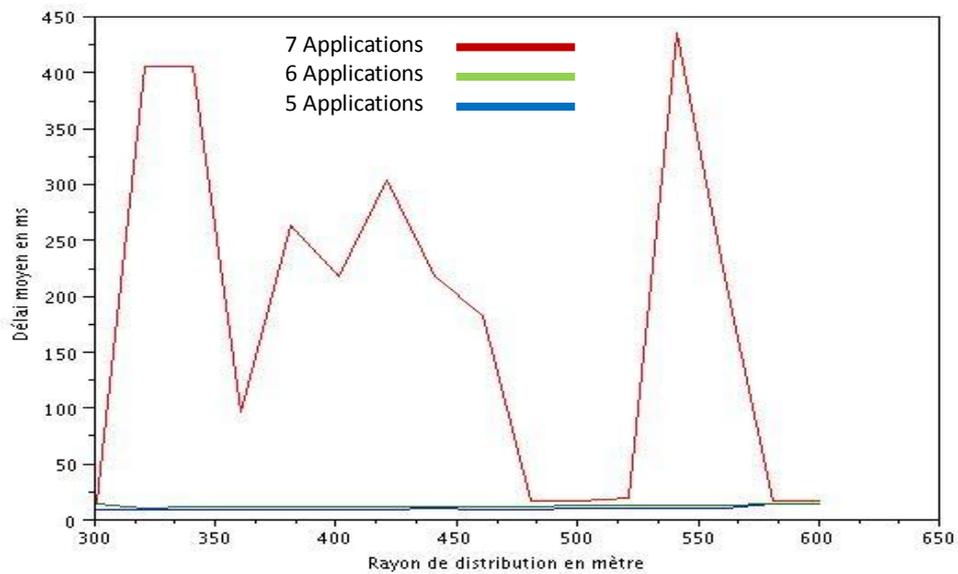


Figure III.30 : Variation du délai moyen pour les flux de données Mixte en fonction du rayon de distribution des nœuds.

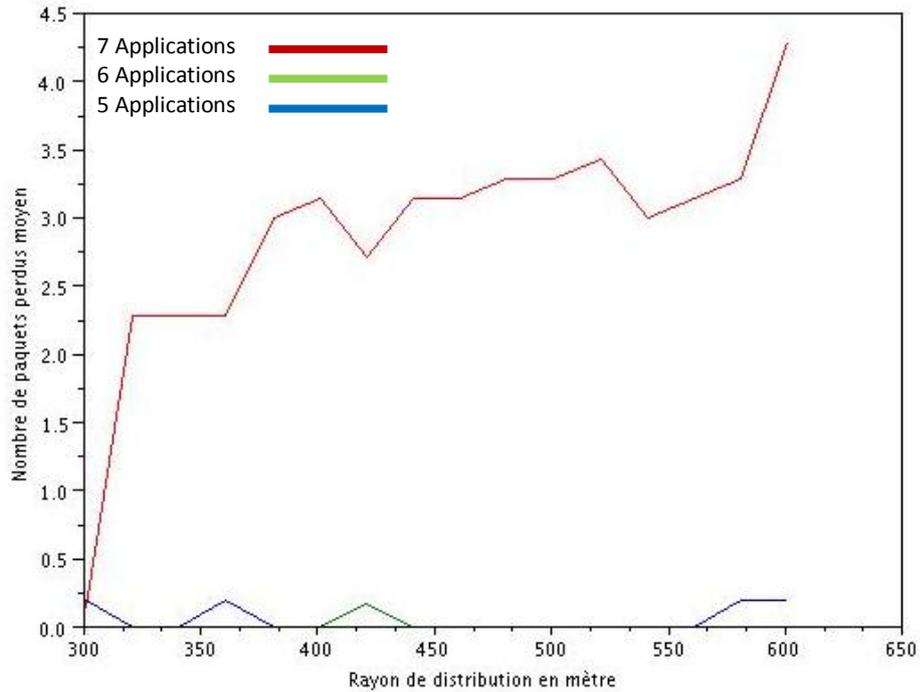


Figure III.31: Variation du nombre de paquets perdus moyen pour les flux de données Mixte en fonction du rayon de distribution des nœuds.

Dès que les deux réseaux sont utilisés pour travailler en complémentarité parfaite, les paramètres de QoS sont garantis sur des rayons allant jusqu'à 700m comme on peut le voir pour le paramètre de QoS le délai moyen pour des flux vidéo, voix et data dans les figures 26, 28 et 30. Le délai est majoré par la valeur 100,500,350 pour le wifi ; 140,600, 40 pour le wimax et 70,280,400 pour le mixte ; pour les flux vidéo, voix et data respectivement. Ce qu'on remarque c'est que les paramètres de QoS des flux vidéo et voix sont améliorés de façon significative mais dégradés pour le flux data, chose qui peut s'expliquer par le fait que on préfère dégrader les applications ne nécessitant pas de QoS, plutôt que celles temps réel. Par contre, on remarque que le paramètre nombre de paquets perdus a significativement augmenté pour le cas mixte par rapport aux réseaux wifi ou wimax seuls, et ceci peut s'expliquer par le fait que le handover ne se fait pas sans une certaine perte de paquets, et donc il faudrait prévoir des mécanismes supplémentaires pour faire face à ce problème.

Les résultats précédents ont été utilisés pour déterminer la distance limite D qui sépare les deux Zone Z1 et Z2 mentionnées dans la section 3.2. Ainsi une limite acceptable a été fixée pour le délai moyen, le délai maximum, le nombre de paquets perdus moyen et le nombre de paquets perdu maximum. Par la suite, pour chaque rayon, nous avons déterminé le nombre d'application nWifi et nWiMAX maximum satisfaisant les conditions limites de la qualité de service acceptables pour les nœuds Wifi et WiMAX respectivement. La figure suivante montre la variation de ces derniers en fonction du rayon.

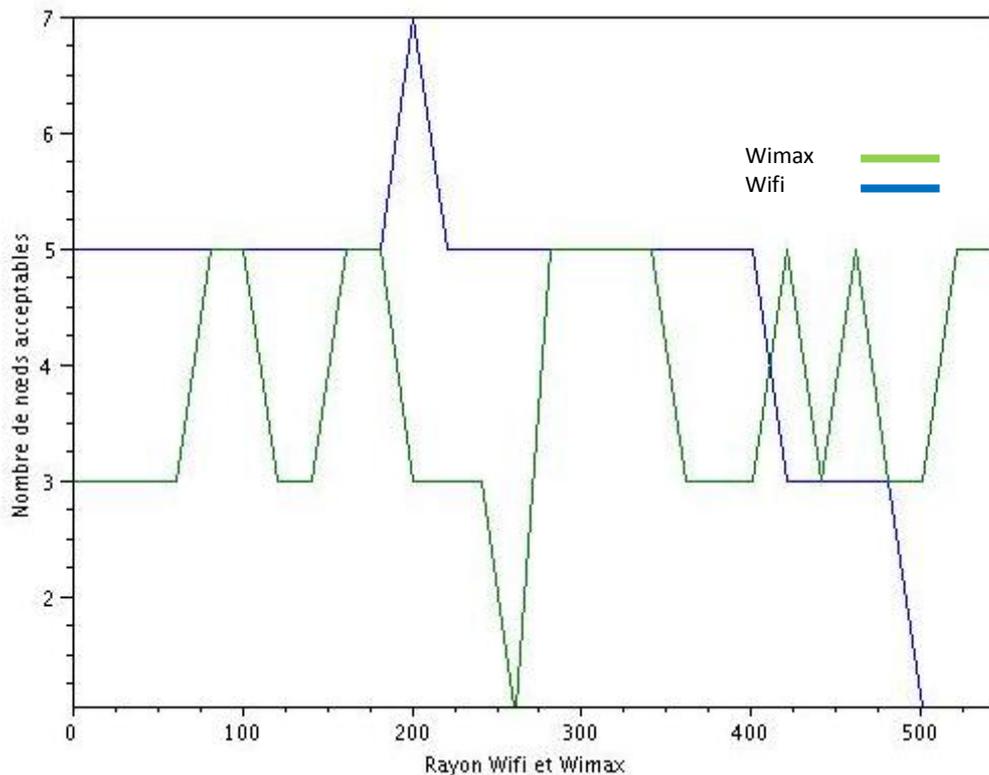


Figure III.32 : Nombre d'applications admissibles en fonction du rayon de distribution pour les nœuds Wifi et WiMAX.

On remarque que le nombre de station nWifi est supérieur au nWiMAX pour les rayons inférieurs à 400m, en revanche le nombre de station nWifi diminue alors que nWiMAX reste plus au moins stable pour les rayons supérieurs à 400m.

Par conséquent nous avons décidé de prendre le chiffre 400m comme valeur pour la distance D puisque le nWiMAX est supérieur au nWifi pour les rayons supérieurs à 400m. Nous avons également décidé de prendre la valeur 5 comme le nombre de stations Wifi maximum, ce qui permet de garantir une QoS pour toutes ces stations de façon sûre.

Après l'application des conditions limites aux résultats du deuxième script, nous avons calculé le nombre de nœuds nMixte maximum acceptable pour chaque rayon. La figure suivante montre la variation de ce dernier en fonction du rayon.

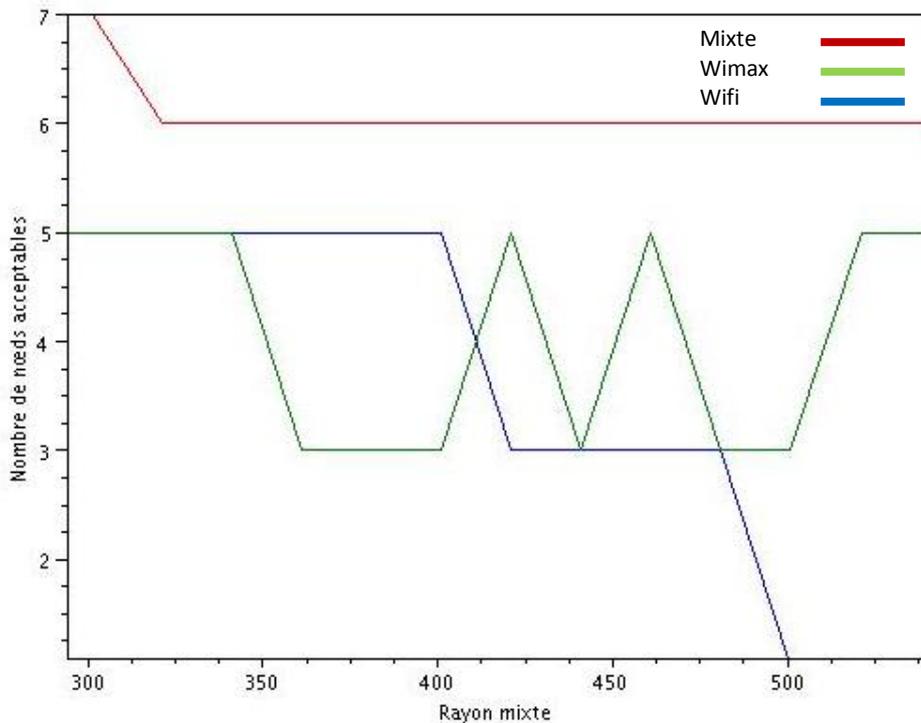


Figure III.33: Nombre d'applications admissibles en fonction du rayon de distribution pour les nœuds Wifi, WiMAX et Mixte.

