TABLE DES MATIÈRES

Page

INTRODUCTION 1

CHAPITRE 1 WiMAX : Présentation générale 3
1.1 Introduction.....3
1.2 Évolution de la norme.....3
1.3 Architecture d'un réseau WiMAX.....4
1.4 Couche physique.....8
1.4.1 Description 8
1.4.2 OFDM et OFDMA..... 10
1.4.3 Time Division Duplexing et Frequence Division Duplexing 11
1.5 MAC 13
1.5.1 Sous couche de convergence 13
1.5.2 Common Part Sublayer 14
1.5.3 Procédure d'entrée dans le réseau..... 15
1.5.3.1 Écoute du canal par la MS et synchronisation avec une BS 17
1.5.3.2 Télémétrie ou ajustement des paramètres 17
1.5.3.3 Négociation des paramètres de base 19
1.5.3.4 Autorisation et échange des clés d'authentification..... 20
1.5.3.5 Enregistrement de la station mobile sur le réseau..... 21
1.5.3.6 Établissement de la connexion IP 21
1.5.3.7 Établissement de la date et de l'heure..... 22
1.5.4 Flux de service 22
1.5.5 Qualité de service..... 24
1.6 Fiabilité de la transmission WiMAX 24
1.6.1 Automatic Repeat-request..... 24
1.6.2 HARQ 25
1.7 Conclusion 26

CHAPITRE 2 NS3 : Description et contexte 27
2.1 Introduction.....27
2.2 Méthodes d'analyses de réseau.....27
2.3 Le simulateur NS3 28
2.3.1 Environnement de développement..... 29
2.3.2 Fonctionnement et concepts clés 30
2.3.2.1 Fonctionnement..... 30
2.3.2.2 Concepts clés 30
2.3.2.3 Nœud..... 30
2.3.2.4 L'application 31
2.3.2.5 Le canal 31
2.3.2.6 Éléments de réseau..... 31
2.3.2.7 Topologyhelpers 31
2.3.3 WiMAX dans NS3 33

2.3.3.1 Modèle de couche physique 33
2.3.3.2 Modèle du canal 34
2.3.3.3 Modèle MAC 35
2.3.3.4 La Convergence Sublayer (CS) 35
2.3.3.5 La Common Part Sublayer(CPS) 37
2.3.3.6 Connexion et adressage 38
2.3.3.7 Entrée dans le réseau 39
2.3.3.8 Le service d'ordonnancement 39
2.3.3.9 Trames et messages de gestion 42
2.3.3.10 Message de gestion 42
2.3.3.11 Flux de service 44
2.3.3.12 OFDMA dans NS3 45
2.3.3.13 Élément de réseaux (Devices) 45
2.3.3.14 Phase de télémétrie 45
2.3.3.15 Message de gestion et ordonnancement 46
2.4 Conclusion 48

CHAPITRE 3 MODÈLE PROPOSÉ 49
3.1 Introduction 49
3.2 Objectives et méthodologies 49
3.2.1 Objectifs 49
3.2.2 Méthodologies 50
3.3 Identifications et classifications des améliorations 50
3.4 Présentation des améliorations apportées 52
3.4.1 Amélioration de la couche physique 52
3.4.2 Détermination de la ressource nécessaire pour une slave 53
3.4.3 Détermination de la durée de la transmission 56
3.4.4 Détermination de la charge utile de symbole 57
3.4.5 Détermination du débit de la transmission 58
3.4.6 La création et le traitement du DL_MAP 59
3.4.7 Détermination des paramètres initiaux de la couche physique 60
3.5 Amélioration de la couche MAC 61
3.6 Détermination de la ressource requise et de la ressource disponible 62
3.7 Détermination de la taille des régions de la DL_MAP 62
3.8 Conclusion 63

CHAPITRE 4 SIMULATION ET ANALYSE DES RÉSULTATS 64
4.1 Introduction 64
4.2 Architecture de test 64
4.2.1 Présentation de l'architecture réseau 65
4.2.2 Méthodes d'extraction des résultats 66
4.2.3 Paramètres et critères de comparaison 71
4.3 Résultats 74
4.3.1 Analyse des résultats 75
4.3.1.1 Délai moyen 75

4.3.1.2 Variation moyenne du délai 77
4.3.1.3 Débit réel de simulation 80
4.4 Conclusion 81

CONCLUSION.....82

BIBLIOGRAPHIE.....83

LISTE DES TABLEAUX

Page

Tableau 3.1 Regroupement des corrections par fichier et par couche 52

Tableau 4.1 Données des statistiques 68

Tableau 4.2 Paramètres de configuration des nœuds et du client UDP 71

Tableau 4.3 Paramètres de configuration du client UDP.....72

Tableau 4.4 Paramètres des règles de classification et durée de la simulation.....73