Donc on voit bien que on peut se permettre d'accepter bien plus de stations quant on travaille en collaboration qu'avec un ou l'autre des réseaux seul.

III.7 CONCLUSION

Si on compare les résultats des deux scripts (la figure ci-dessus), on peut constater clairement l'amélioration du nombre de nœuds acceptable globale apporté par notre stratégie d'admission pour les rayons élevés supérieur à 500m. Ceci veut dire que l'efficacité de notre algorithme est apprécié dans le cas ou les nœuds son éparpillés sur une grande distance. Ce qui est le cas le plus probable dans un scénario réaliste dans des espaces métropolitains. Donc au lieu de ne déployer qu'un réseau Wifi ou Wimax et ne couvrir qu'une zone limitée ou n'offrir qu'une QoS bas de gamme, en jumelant les deux technologies, on arrive à augmenter la zone de couverture et à garantir une QoS stricte avec un nombre de stations qu'on peut accepter, de plus en plus croissant.

CONCLUSION ET PERSPECTIVES

Les réseaux du futur seront inévitablement un ensemble d'une multitude de technologies opérant conjointement d'une manière transparente à l'utilisateur qui ne sera concerné que par la QoS offerte. Ce même utilisateur du futur ne sera plus obligé de changer son mobile pour un nouveau, suite à l'apparition d'une nouvelle technologie car une simple mise à jour de son ancien mobile sera suffisante. La clé à ce réseau du futur est la maîtrise de la convergence des différentes technologies d'accès réseau.

Dans le premier chapitre, nous sommes arrivés à la confirmation qu'aujourd'hui, l'existence d'une étroite relation entre le monde de la téléphonie mobile et celui de la toile cybernautique fait l'unanimité de tous les acteurs principaux de l'évolution des réseaux sans fil. En effet Internet est construit sur des standards vivants et évolutifs apportant aux opérateurs la possibilité d'enrichir leurs services et offrant ainsi au grand public une riche expérience et une simplicité d'utilisation. La question principale est maintenant comment rendre l'accès à Internet omniprésent ? Ce qui impose la notion de convergence des différentes technologies d'accès réseau sans fil vers une architecture globale centralisée autour d'Internet.

Nous nous sommes intéressés à la convergence entre les réseaux sans fil Wifi et WiMAX. Dans le deuxième chapitre, nous avons présenté les principales notions de fonctionnement et de gestion de la QoS pour ces deux derniers. Nous avons constaté que ces deux technologies peuvent être combinées pour réaliser une convergence basée sur la collaboration et qui aura pour objectif l'optimisation de la QoS offerte. Ceci n'est possible qu'avec l'introduction d'une stratégie de contrôle d'accès des flux dans le réseau composite résultant.

Dans le troisième chapitre, nous avons construit l'idée principale dans laquelle un réseau WiMAX vient s'ajouter à un réseau Wifi préexistant dans le but d'augmenter la capacité d'accès réseau de façon optimal. Ainsi, notre contribution se résume à la proposition d'une stratégie de couplage de la gestion de la QoS dans ce cas de figure. Cette stratégie est présentée sous la forme d'un algorithme centralisé de contrôle d'accès des flux des deux réseaux. La proposition trouve son origine dans une récente tendance de convergence des réseaux UMTS et Wifi à laquelle nous avons ajouté un critère lié à la distribution géographique.

Notre algorithme donne d'assez bon résultats qu'il faudrait encore améliorer, surtout par rapport aux réseaux wifi et wimax seuls, comme les résultats de simulation l'ont démontré tout le long du chapitre trois. Donc l'idée de la convergence de plusieurs technologies réseaux semble bien partie pour durer et produire à l'utilisateur final un accès à internet avec une QoS garantie, leur permettant ainsi d'utiliser les applications temps réel comme la vidéo à la demande, la ToIP, etc.

L'implémentation de l'algorithme proposé passe par une étape de détermination d'une distance limite séparant deux zones ayant des règles de contrôle d'accès différentes. Cette distance est déterminée par le biais de la simulation. Après la comparaison de plusieurs simulateurs, nous avons opté pour le simulateur NS-3 dans lequel nous avons implémenté deux scripts principaux de simulation. Le premier script concerne l'évaluation de la QoS dans le réseau Wifi et WiMAX séparément tout en intégrant les étapes nécessaires pour la détermination de la distance limite. Le deuxième script simule la gestion de la QoS couplé en utilisant les informations déduites du premier script. Les résultats trouvés ont confirmé l'efficacité de l'algorithme proposé pour l'augmentation de la capacité du réseau composite globale.

Les perspectives de notre travail sont multiples et variées. Par exemple, de nombreuses améliorations peuvent être ajoutées à la stratégie de couplage pour satisfaire des buts alternatifs comme :

- ✓ L'optimisation de l'énergie consommée.
- ✓ L'optimisation de la QoS d'une classe d'application particulière.
- ✓ L'optimisation du coût.
- ✓ Autres...

Un autre exemple est la variation de l'implémentation de notre algorithme dans un environnement réel ou de nouvelles variables peuvent s'ajouter au problème original. On peut également imaginer l'association de notre algorithme à une structure de gestion réseau hybride construite autour du protocole MIH.

Pour finir, nous concluons que nous avons répondu au but que nous nous sommes fixées au tout début dans cette thèse, en contribuant par une solution simple et efficace au couplage et la convergence des différentes technologies. Nous sommes également parvenues à acquérir un niveau de maîtrise satisfaisant de l'outil de simulation NS-3 que nous estimons comme un atout important dans nos futurs travaux de recherche.

BIBLIOGRAPHIE

- [1] W. Webb, *The complete wireless communications professional: a guide for engineers and managers*. Boston: Artech House, 1999.
- [2] M. Sauter, *Beyond 3G - Bringing Networks, Terminals and the Web Together: LTE, WiMAX, IMS, 4G Devices and the Mobile Web 2.0*. Wiley-Blackwell, 2008.
- [3] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2e éd. Prentice Hall, 2002.
- [4] I. Ghetie, "Fixed-mobile wireless networks convergence: technologies, solutions, services. Cambridge UK" New York: Cambridge University Press, 2008.
- [5] H. Labiod, H. Afifi, et C. de Santis, *Wi-Fi, Bluetooth, Zigbee and WiMax*, 1er éd. Springer, 2010.
- [6] N. Cranley et L. Murphy, *Handbook of Research on Wireless Multimedia: Quality of Service and Solutions*. Information Science Reference, 2008.
- [7] B. Hour et S. Hameed, « Proposed Enhancement of IEEE802.11e WLAN Through Real Time Simulation Study », 2007.
- [8] T. Ali Yahiya, « Approches inter-couches pour l'allocation de ressources dans les réseaux WiMAX mobiles », Thèse de doctorat, Université Pierre et Marie Curie (Paris), 2008.
- [9] K. Daniel, S. Rohde, S. Šubik, et C. Wietfeld, « Performance Evaluation for Mobile WiMAX Handover with a Continuous Scanning Algorithm », in *2009 IEEE Mobile WiMAX Symposium*, Napa Valley, California, USA, 2009, p. 30-35.
- [10] S. Bastani, S. Yousefi, M. Mazoochi, et A. Ghiamatyoun, « On the QoS tree construction in WiMAX mesh networks based on genetic algorithm approach », in *Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks - Q2SWinet '09*, Tenerife, Canary Islands, Spain, 2009, p. 1.
- [11] D. Kim et al., « An FBAR- and LTCC-based RF front-end module for Wi-Fi and WiMAX dual-mode systems », *Microwave and Optical Technology Letters*, vol. 52, n^o. 3, p. 753-757, 2010.
- [12] M. Ma, *Current technology developments of Wimax systems*, 1 vol. Dordrecht: Springer, 2009.
- [13] L. F. C. C. de Figueiredo, H. M. C. da Silva, C. M. da S. Rabadao, et A. M. de J. Pereira, « Wireless Networks Interoperability - Wifi Wimax Handover », in *2009 Fourth International Conference on Systems and Networks Communications*, Porto, Portugal, 2009, p. 100-104.
- [14] A. P. Iyer et J. Iyer, « Handling mobility across WiFi and WiMAX », in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing Connecting the World Wirelessly - IWCMC '09*, Leipzig, Germany, 2009, p. 537.

- [15] T. Ali Yahiya et H. Chaouchi, « An optimized handover decision for heterogenous wireless networks », in Proceedings of the 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks - PM2HW2N '09, Tenerife, Canary Islands, Spain, 2009, p. 137-142.
- [16] Nguyen-Vuong et N. Agoulmine, Mobility management in 4G wireless heterogeneous networks. S. l.: Evry-Val d'Essonne, 2008.
- [17] F. Didi, H. Labiod, G. Pujolle, et M. Feham, « Physical rate and contention window based admission control (PRCW) for 802.11 WLANs », in Computers and Communications, IEEE Symposium on, Los Alamitos, CA, USA, 2010, vol. 0, p. 1-7.
- [18] « A Survey of Network Simulation Tools: Current Status and Future Development ». [Online]. Available: <http://www.cs.wustl.edu/~jain/cse567-08/ftp/simtools/index.html>. [Accessed: 05-avr-2010].
- [19] E. Weingärtner, H. vom Lehn, et K. Wehrle, « A performance comparison of recent network simulators », in ICC 2009: IEEE International Conference on Communications, 2009.
- [20] « The ns-3 network simulator ». [Online]. Available: <http://www.nsnam.org/>. [Accessed: 04-mars-2011].
- [21] M. Lacage et T. R. Henderson, « Yet another network simulator », Proceeding from the 2006 workshop on ns-2: the IP network simulator, 2006.
- [22] J. Farooq et T. Turetli, « An IEEE 802.16 WiMAX Module for the NS-3 Simulator », in 2nd International Conference on Simulation Tools and Techniques (SIMUTools'09), 2009.
- [23] M. A. Ismail, G. Piro, L. A. Grieco, et T. Turetli, « An improved IEEE 802.16 WiMAX module for the ns-3 simulator », in Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, Torremolinos, Malaga, Spain, 2010, p. 1-10.

LISTE DES ABRÉVIATIONS

AC	Access Category
ACK	Acknowledgement
AIFS	Arbitration Inter Frame Space
AMC	Adaptive Modulation and Coding
AMPS	Advanced Mobile Phone Service / System
AP	Access point (point d'accès Wifi)
BE	Best Effort
BS	Base Station (Station de base WiMAX)
BSS	Base Service Set
CA	Collision Avoidance
CAC	Call Admission Control
CDMA	Code Division Multiple Access (utilisé par l'UMTS)
CF	Contention Free
CFP	Contention Free Period
CID	Connection Identifier.
CQI	Channel Quality Indicator
CP	Contention Period
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to send
CW	Contention Window
DCF	Distributed Coordination function
DCS	Digital Cellular System (nom donné initialement au GSM opérant dans la bande 1800 MHz)
DECT	Digital Enhanced Cordless Telecommunications
DIFS	Distributed Inter Frame Space
DL-MAP	Down Link Multiple Access Protocol
DSL	Digital Subscriber Line
DSSS	

	Direct Sequence Spread Spectrum
EDCA	Enhanced Distributed Channel Access
EDGE	Enhanced Data rates for GSM Evolution (norme de 2,75G)
EGPRS	Enhanced GPRS
EIFS	Enhanced Inter Frame Space
ertPS	Extended Real-Time Polling Service
ETSI	European Telecommunications Standards Institute (organisme qui publie les normes des réseaux cellulaires 2G et 3G)
ETTD	Équipement Terminal de Traitement de Données (il peut s'agir d'un ordinateur par exemple)
FCH	Frame Control Header
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access (utilisé par le GSM)
FFT	Fast Fourier Transform
FHSS	Frequency-Hopping Spread Spectrum
GPRS	General Packet Radio Service (norme de 2,5G)
GSM	Global System for Mobile communications
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
IEEE	Institut of Electrical and Electronics Engineers
IFS	Inter Frame Space
IP	Internet Protocol
ISM	Industriel, Scientifique, et Médical (bandes de fréquences radio sans licence et à faible puissance émise)
LAN	Local Area Network
LLC	Logical Link Control (sous-couche supérieure de la couche « liaison de donnée » dans le modèle OSI)
MAC	Medium Access Control
MAN	Metropolitan Area Network
ME	Mobile Equipment (le téléphone lui-même, sans la carte SIM)

MICS	Media Independent Command Service
MIES	Media Independent Events Service
MIIS	Media Independent Information Service
MIH	Media Independent Handover
MIMO	Multiple Input Multiple Output
MS	Mobile Station (ensemble telephone + carte SIM)
MSDU	MAC Service Data Unit
NAV	Network Allocation Vector
NMT	Nordic Mobile Telephone (norme de téléphonie mobile analogique de première génération)
NGN	Next Generation Networks
nrtPS	Non Real-Time Polling Service
OFDM	Orthogonal Frequency Division Multiplexing
PAN	Personal Area Network
PCF	Point Coordination Function
PDA	Personal Data Assistance
PDU	Protocol Data Unit
PHY	Physical Layer
PIFS	Priority Inter Frame
QAP	Quality of Service Access Point
QoS	Quality of Service
QSTA	Quality of Service Station
RNIS	Réseau Numérique à Intégration de Services
rtPS	Real-Time Polling Service
RTS	Request to send
SDU	Service Data Unit
SF	Service Flow
SFID	Service Flow Identifier

SMS	Short Message Service (service de messagerie qui permet de transmettre de petits textes)
SSID	Service Set Identifier
STA	Station
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access (utilisé par le GSM et accessoirement par l'UMTS)
TS	Traffic Stream
TSPEC	Traffic Specification
TXOP	Transmission Opportunity
UL-MAP	Up Link Multiple Access Protocol
UMTS	Universal Mobile Telecommunications System (norme de 3G)
UGS	Unsolicited Grant Service
VoIP	Voice over Internet Protocol
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network

ANNEXE

Script NS-3 Pour la Simulation des réseaux Wifi et WiMAX fonctionnant indépendamment :

```
#include "ns3/core-module.h"
#include "ns3/common-module.h"
#include "ns3/node-module.h"
#include "ns3/helper-module.h"
#include "ns3/mobility-module.h"
#include "ns3/contrib-module.h"
#include "ns3/wimax-module.h"
#include <iostream>
#include "ns3/global-route-manager.h"
#include "ns3/constant-position-mobility-model.h"
#include "ns3/random-waypoint-mobility-model.h"
#include "ns3/wifi-module.h"
#include "ns3/vector.h"

NS_LOG_COMPONENT_DEFINE ("ExtensionWifiWimax");

#define MAXSS 1000
#define MAXDIST 10 // km
using namespace ns3;

int main (int argc, char *argv[])
{
    ////////////////////////////////////////////////////////////////////
    // Paramètres de la simulation                                     //
    ////////////////////////////////////////////////////////////////////

    // valeurs par défaut des paramètres de simulation

    // default values
    int nbSS = 4, duration = 2, schedType = 0, nbSta = 4, nbSSVideo = 4, nbStaVideo = 4,
    nbSSVoix = 4, nbSSDonnees = 4, nbStaVoix = 4, nbStaDonnees = 4;
    // Scheduler est un objet d'aide pour la création et l'installation d'un planificateur MAC
    Wimax
    WimaxHelper::SchedulerType scheduler = WimaxHelper::SCHED_TYPE_SIMPLE;
    bool verbose = false;
    double rayonWifi = 400;
    double rayonWimax = 400;

    // Les instructions qui permettent à l'utilisateur de modifier les paramètres de simulation

    CommandLine cmd;
    cmd.AddValue ("nbSS", "le nombre de station WiMAX à créer", nbSS);
    cmd.AddValue ("nbSSVideo", "le nombre de station WiMAX Vidéo à créer", nbSSVideo);
    cmd.AddValue ("nbSSVoix", "le nombre de station WiMAX Voix à créer", nbSSVoix);
    cmd.AddValue ("nbSSDonnees", "le nombre de station WiMAX données à créer", nbSSDonnees);
    cmd.AddValue ("nbSta", "le nombre de station Wifi à créer", nbSta);
    cmd.AddValue ("nbStaVideo", "le nombre de station Wifi Vidéo à créer", nbStaVideo);
    cmd.AddValue ("nbStaVoix", "le nombre de station Wifi Voix à créer", nbStaVoix);
    cmd.AddValue ("nbStaDonnees", "le nombre de station Wifi Données à créer", nbStaDonnees);
    cmd.AddValue ("scheduler", "type de planificateur de connection Wimax", schedType);
    cmd.AddValue ("dure", "duration of the simulation in seconds", duration);
    cmd.AddValue ("rayonWifi", "Le rayon du disc autour du ap-bs dans lequel sont distribué les
stations Wifi", rayonWifi);
    cmd.AddValue ("rayonWimax", "Le rayon du disc autour du ap-bs dans lequel sont distribué les
SSs Wimax", rayonWimax);
    cmd.AddValue ("verbose", "turn on all WimaxNetDevice log components", verbose);
    cmd.Parse (argc, argv);

    /*
    Selon le choix de l'utilisateur, le planificateur peut être soit:
    0 : Un planificateur simple basé à priorité simple premier venu premier servi. Pour des
services temps réels (UGS, rtPS), la bs alloue des grants/polls de façon périodique suivant
```

l'intervalle calculé. Pour les services nrtPS et BE, la bs alloue seulement la bande passante minimale qui reste après l'allocation pour les services temps réels.

1 : Un planificateur type RTPS, similaire au premier sauf pour le flux de service rtPS où l'algorithme essaye de transmettre tous les paquets de tous les flux rtPS quelque soit leur ordre d'arrivée dans la file.

2 : Un planificateur type QoS basé migration, qui utilise trois files d'attente à priorité différente. La file d'attente la moins prioritaire héberge les services BE, l'intermédiaire héberge les services rtps et nrtps et la plus haute héberge les services UGS, certain service d'administration. Les paquets de certains services rtPS/nrtPS peuvent migrer de la file intermédiaire vers la file à haute priorité pour satisfaire leurs exigences en QoS. Ce planificateur essaye de respecter la contrainte de délai des services rtPS et de garantir le débit minimum pour les services rtps/nrtps.

```

*/
switch (schedType)
{
case 0:
    scheduler = WimaxHelper::SCHED_TYPE_SIMPLE;
    break;
case 1:
    scheduler = WimaxHelper::SCHED_TYPE_RTPS;
    break;
case 2:
    scheduler = WimaxHelper::SCHED_TYPE_MBQOS;
    break;
default:
    scheduler = WimaxHelper::SCHED_TYPE_SIMPLE;
}
NS_LOG_INFO ("Fin de l'initialisation des paramètres");

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
// Création de l'infrastructure du réseau
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

    // Création du réseau Wimax

NodeContainer ssNodes; // La liste de noeuds SS WiMAX
NodeContainer bsNodes; // La liste de noeuds BS WiMAX, dans ce cas 1 seule BS

Ptr<SubscriberStationNetDevice> ss[MAXSS]; // Création d'une liste de pointeurs intelligents
vers les objets // de type SubscriberStationNetDevice (SS[i],
avec i=0 à MAXSS)
NetDeviceContainer ssDevs; // La liste des objets de type NetDevice associés aux SS contenus
dans ssNodes
Ipv4InterfaceContainer SSinterfaces; // La liste des interfaces associés aux NetDevice
contenus dans ssDevs

Ptr<BaseStationNetDevice> bs; // La création d'un pointeur intelligent bs vers l'objet de
type BaseStationNetDevice
NetDeviceContainer bsDevsWimax; // La liste de l'objet NetDevice associé à la bsNodes côté
sans fil (WiMAX)
Ipv4InterfaceContainer BSWimaxInterfaces; // la liste des interfaces associées à la
bsDevsWimax

Ptr<SimpleOfdmWimaxChannel> channel; //La création d'un pointeur intelligent vers un objet
de type SimpleOfdmWimaxChannel

Ptr<RandomWaypointMobilityModel> SSPosition[MAXSS]; // La création d'un tableau de pointeur
intelligent vers // l'objet de type
RandomWaypointMobilityModel
Ptr<RandomRectanglePositionAllocator> SSPosAllocator[MAXSS]; //La création d'un tableau de
pointeur intelligent // vers l'objet de type
RandomRectanglePositionAllocator
ssNodes.Create (nbSS); // La création des noeuds SS
bsNodes.Create (1); // La création du noeud bs

WimaxHelper wimax; // Un objet d'aide à l'installation de WimaxNetDevice

channel = CreateObject<SimpleOfdmWimaxChannel> (); // Instanciation d'un objet NS3 de type
SimpleOfdmWimaxChannel (canal Wimax sans fil) qui sera pointé par channel

```

```

channel->SetPropagationModel (SimpleOfdmWimaxChannel::COST231_PROPAGATION); // Choisir le
modèle de propagation pour le canal sans fil Wimax
ssDevs = wimax.Install (ssNodes,
    WimaxHelper::DEVICE_TYPE_SUBSCRIBER_STATION,
    WimaxHelper::SIMPLE_PHY_TYPE_OFDM,
    channel,
    scheduler); // Création et installation des wimaxNetDevices, la
couche physique Wimax, canal wimax sans fil et le planificateur dans les noeuds SSs, avec la
récupération des adresses des wimaxNetDevice créés dans la liste ssDevs
Ptr<WimaxNetDevice> dev = wimax.Install (bsNodes.Get (0),
    WimaxHelper::DEVICE_TYPE_BASE_STATION,
    WimaxHelper::SIMPLE_PHY_TYPE_OFDM,
    channel,
    scheduler); // Création et installation du
wimaxNetDevice, la couche physique Wimax, canal wimax sans fil et le planificateur dans le
noeud BS avec la récupération des adresses des wimaxNetDevice créés dans la liste ssDevs
MobilityHelper mobility; // Un objet d'aide à l'installation du modèle de mobilité
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel",
    "Position", Vector3DValue (Vector (0, 0, 0)));

mobility.Install (bsNodes); // Installation du module de mobilité dans le noeud BS

bsDevsWimax.Add (dev);
if (verbose)
{
    wimax.EnableLogComponents (); // Activer l'affichage du log de tous les objets de type
WimaxNetDevice
}

mobility.SetPositionAllocator ("ns3::UniformDiscPositionAllocator",
    "rho", DoubleValue (rayonWimax),
    "X", DoubleValue (0),
    "Y", DoubleValue (0)
);

mobility.SetMobilityModel ("ns3::RandomWaypointMobilityModel",
    "Speed", RandomVariableValue (UniformVariable (10.3, 40.7)),
    "Pause", RandomVariableValue (ConstantVariable (0.1)));
mobility.Install (ssNodes);
Vector position;
// Boucle pour l'initialisation et l'installation des objets SSPositions dans les noeuds SSs
for (int i = 0; i < nbSS; i++)
{
    ss[i] = ssDevs.Get (i)->GetObject<SubscriberStationNetDevice> ();
    ss[i]->SetModulationType (WimaxPhy::MODULATION_TYPE_QAM16_12);

    SSPosition[i] = ssNodes.Get(i) -> GetObject<RandomWaypointMobilityModel> ();
    position = SSPosition[i]->GetPosition ();
    SSPosAllocator[i] = CreateObject<RandomRectanglePositionAllocator> ();
    SSPosAllocator[i]->SetX (UniformVariable (position.x-rayonWimax/100,
position.x+rayonWimax/100));
    SSPosAllocator[i]->SetY (UniformVariable (position.y-rayonWimax/100,
position.y+rayonWimax/100));
    SSPosition[i]->SetAttribute ("PositionAllocator", PointerValue (SSPosAllocator[i]));
}

//    ssNodes.Get (i)->AggregateObject (SSPosition[i]);

bs = bsDevsWimax.Get (0)->GetObject<BaseStationNetDevice> (); // Récupérer l'adresse de
l'objet BaseStationNetDevice installé dans le noeud bs

InternetStackHelper stack; // Un objet d'aide à l'installation de la pile de protocoles
TCP/IP
stack.Install (bsNodes); // Installation de la pile de protocole TCP/IP dans le le noeud BS
// mobility.Install (ssNodes); // Installation du module de mobilité dans les noeuds SSs
stack.Install (ssNodes); // Installation de la pile de protocole TCP/IP dans les noeuds SSs

Ipv4AddressHelper address; // Un objet d'aide à la distribution des adresses Ipv4
// Distribution des adresses Ipv4 dans le réseau Wimax et récupération des adresses des
objets "Ipv4interface"

```

```

address.SetBase ("10.1.0.0", "255.255.255.0");
BSWimaxInterfaces = address.Assign (bsDevsWimax);
SSInterfaces = address.Assign (ssDevs);

NS_LOG_INFO ("Fin de la création WIMAX");

////////// Création du réseau Wifi //////////

NodeContainer stas; // La liste de noeuds sta WIFI
stas.Create (nbSta);
Ptr<RandomWaypointMobilityModel> StaPosition[MAXSS]; // La création d'un tableau de
pointeur intelligent vers // l'objet de type
RandomWaypointMobilityModel
Ptr<RandomRectanglePositionAllocator> StaPosAllocator[MAXSS]; //La création d'un tableau
de pointeur intelligent // vers l'objet de type
RandomRectanglePositionAllocator

Ipv4InterfaceContainer StaWifiInterfaces; // La liste des interfaces associés aux
NetDevice contenus dans staDevs
Ipv4InterfaceContainer BSWifiInterfaces; // la liste des interfaces associé à la
bsDevsWifi

// Création d'un conteneur des noeuds wifi
NodeContainer infra (bsNodes.Get (0), stas);

YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::YansWifiChannelHelper ();
wifiChannel.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss("ns3::ThreeLogDistancePropagationLossModel");
wifiChannel.AddPropagationLoss("ns3::NakagamiPropagationLossModel",
                               "m0", DoubleValue(1.5),
                               "m1", DoubleValue(1.0),
                               "m2", DoubleValue(1.0));
WifiHelper wifiInfra = WifiHelper::Default ();
QosWifiMacHelper macInfra = QosWifiMacHelper::Default ();
wifiPhy.SetChannel (wifiChannel.Create ());

std::string ssidString("wifi-infra");
Ssid ssid = Ssid (ssidString);
wifiInfra.SetRemoteStationManager ("ns3::OnoeWifiManager");

macInfra.SetType ("ns3::QstaWifiMac",
                 "Ssid", SsidValue (ssid),
                 "ActiveProbing", BooleanValue (false));
NetDeviceContainer staDevices = wifiInfra.Install (wifiPhy, macInfra, stas);

macInfra.SetType ("ns3::QapWifiMac", "Ssid", SsidValue (ssid),
                 "BeaconGeneration", BooleanValue (true),
                 "BeaconInterval", TimeValue (Seconds (2.5)));
NetDeviceContainer apDevices = wifiInfra.Install (wifiPhy, macInfra, bsNodes.Get (0));

NetDeviceContainer infraDevices (apDevices, staDevices);

mobility.SetPositionAllocator ("ns3::UniformDiscPositionAllocator",
                               "rho", DoubleValue (rayonWifi),
                               "x", DoubleValue (0),
                               "y", DoubleValue (0)
                               );

mobility.SetMobilityModel ("ns3::RandomWaypointMobilityModel",
                           "Speed", RandomVariableValue (UniformVariable (10.3, 40.7)),
                           "Pause", RandomVariableValue (ConstantVariable (0.1)));
mobility.Install (stas); // Installation du module de mobilité dans les noeuds Wifi
// Boucle pour l'initialisation et l'intallation des objets SSPositions dans les noeus SSS

```

```

for (int i = 0; i < nbSta; i++)
{
    StaPosition[i] = stas.Get(i) -> GetObject<RandomWaypointMobilityModel> ();
    position = StaPosition[i]->GetPosition ();
    StaPosAllocator[i] = CreateObject<RandomRectanglePositionAllocator> ();
    StaPosAllocator[i]->SetX (UniformVariable (position.x-rayonWifi/100,
position.x+rayonWifi/100));
    StaPosAllocator[i]->SetY (UniformVariable (position.y-rayonWifi/100,
position.y+rayonWifi/100));
    StaPosition[i]->SetAttribute ("PositionAllocator", PointerValue (StaPosAllocator[i]));
}

stack.Install (stas); // Installation de la pile de protocole TCP/IP et affectation
d'adresses
address.SetBase ("9.1.0.0", "255.255.255.0");
BSWifiInterfaces = address.Assign (apDevices);
StaWifiInterfaces = address.Assign (staDevices);

NS_LOG_INFO ("Fin de la création Wifi");

////////// Création du réseau LAN1 //////////

NodeContainer ASNGW_Node; // La liste de noeud qui contient le routeur multicast
ASNGW_Node.Create (1); // La création du noeud routeur multicast

CsmaHelper csmaASN_BS; // Un objet d'aide à l'installation du réseau csma entre la BS et le
retour
// La création du premier réseau LAN comprenant la BS et le routeur
NodeContainer LAN_ASN_BS; // La liste des noeuds BS et routeur

LAN_ASN_BS.Add (bsNodes.Get (0)); // Ajout du noeud BS à la liste LAN_ASN_BS

LAN_ASN_BS.Add (ASNGW_Node.Get (0)); // Ajout du noeud routeur à la liste LAN_ASN_BS

// Configuration des paramètres, création et installation du réseau CSMA entre les noeuds BS
et routeur, les adresses des csmaNetDevice sont récupérés dans la liste LAN_ASN_BS_Devs
csmaASN_BS.SetChannelAttribute ("DataRate", DataRateValue (DataRate (10000000)));
csmaASN_BS.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (1)));
csmaASN_BS.SetDeviceAttribute ("Mtu", UIntegerValue (1500));
NetDeviceContainer LAN_ASN_BS_Devs = csmaASN_BS.Install (LAN_ASN_BS);

NetDeviceContainer BS_CSMADevs; // Liste pour enregistrer l'adresse du csmaNetDevice installé
dans le noeud BS

BS_CSMADevs.Add (LAN_ASN_BS_Devs.Get (0)); // L'ajout de l'adresse du csmaNetDevice de la BS
dans la liste BS_CSMADevs

NetDeviceContainer ASN_Devs1; // Liste pour enregistrer l'adresse du csmaNetDevice installé
dans le noeud routeur coté réseau LAN1
ASN_Devs1.Add (LAN_ASN_BS_Devs.Get (1)); //Ajout du noeud routeur à la liste ASN_Devs1

stack.Install (ASNGW_Node); // Installation de la pile de protocole TCP/IP dans le le noeud
routeur

//Distribution des adresses Ipv4 dans le réseau LAN1 et récupération des adresses des objets
"Ipv4interface"
address.SetBase ("11.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer BSCSMAInterfaces = address.Assign (BS_CSMADevs);
Ipv4InterfaceContainer ASNCMAInterfaces1 = address.Assign (ASN_Devs1);

NS_LOG_INFO ("Fin de la création LAN1");

////////// Création du réseau LAN2 //////////

NodeContainer Streamer_Node; // La liste de noeud qui contient la source de diffusion
multicast

```

```

    Streamer_Node.Create (nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix+nbSSDonnees+nbStaDonnees); //
La création du noeud diffuseur multicast

    CsmHelper csmaStreamer_ASN; // Un objet d'aide à l'installation du réseau csma entre le
retour et le diffuseur multicast
    // La création du deuxième réseau LAN comprenant le routeur et le diffuseur multicast

    NodeContainer LAN_ASN_STREAMER; // La liste des noeuds routeur et diffuseur multicast
LAN_ASN_STREAMER.Add (ASNGW_Node.Get (0)); // Ajout du noeud diffuseur multicast à la liste
LAN_ASN_STREAMER
LAN_ASN_STREAMER.Add (Streamer_Node);

    // Configuration des paramètres, création et installation du réseau CSMA entre les noeuds le
diffuseur multicast et le routeur, les adresses des csmaNetDevice sont récupérés dans la liste
LLAN_ASN_STREAMER_Devs
    csmaStreamer_ASN.SetChannelAttribute ("DataRate", DataRateValue (DataRate (10000000)));
    csmaStreamer_ASN.SetChannelAttribute ("Delay", TimeValue (Milliseconds (1)));
    csmaStreamer_ASN.SetDeviceAttribute ("Mtu", UIntegerValue (1500));

    NetDeviceContainer LAN_ASN_STREAMER_Devs = csmaStreamer_ASN.Install (LAN_ASN_STREAMER);

    NetDeviceContainer STREAMER_Devs; // Liste pour enregistrer l'adresse du csmaNetDevice
installé dans le noeud diffuseur multicast
    NetDeviceContainer ASN_Devs2; // Liste pour enregistrer l'adresse du csmaNetDevice installé
dans le noeud routeur côté réseau LAN2
    ASN_Devs2.Add (LAN_ASN_STREAMER_Devs.Get (0)); //Ajout du noeud routeur à la liste
ASN_Devs2

    for (int i = 0; i < nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix+nbSSDonnees+nbStaDonnees; i++)
    {
        STREAMER_Devs.Add (LAN_ASN_STREAMER_Devs.Get (i+1)); // L'ajout de l'adresse du
csmaNetDevice du diffuseur multicast dans la liste STREAMER_Devs
    }

    stack.Install (Streamer_Node); // Installation de la pile de protocole TCP/IP dans le le
noeud diffuseur multicast

    //Distribution des adresses Ipv4 dans le réseau LAN2 et récupération des adresses des objets
"Ipv4interface"
    address.SetBase ("12.1.1.0", "255.255.255.0");
    Ipv4InterfaceContainer ASNCSMASInterfaces2 = address.Assign (ASN_Devs2);
    Ipv4InterfaceContainer StreamerCSMASInterfaces = address.Assign (STREAMER_Devs);

    NS_LOG_INFO ("Fin de la création LAN2");

    ////////////////////////////////////////
    //                                     //
    // Création des applications et configuration des roueurs //
    //                                     //
    ////////////////////////////////////////

    UdpServerHelper udpServer; //Un objet d'aide à la création d'applications type udpserveur
ApplicationContainer serverApps[MAXSS]; //La liste des applications type udpServer
(recepteur udp)intallé dans les noeuds SS[i]
    UdpTraceClientHelper udpClient[MAXSS]; // Un objet d'aide à la création d'applications type
TraceCleint (emetteur udp avec etiquette de temps)
    ApplicationContainer clientApps[MAXSS]; //La liste des applications type udpTraceClient

    ApplicationContainer serverAppsWifi[MAXSS]; //La liste des applications type udpServer
(recepteur udp)intallé dans les noeuds SS[i]
    UdpTraceClientHelper udpClientWifi[MAXSS]; // Un objet d'aide à la création d'applications
type TraceCleint (emetteur udp avec etiquette de temps)
    ApplicationContainer clientAppsWifi[MAXSS]; //La liste des applications type udpTraceClient

    LogComponentEnable ("UdpTraceClient", LOG_LEVEL_INFO); // Activer l'affichage du log des
objets de type UdpTraceClient
    LogComponentEnable ("UdpServer", LOG_LEVEL_INFO); // Activer l'affichage du log des objets de
type UdpServer

    uint16_t stream_port = 100;
    udpServer = UdpServerHelper (stream_port); //Objet d'aide à la création et l'installation
d'une application de type udpServer (recepteur udp)
    //L'installation d'une application de type udpServer dans tous les noeuds SSS

```

```

for (int i = 0; i < nbSS; i++)
{
    serverApps[i] = udpServer.Install (ssNodes.Get (i));
    serverApps[i].Start (Seconds (1));
    serverApps[i].Stop (Seconds (duration+1));
}
for (int i = 0; i < nbSta; i++)
{
    serverAppsWifi[i] = udpServer.Install (stas.Get (i));
    serverAppsWifi[i].Start (Seconds (1));
    serverAppsWifi[i].Stop (Seconds (duration+1));
}

//L'installation d'une application de type udpTraceClient dans le noeud diffuseur
for (int i = 0; i < nbSSVideo; i++)
{
    udpClient[i] = UdpTraceClientHelper (SSinterfaces.GetAddress(i%nbSS), stream_port,
"Verbose_Jurassic.dat"); //Objet d'aide à la
    // création et l'installation d'une application de type udpTraceClient(emetteur udp)
    clientApps[i] = udpClient[i].Install (Streamer_Node.Get (i));
    clientApps[i].Start (Seconds (1));
    clientApps[i].Stop (Seconds (duration));
}
for (int i = 0; i < nbStaVideo; i++)
{
    udpClientWifi[i] = UdpTraceClientHelper (StaWifiinterfaces.GetAddress(i%nbSta),
stream_port, "Verbose_Jurassic.dat",UP_VI); //Objet d'aide à la
    // création et l'installation d'une application de type udpTraceClient(emetteur udp)
    clientAppsWifi[i] = udpClientWifi[i].Install (Streamer_Node.Get (i+nbSSVideo));
    clientAppsWifi[i].Start (Seconds (1));
    clientAppsWifi[i].Stop (Seconds (duration));
}
for (int i = 0; i < nbSSVoix; i++)
{
    udpClient[i+ nbSSVideo] = UdpTraceClientHelper (SSinterfaces.GetAddress(i%nbSS),
stream_port, "voip.dat"); //Objet d'aide à la
    // création et l'installation d'une application de type udpTraceClient(emetteur udp)
    clientApps[i+ nbSSVideo] = udpClient[i+ nbSSVideo].Install (Streamer_Node.Get
(i+nbSSVideo+nbStaVideo));
    clientApps[i+ nbSSVideo].Start (Seconds (1));
    clientApps[i+ nbSSVideo].Stop (Seconds (duration));
}
for (int i = 0; i < nbStaVoix; i++)
{
    udpClientWifi[i+nbStaVideo] = UdpTraceClientHelper
(StaWifiinterfaces.GetAddress(i%nbSta), stream_port, "voip.dat",UP_VO); //Objet d'aide à la
    // création et l'installation d'une application de type udpTraceClient(emetteur udp)
    clientAppsWifi[i+nbStaVideo] = udpClientWifi[i+nbStaVideo].Install
(Streamer_Node.Get (i+nbSSVideo+nbStaVideo+nbSSVoix));
    clientAppsWifi[i+nbStaVideo].Start (Seconds (1));
    clientAppsWifi[i+nbStaVideo].Stop (Seconds (duration));
}
for (int i = 0; i < nbSSDonnees; i++)
{
    udpClient[i+ nbSSVideo +nbSSVoix] = UdpTraceClientHelper
(SSinterfaces.GetAddress(i%nbSS), stream_port, "data.dat"); //Objet d'aide à la
    // création et l'installation d'une application de type udpTraceClient(emetteur udp)
    clientApps[i+ nbSSVideo +nbSSVoix] = udpClient[i+ nbSSVideo +nbSSVoix].Install
(Streamer_Node.Get (i+nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix));
    clientApps[i+ nbSSVideo +nbSSVoix].Start (Seconds (1));
    clientApps[i+ nbSSVideo +nbSSVoix].Stop (Seconds (duration));
}
for (int i = 0; i < nbStaDonnees; i++)
{
    udpClientWifi[i+nbStaVideo+nbStaVoix] = UdpTraceClientHelper
(StaWifiinterfaces.GetAddress(i%nbSta), stream_port, "data.dat",UP_BE); //Objet d'aide à la
    // création et l'installation d'une application de type udpTraceClient(emetteur udp)
    clientAppsWifi[i+nbStaVideo+nbStaVoix] =
udpClientWifi[i+nbStaVideo+nbStaVoix].Install (Streamer_Node.Get
(i+nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix+nbSSDonnees));
    clientAppsWifi[i+nbStaVideo+nbStaVoix].Start (Seconds (1));
    clientAppsWifi[i+nbStaVideo+nbStaVoix].Stop (Seconds (duration));
}

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

```

```

// Création d'un enregistrement de classification dans la couche CS Wimax
IpcsClassifierRecord MulticastClassifier[MAXSS];
ServiceFlow MulticastServiceFlow[MAXSS];
for (int i = 0; i < nbSSVideo; i++)
{
    MulticastClassifier[i] =IpcsClassifierRecord (StreamerCSMAInterfaces.GetAddress(i),
        Ipv4Mask ("255.255.255.255"),
        SSinterfaces.GetAddress(i%nbSS),
        Ipv4Mask ("255.255.255.255"),
        0,
        65000,
        stream_port,
        stream_port,
        17,
        1);

    //Création d'un flux de service dans le lien descendant de type rtPS associé avec
les paquets qui correspondent à
    // l'enregistrement de classification MulticastClassifier
    MulticastServiceFlow[i] = wimax.CreateServiceFlow (ServiceFlow::SF_DIRECTION_DOWN,
ServiceFlow::SF_TYPE_RTPTS,
        MulticastClassifier[i]);

    //Ajouter le flux de service créé dans la BS
    ss[i%nbSS]->AddServiceFlow (MulticastServiceFlow[i]);
}
NS_LOG_INFO ("Fin de la création des FS vidéos");

for (int i = 0; i < nbSSVoix; i++)
{
    MulticastClassifier[i+nbSSVideo] =IpcsClassifierRecord
(StreamerCSMAInterfaces.GetAddress(i+nbSSVideo+nbStaVideo),
        Ipv4Mask ("255.255.255.255"),
        SSinterfaces.GetAddress(i%nbSS),
        Ipv4Mask ("255.255.255.255"),
        0,
        65000,
        stream_port,
        stream_port,
        17,
        1);

    //Création d'un flux de service dans le lien descendant de type UGS associé avec les
paquets qui correspondent à
    // l'enregistrement de classification MulticastClassifier
    MulticastServiceFlow[i+nbSSVideo] = wimax.CreateServiceFlow
(ServiceFlow::SF_DIRECTION_DOWN,
ServiceFlow::SF_TYPE_UGS,
        MulticastClassifier[i+nbSSVideo]);

    //Ajouter le flux de service créé dans la BS
    ss[i%nbSS]->AddServiceFlow (MulticastServiceFlow[i+nbSSVideo]);
}
NS_LOG_INFO ("Fin de la création des FS voix");

for (int i = 0; i < nbSSDonnees; i++)
{
    MulticastClassifier[i+nbSSVideo+nbSSVoix] =IpcsClassifierRecord
(StreamerCSMAInterfaces.GetAddress(i+nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix),
        Ipv4Mask ("255.255.255.255"),
        SSinterfaces.GetAddress(i%nbSS),
        Ipv4Mask ("255.255.255.255"),
        0,
        65000,
        stream_port,
        stream_port,
        17,
        1);

    //Création d'un flux de service dans le lien descendant de type BE associé avec les
paquets qui correspondent à
    // l'enregistrement de classification MulticastClassifier
    MulticastServiceFlow[i+nbSSVideo+nbSSVoix] = wimax.CreateServiceFlow
(ServiceFlow::SF_DIRECTION_DOWN,
        ServiceFlow::SF_TYPE_BE,

```

```

MulticastClassifier[i+nbSSVideo+nbSSVoix]);
    //Ajouter le flux de service créé dans la BS
    ss[i%nbSS]->AddServiceFlow (MulticastServiceFlow[i+nbSSVideo+nbSSVoix]);
}

NS_LOG_INFO ("Fin de la création des FS Données");

NS_LOG_INFO ("Fin de la création des routes et FSs");

////////////////////////////////////
//
//          Lancement de la simulation          //
//
////////////////////////////////////

NS_LOG_INFO ("Fin de la configuration");
Simulator::Stop (Seconds (duration + 1.5));
NS_LOG_INFO ("Starting simulation.....");
Simulator::Run ();

for (int i = 0; i < nbSS; i++)
{
    ss[i] = 0;
    SSPosition[i] = 0;
    SSPosAllocator[i] = 0;
}
for (int i = 0; i < nbSta; i++)
{
    StaPosition[i] = 0;
    StaPosAllocator[i] = 0;
}

bs = 0;

Simulator::Destroy ();
NS_LOG_INFO ("Done.");

return 0;
}

```

Script NS-3 Pour la Simulation du réseau mixte :

```
#include "ns3/core-module.h"
#include "ns3/common-module.h"
#include "ns3/node-module.h"
#include "ns3/helper-module.h"
#include "ns3/mobility-module.h"
#include "ns3/contrib-module.h"
#include "ns3/wimax-module.h"
#include <iostream>
#include "ns3/global-route-manager.h"
#include "ns3/constant-position-mobility-model.h"
#include "ns3/random-waypoint-mobility-model.h"
#include "ns3/wifi-module.h"
#include "ns3/vector.h"

NS_LOG_COMPONENT_DEFINE ("ExtensionWifiWimax_Mixte");

#define MAXSS 1000
#define MAXDIST 10 // km
using namespace ns3;

int main (int argc, char *argv[])
{
    ////////////////////////////////////////////////////////////////////
    // Paramètres de la simulation                                     //
    ////////////////////////////////////////////////////////////////////

    // valeurs par défaut des paramètres de simulation

    int nbNoueds = 3, nbStaMax=4, duration = 2, schedType = 0, nbVideo = 3, nbVoix =3, nbDonnees
= 3;
    // Sceduler est un objet d'aide pour la création et l'installation d'un planificateur MAC
Wimax
    WimaxHelper::SchedulerType scheduler = WimaxHelper::SCHED_TYPE_SIMPLE;
    bool verbose = false;
    double rayonWifiWimax = 100;
    double rayonLimiteQos = 250;
    // Les instructions qui permettent à l'utilisateur de modifier les paramètres de simulation
    CommandLine cmd;
    cmd.AddValue ("nbNoueds", "number of subscriber station to create", nbNoueds);
    cmd.AddValue ("nbStaMax", "number of subscriber station to create", nbStaMax);
    cmd.AddValue ("nbVideo", "number of subscriber station to create", nbVideo);
    cmd.AddValue ("nbVoix", "number of subscriber station to create", nbVoix);
    cmd.AddValue ("nbDonnees", "number of subscriber station to create", nbDonnees);
    cmd.AddValue ("scheduler", "type of scheduler to use with the netdevices", schedType);
    cmd.AddValue ("dure", "duration of the simulation in seconds", duration);
    cmd.AddValue ("rayonWifiWimax", "Le rayon du disc autour du ap-bs dans lequel sont distribué
les stations Wifi", rayonWifiWimax);
    cmd.AddValue ("rayonLimiteQos", "Le rayon du disc autour du ap-bs dans lequel sont distribué
les SSs Wimax", rayonLimiteQos);
    cmd.AddValue ("verbose", "turn on all WimaxNetDevice log components", verbose);
    cmd.Parse (argc, argv);

    /*
    Selon le choix de l'utilisateur, le planificateur peut être soit:
    0 : Un planificateur simple basé à priorité simple premier venu premier servi. Pour des
services temps réels (UGS, rtPS), la bs alloue des grants/polls de façon périodique suivant
l'intervalle calculé. Pour les services nrtPS et BE, la bs alloue seulement la bande passante
minimale qui reste après l'allocation pour les services temps réels.
    1 : Un planificateur type RTPS, similaire au premier sauf pour le flux de service rtPS
où l'algorithmme essay de transmettre tous les paquets de tous les flux rtPS quelque soit leur
ordre d'arrivé dans la file.
    2 : Un planificateur type QoS basé migration, qui utilise trois files d'attente à
priorité différente. La file d'attente la moins prioritaire héberge les services BE,
l'intermédiaire héberge les services rtps et nrtps et la plus haute héberge les services UGS,
certain service d'administration. Les paquets de certains services rtPS/nrtPS peuvent migrer de
la file intermédiaire vers la file à haute priorité pour satisfaire leurs exigence en QoS. Ce
```

planificateur essaie de respecter la contrainte de délai des services rtPS et de garantir le débit minimum pour les services rtps/nrtps.

```

*/
switch (schedType)
{
case 0:
    scheduler = WimaxHelper::SCHED_TYPE_SIMPLE;
    break;
case 1:
    scheduler = WimaxHelper::SCHED_TYPE_RTPTS;
    break;
case 2:
    scheduler = WimaxHelper::SCHED_TYPE_MBQOS;
    break;
default:
    scheduler = WimaxHelper::SCHED_TYPE_SIMPLE;
}
NS_LOG_INFO ("Fin de l'initialisation des paramètres");

////////////////////////////////////
//
// Création de l'infrastructure du réseau
//
//
////////////////////////////////////

////////// Création du réseau mixte //////////

NodeContainer noueds; // La liste des noeuds mixtes Wifi Wimax
noueds.Create(nbNoueds); // Création des noeuds

NodeContainer ssNodes; // La liste de noeuds mixte où l'interface Wimax est activée
NodeContainer bsNodes; // La liste de noeuds BS WiMAX, dans ce cas 1 seule BS

Ptr<SubscriberStationNetDevice> ss[MAXSS]; // Création d'une liste de pointeurs intelligent
vers les objets
// de type SubscriberStationNetDevice (SS[i],
avec i=0 à MAXSS)
NetDeviceContainer ssDevs; // La liste des objets de type NetDevice associés aux SS contenus
dans ssNodes
Ipv4InterfaceContainer SSinterfaces; // La liste des interfaces associés aux NetDevice
contenus dans ssDevs

Ptr<BaseStationNetDevice> bs; // La création d'un pointeur intelligent bs vers l'objet de
type BaseStationNetDevice
NetDeviceContainer bsDevsWimax; // La liste de l'objet NetDevice associé à la bsNodes côté
sans fil (WiMAX)
Ipv4InterfaceContainer BSWimaxInterfaces; // la liste des interfaces associé à la
bsDevsWimax

Ptr<SimpleOfdmWimaxChannel> channel; //La création d'un pointeur intelligent de type
SimpleOfdmWimaxChannel

Ptr<RandomWaypointMobilityModel> SSPosition[MAXSS]; // La création d'un tableau de pointeur
intelligent vers
// l'objet de type
RandomWaypointMobilityModel
Ptr<RandomRectanglePositionAllocator> SSPosAllocator[MAXSS]; //La création d'un tableau de
pointeur intelligent
// vers l'objet de type
RandomRectanglePositionAllocator
NodeContainer stas; // La liste de noeuds mixtes où l'interface Wifi est activée

MobilityHelper mobility; // Un objet d'aide à l'installation du modèle de mobilité
mobility.SetPositionAllocator ("ns3::UniformDiscPositionAllocator",
    "rho", DoubleValue(rayonWifiWimax),
    "X", DoubleValue(0),
    "Y", DoubleValue(0)
);

mobility.SetMobilityModel ("ns3::RandomWaypointMobilityModel",
    "Speed", RandomVariableValue (UniformVariable (10.3, 40.7)),
    "Pause", RandomVariableValue (ConstantVariable (0.1)));
mobility.Install (noueds);
Vector position;
double distanceAuBS=0;

```

```

int nbSS=0;
int nbSta=0;
// Boucle pour l'initialisation et l'installation des objets SSPositions dans les noeuds
mixtes
//et activation de l'interface Wifi ou Wimax selon l'algorithme de couplage d'admission
for (int i = 0; i < nbNoeuds; i++)
{
    SSPosition[i] = noeuds.Get(i) -> GetObject<RandomWaypointMobilityModel> ();
    position = SSPosition[i]->GetPosition ();
    SSPosAllocator[i] = CreateObject<RandomRectanglePositionAllocator> ();
    SSPosAllocator[i]->SetX (UniformVariable (position.x-rayonWifiWimax/100,
position.x+rayonWifiWimax/100));
    SSPosAllocator[i]->SetY (UniformVariable (position.y-rayonWifiWimax/100,
position.y+rayonWifiWimax/100));
    SSPosition[i]->SetAttribute ("PositionAllocator", PointerValue (SSPosAllocator[i]));
    distanceAuBS=(position.x*position.x)+(position.y*position.y);
    if (distanceAuBS<=(rayonLimiteQos*rayonLimiteQos))
    {
        if (nbSta<nbStaMax)
        {
            stas.Add(noeuds.Get(i));
            nbSta++;
        }
        else{
            ssNodes.Add(noeuds.Get(i));
            nbSS++;
        }
    }
    else {
        ssNodes.Add(noeuds.Get(i));
        nbSS++;
    }
}

int nbSSVideo = 0, nbStaVideo = 0, nbSSVoix = 0, nbSSDonnees = 0, nbStaVoix = 0,
nbStaDonnees = 0;
nbStaVideo=nbVideo/nbNoeuds*nbSta;
nbStaVoix=nbVoix/nbNoeuds*nbSta;
nbStaDonnees=nbDonnees/nbNoeuds*nbSta;
nbSSVideo=nbVideo/nbNoeuds*nbSS;
nbSSVoix=nbVoix/nbNoeuds*nbSS;
nbSSDonnees=nbDonnees/nbNoeuds*nbSS;
NS_LOG_INFO ("NbrNoeudsWifi="+nbSta);
NS_LOG_INFO ("NbrNoeudsWimax="+nbSS);
NS_LOG_INFO ("NbrVideoWifi="+nbStaVideo);
NS_LOG_INFO ("NbrVideoWimax="+nbSSVideo);

bsNodes.Create (1); // La création du noeud bs

WimaxHelper wimax; // Un objet d'aide à l'installation de WimaxNetDevice

channel = CreateObject<SimpleOfdmWimaxChannel> ();// Instanciation d'un objet NS3 de type
SimpleOfdmWimaxChannel (canal Wimax sans fil) qui sera pointé par channel
channel->SetPropagationModel (SimpleOfdmWimaxChannel::COST231_PROPAGATION);// Choisir le
modèle de propagation pour le canal sans fil Wimax
ssDevs = wimax.Install (ssNodes,
    WimaxHelper::DEVICE_TYPE_SUBSCRIBER_STATION,
    WimaxHelper::SIMPLE_PHY_TYPE_OFDM,
    channel,
    scheduler);// Création et installation des wimaxNetDevices, la
couche physique Wimax, canal wimax sans fil et le planificateur dans les noeuds SSs, avec la
récupération des adresses des wimaxNetDevice créés dans la liste ssDevs
Ptr<WimaxNetDevice> dev = wimax.Install (bsNodes.Get (0),
    WimaxHelper::DEVICE_TYPE_BASE_STATION,
    WimaxHelper::SIMPLE_PHY_TYPE_OFDM,
    channel,
    scheduler);// Création et installation du
wimaxNetDevice, la couche physique Wimax, canal wimax sans fil et le planificateur dans le
noeud BS avec la récupération des adresses des wimaxNetDevice créés dans la liste ssDevs
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel",
    "Position", Vector3DValue (Vector (0, 0, 0)));

mobility.Install (bsNodes); // Installation du module de mobilité dans le noeud BS

```

```

bsDevsWimax.Add (dev);
if (verbose)
{
    wimax.EnableLogComponents (); //Activer l'affichage du log de tous les objets de type
WimaxNetDevice
}

for (int i = 0; i < nbSS; i++)
{
    ss[i] = ssDevs.Get (i)->GetObject<SubscriberStationNetDevice> ();
    ss[i]->SetModulationType (WimaxPhy::MODULATION_TYPE_QAM16_12);
}

bs = bsDevsWimax.Get (0)->GetObject<BaseStationNetDevice> (); // Récupérer l'adresse de
l'objet BaseStationNetDevice installé dans le noeub bs

InternetStackHelper stack; // Un objet d'aide à l'installation de la pile de protocoles
TCP/IP
stack.Install (bsNodes); // Installation de la pile de protocole TCP/IP dans le le noeud BS
// mobility.Install (ssNodes); // Installation du module de mobilité dans les noeuds SSs
stack.Install (ssNodes); // Installation de la pile de protocole TCP/IP dans les noeuds SSs

Ipv4AddressHelper address; // Un objet d'aide à la distribution des adresses Ipv4
//Distribution des adresses Ipv4 dans le réseau Wimax et récupération des adresses des
objets "Ipv4interface"
address.SetBase ("10.1.0.0", "255.255.255.0");
BSWimaxInterfaces = address.Assign (bsDevsWimax);
SSinterfaces = address.Assign (ssDevs);

Ipv4InterfaceContainer StaWifiinterfaces; // La liste des interfaces associés aux NetDevice
contenus dans ssDevs
Ipv4InterfaceContainer BSWifiInterfaces; // la liste des interfaces associé à la bsDevsWimax

NodeContainer infra (bsNodes.Get (0), stas);

YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::YansWifiChannelHelper ();
wifiChannel.SetPropagationDelay("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss("ns3::ThreeLogDistancePropagationLossModel");
wifiChannel.AddPropagationLoss("ns3::NakagamiPropagationLossModel",
                               "m0", DoubleValue(1.5),
                               "m1", DoubleValue(1.0),
                               "m2", DoubleValue(1.0));

WifiHelper wifiInfra = WifiHelper::Default ();
QosWifiMacHelper macInfra = QosWifiMacHelper::Default ();
wifiPhy.SetChannel (wifiChannel.Create ());

std::string ssidString("wifi-infra");
Ssid ssid = Ssid (ssidString);
wifiInfra.SetRemoteStationManager ("ns3::OnoeWifiManager");

macInfra.SetType ("ns3::QstaWifiMac",
                 "Ssid", SsidValue (ssid),
                 "ActiveProbing", BooleanValue (false));
NetDeviceContainer staDevices = wifiInfra.Install (wifiPhy, macInfra, stas);

macInfra.SetType ("ns3::QapWifiMac", "Ssid", SsidValue (ssid),
                 "BeaconGeneration", BooleanValue (true),
                 "BeaconInterval", TimeValue (Seconds (2.5)));
NetDeviceContainer apDevices = wifiInfra.Install (wifiPhy, macInfra, bsNodes.Get (0));

NetDeviceContainer infraDevices (apDevices, staDevices);

stack.Install (stas);

address.SetBase ("9.1.0.0", "255.255.255.0");
BSWifiInterfaces = address.Assign (apDevices);
StaWifiinterfaces = address.Assign (staDevices);

NS_LOG_INFO ("Fin de la création du réseau mixte");

////////// Création du réseau LAN1 //////////

```

```

NodeContainer ASNGW_Node; // La liste de noeud qui contient le routeur multicast

ASNGW_Node.Create (1); // La création du noeud routeur multicast

CsmaHelper csmaASN_BS; // Un objet d'aide à l'installation du réseau csma entre la BS et le
retour
// La création du premier réseau LAN comprenant la BS et le routeur
NodeContainer LAN_ASN_BS; // La liste des noeuds BS et routeur

LAN_ASN_BS.Add (bsNodes.Get (0)); // Ajout du noeud BS à la liste LAN_ASN_BS

LAN_ASN_BS.Add (ASNGW_Node.Get (0)); // Ajout du noeud routeur à la liste LAN_ASN_BS

// Configuration des paramètres, création et installation du réseau CSMA entre les noeuds BS
et routeur, les adresses des csmaNetDevice sont récupérés dans la liste LAN_ASN_BS_Devs
csmaASN_BS.SetChannelAttribute ("DataRate", DataRateValue (DataRate (100000000)));
csmaASN_BS.SetChannelAttribute ("Delay", TimeValue (Milliseconds (1)));
csmaASN_BS.SetDeviceAttribute ("Mtu", UIntegerValue (1500));
NetDeviceContainer LAN_ASN_BS_Devs = csmaASN_BS.Install (LAN_ASN_BS);

NetDeviceContainer BS_CSMADevs; // Liste pour enregistrer l'adresse du csmaNetDevice installé
dans le noeud BS

BS_CSMADevs.Add (LAN_ASN_BS_Devs.Get (0)); // L'ajout de l'adresse du csmaNetDevice de la BS
dans la liste BS_CSMADevs

NetDeviceContainer ASN_Devs1; // Liste pour enregistrer l'adresse du csmaNetDevice installé
dans le noeud routeur coté réseau LAN1
ASN_Devs1.Add (LAN_ASN_BS_Devs.Get (1)); //Ajout du noeud routeur à la liste ASN_Devs1

stack.Install (ASNGW_Node); // Installation de la pile de protocole TCP/IP dans le le noeud
routeur

//Distribution des adresses Ipv4 dans le réseau LAN1 et récupération des adresses des objets
"Ipv4interface"
address.SetBase ("11.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer BSCSMAInterfaces = address.Assign (BS_CSMADevs);
Ipv4InterfaceContainer ASNCMAInterfaces1 = address.Assign (ASN_Devs1);

NS_LOG_INFO ("Fin de la création LAN1");

////////// Création du réseau LAN2 //////////

NodeContainer Streamer_Node; // La liste de noeud qui contient la source de diffusion
multicast
Streamer_Node.Create (nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix+nbSSDonnees+nbStaDonnees); //
La création du noeud diffuseur multicast

CsmaHelper csmaStreamer_ASN; // Un objet d'aide à l'installation du réseau csma entre le
retour et le diffuseur multicast
// La création du deuxième réseau LAN comprenant le routeur et le diffuseur multicast

NodeContainer LAN_ASN_STREAMER; // La liste des noeuds routeur et diffuseur multicast
LAN_ASN_STREAMER.Add (ASNGW_Node.Get (0)); // Ajout du noeud diffuseur multicast à la liste
LAN_ASN_STREAMER
LAN_ASN_STREAMER.Add (Streamer_Node);

// Configuration des paramètres, création et installation du réseau CSMA entre les noeuds le
diffuseur multicast et le routeur, les adresses des csmaNetDevice sont récupérés dans la liste
LIAN_ASN_STREAMER_Devs
csmaStreamer_ASN.SetChannelAttribute ("DataRate", DataRateValue (DataRate (100000000)));
csmaStreamer_ASN.SetChannelAttribute ("Delay", TimeValue (Milliseconds (1)));
csmaStreamer_ASN.SetDeviceAttribute ("Mtu", UIntegerValue (1500));

NetDeviceContainer LAN_ASN_STREAMER_Devs = csmaStreamer_ASN.Install (LAN_ASN_STREAMER);

NetDeviceContainer STREAMER_Devs; // Liste pour enregistrer l'adresse du csmaNetDevice
installé dans le noeud diffuseur multicast
NetDeviceContainer ASN_Devs2; // Liste pour enregistrer l'adresse du csmaNetDevice installé
dans le noeud routeur côté réseau LAN2
ASN_Devs2.Add (LAN_ASN_STREAMER_Devs.Get (0)); //Ajout du noeud routeur à la liste
ASN_Devs2

```

```

for (int i = 0; i < nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix+nbSSDonnees+nbStaDonnees; i++)
{
    STREAMER_Devs.Add (LAN_ASN_STREAMER_Devs.Get (i+1)); // L'ajout de l'adresse du
csmaNetDevice du diffuseur multicast dans la liste STREAMER_Devs
}

stack.Install (Streamer_Node); // Installation de la pile de protocole TCP/IP dans le le
noeud diffuseur multicast

//Distribution des adresses Ipv4 dans le réseau LAN2 et récupération des adresses des objets
"Ipv4interface"
address.SetBase ("12.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer ASNCSMASInterfaces2 = address.Assign (ASN_Devs2);
Ipv4InterfaceContainer StreamerCSMAInterfaces = address.Assign (STREAMER_Devs);

NS_LOG_INFO ("Fin de la création LAN2");

////////////////////////////////////
//
// Création des applications et configuration des roueurs
//
//
////////////////////////////////////

UdpServerHelper udpServer; //Un objet d'aide à la création d'applications type udpserveur
ApplicationContainer serverApps[MAXSS]; //La liste des applications type udpserveur
(recepteur udp)intallé dans les noeuds SS[i]
UdpTraceClientHelper udpClient[MAXSS]; // Un objet d'aide à la création d'applications type
TraceCleint (emetteur udp avec etiquette de temps)
ApplicationContainer clientApps[MAXSS]; //La liste des applications type udpTraceClient

ApplicationContainer serverAppsWifi[MAXSS]; //La liste des applications type udpserveur
(recepteur udp)intallé dans les noeuds SS[i]
UdpTraceClientHelper udpClientWifi[MAXSS]; // Un objet d'aide à la création d'applications
type TraceCleint (emetteur udp avec etiquette de temps)
ApplicationContainer clientAppsWifi[MAXSS]; //La liste des applications type udpTraceClient

LogComponentEnable ("UdpTraceClient", LOG_LEVEL_INFO); // Activer l'affichage du log des
objets de type UdpTraceClient
LogComponentEnable ("UdpServer", LOG_LEVEL_INFO); // Activer l'affichage du log des objets de
type UdpServer

uint16_t stream_port = 100;
udpServer = UdpServerHelper (stream_port); //Objet d'aide à la création et l'installation
d'une application de type udpServer (recepteur udp)
//L'installation d'une application de type udpServer dans tous les noeuds SSs
for (int i = 0; i < nbSS; i++)
{
    serverApps[i] = udpServer.Install (ssNodes.Get (i));
    serverApps[i].Start (Seconds (1));
    serverApps[i].Stop (Seconds (duration+1));
}
for (int i = 0; i < nbSta; i++)
{
    serverAppsWifi[i] = udpServer.Install (stas.Get (i));
    serverAppsWifi[i].Start (Seconds (1));
    serverAppsWifi[i].Stop (Seconds (duration+1));
}

//L'installation d'une application de type udpTraceClient dans le noeud diffuseur
for (int i = 0; i < nbSSVideo; i++)
{
    udpClient[i] = UdpTraceClientHelper (SSinterfaces.GetAddress(i%nbSS), stream_port,
"Verbose_Jurassic.dat"); //Objet d'aide à la
// création et l'installation d'une application de type udpTraceClient(emetteur udp)
    clientApps[i] = udpClient[i].Install (Streamer_Node.Get (i));
    clientApps[i].Start (Seconds (1));
    clientApps[i].Stop (Seconds (duration));
}
for (int i = 0; i < nbStaVideo; i++)

```

```

    {
        udpClientWifi[i] = UdpTraceClientHelper (StaWifiinterfaces.GetAddress(i%nbSta),
stream_port, "Verbose_Jurassic.dat",UP_VI); //Objet d'aide à la
        // création et l'installation d'une application de type udpTraceClient(emetteur udp)
        clientAppsWifi[i] = udpClientWifi[i].Install (Streamer_Node.Get (i+nbSSVideo));
        clientAppsWifi[i].Start (Seconds (1));
        clientAppsWifi[i].Stop (Seconds (duration));
    }
    for (int i = 0; i < nbSSVoix; i++)
    {
        udpClient[i+ nbSSVideo] = UdpTraceClientHelper (SSinterfaces.GetAddress(i%nbSS),
stream_port, "voip.dat"); //Objet d'aide à la
        // création et l'installation d'une application de type udpTraceClient(emetteur udp)
        clientApps[i+ nbSSVideo] = udpClient[i+ nbSSVideo].Install (Streamer_Node.Get
(i+nbSSVideo+nbStaVideo));
        clientApps[i+ nbSSVideo].Start (Seconds (1));
        clientApps[i+ nbSSVideo].Stop (Seconds (duration));
    }
    for (int i = 0; i < nbStaVoix; i++)
    {
        udpClientWifi[i+nbStaVideo] = UdpTraceClientHelper
(StaWifiinterfaces.GetAddress(i%nbSta), stream_port, "voip.dat",UP_VO); //Objet d'aide à la
        // création et l'installation d'une application de type udpTraceClient(emetteur udp)
        clientAppsWifi[i+nbStaVideo] = udpClientWifi[i+nbStaVideo].Install
(Streamer_Node.Get (i+nbSSVideo+nbStaVideo+nbSSVoix));
        clientAppsWifi[i+nbStaVideo].Start (Seconds (1));
        clientAppsWifi[i+nbStaVideo].Stop (Seconds (duration));
    }
    for (int i = 0; i < nbSSDonnees; i++)
    {
        udpClient[i+ nbSSVideo +nbSSVoix] = UdpTraceClientHelper
(SSinterfaces.GetAddress(i%nbSS), stream_port, "data.dat"); //Objet d'aide à la
        // création et l'installation d'une application de type udpTraceClient(emetteur udp)
        clientApps[i+ nbSSVideo +nbSSVoix] = udpClient[i+ nbSSVideo +nbSSVoix].Install
(Streamer_Node.Get (i+nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix));
        clientApps[i+ nbSSVideo +nbSSVoix].Start (Seconds (1));
        clientApps[i+ nbSSVideo +nbSSVoix].Stop (Seconds (duration));
    }
    for (int i = 0; i < nbStaDonnees; i++)
    {
        udpClientWifi[i+nbStaVideo+nbStaVoix] = UdpTraceClientHelper
(StaWifiinterfaces.GetAddress(i%nbSta), stream_port, "data.dat",UP_BE); //Objet d'aide à la
        // création et l'installation d'une application de type udpTraceClient(emetteur udp)
        clientAppsWifi[i+nbStaVideo+nbStaVoix] =
udpClientWifi[i+nbStaVideo+nbStaVoix].Install (Streamer_Node.Get
(i+nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix+nbSSDonnees));
        clientAppsWifi[i+nbStaVideo+nbStaVoix].Start (Seconds (1));
        clientAppsWifi[i+nbStaVideo+nbStaVoix].Stop (Seconds (duration));
    }
}

NS_LOG_INFO ("Fin de la création des applications");

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

// Création d'un enregistrement de classification dans la couche CS Wimax
IpcsClassifierRecord MulticastClassifier[MAXSS];
ServiceFlow MulticastServiceFlow[MAXSS];
for (int i = 0; i < nbSSVideo; i++)
{
    MulticastClassifier[i] =IpcsClassifierRecord (StreamerCSMAInterfaces.GetAddress(i),
                                                Ipv4Mask ("255.255.255.255"),
                                                SSinterfaces.GetAddress(i%nbSS),
                                                Ipv4Mask ("255.255.255.255"),
                                                0,
                                                65000,
                                                stream_port,
                                                stream_port,
                                                17,
                                                1);

    //Création d'un flux de service dand le lien descendant de type rtPS associé avec
les paquets qui correspondent à
    // l'enregistrement de classification MulticastClassifier
    MulticastServiceFlow[i] = wimax.CreateServiceFlow (ServiceFlow::SF_DIRECTION_DOWN,

```

```

ServiceFlow::SF_TYPE_RTPTS,
                                                                    MulticastClassifier[i]);
    //Ajouter le flux de service créé dans la BS
    ss[i%nbSS]->AddServiceFlow (MulticastServiceFlow[i]);
}
NS_LOG_INFO ("Fin de la création des FS vidéos");

for (int i = 0; i < nbSSVoix; i++)
{
    MulticastClassifier[i+nbSSVideo] =IpcsClassifierRecord
(StreamerCSMAInterfaces.GetAddress(i+nbSSVideo+nbStaVideo),
                                                                    Ipv4Mask ("255.255.255.255"),
                                                                    SSinterfaces.GetAddress(i%nbSS),
                                                                    Ipv4Mask ("255.255.255.255"),
                                                                    0,
                                                                    65000,
                                                                    stream_port,
                                                                    stream_port,
                                                                    17,
                                                                    1);

    //Création d'un flux de service dans le lien descendant de type UGS associé avec les
    paquets qui correspondent à
    // l'enregistrement de classification MulticastClassifier
    MulticastServiceFlow[i+nbSSVideo] = wimax.CreateServiceFlow
(ServiceFlow::SF_DIRECTION_DOWN,

ServiceFlow::SF_TYPE_UGS,
MulticastClassifier[i+nbSSVideo]);
    //Ajouter le flux de service créé dans la BS
    ss[i%nbSS]->AddServiceFlow (MulticastServiceFlow[i+nbSSVideo]);
}
NS_LOG_INFO ("Fin de la création des FS voix");

for (int i = 0; i < nbSSDonnees; i++)
{
    MulticastClassifier[i+nbSSVideo+nbSSVoix] =IpcsClassifierRecord
(StreamerCSMAInterfaces.GetAddress(i+nbSSVideo+nbStaVideo+nbSSVoix+nbStaVoix),
                                                                    Ipv4Mask ("255.255.255.255"),
                                                                    SSinterfaces.GetAddress(i%nbSS),
                                                                    Ipv4Mask ("255.255.255.255"),
                                                                    0,
                                                                    65000,
                                                                    stream_port,
                                                                    stream_port,
                                                                    17,
                                                                    1);

    //Création d'un flux de service dans le lien descendant de type BE associé avec les
    paquets qui correspondent à
    // l'enregistrement de classification MulticastClassifier
    MulticastServiceFlow[i+nbSSVideo+nbSSVoix] = wimax.CreateServiceFlow
(ServiceFlow::SF_DIRECTION_DOWN,
                                                                    ServiceFlow::SF_TYPE_BE,
MulticastClassifier[i+nbSSVideo+nbSSVoix]);
    //Ajouter le flux de service créé dans la BS
    ss[i%nbSS]->AddServiceFlow (MulticastServiceFlow[i+nbSSVideo+nbSSVoix]);
}

NS_LOG_INFO ("Fin de la création des FS Données");

NS_LOG_INFO ("Fin de la création des routes et FSs");

////////////////////////////////////
//                                                                    //
// Lancement de la simulation //
//                                                                    //
////////////////////////////////////

NS_LOG_INFO ("Fin de la configuration");
Simulator::Stop (Seconds (duration + 1.5));
NS_LOG_INFO ("Début de la simulation.....");
Simulator::Run ();

```

```
for (int i = 0; i < MAXSS; i++)
{
    ss[i] = 0;
    SSPosition[i] = 0;
    SSPosAllocator[i] = 0;
}
bs = 0;

Simulator::Destroy ();
NS_LOG_INFO ("Fin.");

return 0;
}
```