LISTE DES FIGURES

Page

Figure 1.1 Évolution chronologique de la norme 802.16 4

Figure 1.2 Modèle de référence du réseau 5

Figure 1.3 La couche physique....9

Figure 1.4 Exemple de trame OFDMA TDD....12

Figure 1.5 Procédure d'entrée dans le réseau 16

Figure 1.6 Procédure de création de flux de service 23

Figure 2.1 Composants de base de NS3 29

Figure 2.2 Les modules disponibles dans NS3....32

Figure 2.3 Modèle physique....33

Figure 2.4 Modèle MAC 35

Figure 2.5 Diagramme de classe du classeur de la BS....37

Figure 2.6 Format des trames MAC et des messages de gestions....44

Figure 3.1 Décomposition de la trame OFDMA 53

Figure 3.2 Capture d'écran de la fonction de calcul du débit de la transmission....59

Figure 3.3 Exemple de région de données 62

Figure 4.1 La réalisation d'une simulation dans NS3 64

Figure 4.2 Architecture du réseau 65

Figure 4.3 Architecture du contrôleur de flux....67

Figure 4.4 Capture d'écran montrant la portion de script d'extraction de données. ...74

Figure 4.5 Délai moyen par intervalle entre paquets....77

Figure 4.6 Variation moyenne du délai par intervalle entre paquets....79

Figure 4.7 Débit par intervalle entre paquets 81

Rapport-Gratuit.com

LISTE DES ALGORITHMES

Page

Algorithme 3.1 Algorithme de calcul du nombre de slots 56

Algorithme 3.2 Algorithme de détermination du temps de transmission.....57

Algorithme 3.3 Algorithme de calcul de la charge utile de symbole 58

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AAA	Authentication Authorization and Accounting
ACK	ACKnowledgement
AMC	Adaptative Modulation and Coding
ASN-GW	Access Service Network Gateway
BE	Best Effort
BPSK	Binary Phase Shift Keying
BS	Base Station
CPS	Common Part Sublayer
CSN	Connectivity Service Network
CS	Convergence Sublayer
DIUC	Downlink Interval User code
DL_MAP	Downlink Media Acces Protocol
DSA-ACK	Dynamic Spectrum Access Acknowledgement
DSA-REQ	Dynamic Spectrum Access Request
DSx-RVD	Dynamic Service ReceiVeD
DSx-REQ	Dynamic Service Request
DSA-RSP	Dynamic Spectrum Access Response
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FUSC	Full Usage of SubChannels
GHz	Giga Hertz
Hz	Hertz

IETF	Internet Engineering Task Force
IP	Internet Protocol
LOS	Line Of Sight
LTE	Long Term Evolution
MAC	Media Access Control
MHz	Méga Hertz
MS	Mobile Station
NAP	Network Access Provider
NACK	Negative Acknowledgement
NLOS	Non Line Of Sight
NRM	Reference Network Model
NRTPS	Non-Real-Time Polling Service
NSP	Network Service Provider
NS3	Network Simulator 3
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiplexing Access
OSI	Open Systems Interconnection
PDU	Packet Data Unit
PKM-REQ	Privacy Key Management Request
PKM-RSP	Privacy Key Management Response
PUSC	Partial Usage of SubChannels
Ptx	Puissance de transmission
Prx	Puissance de réception

QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
REG-REQ	Registration Request
REG-RSP	Registration Response
RNG-REQ	Ranging Request
RNG-RSP	Ranging Response
RSSI	Received Signal Strength
RTG	Receive/transmit Transition Gap
RTPS	Real-Time Polling Service
SBC-REQ	Subscriber Station Basic Capability Request
SBC-RSPS	Subscriber Basic Capability Response
SDU-MAC	Service Data Unit MAC
TTG	Transmit/receive Transition Gap
UIUC	Uplink Interval User code
UGS	Unsolicited Grant Service
UL_MAP	Uplink Media Access Protocol
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WiMAX-SC	WiMAX-
WiMAX-SCA	WiMAX-

LISTE DES SYMBOLES ET UNITÉS DE MESURE

Hz	Hertz
MHz	Mégahertz
ms	milliseconde
GHz	Gigahertz
dBm	déciBel par Milliwatt
s	seconde
b/s	bits par seconde

INTRODUCTION

L'augmentation croissante de la demande en données au niveau des utilisateurs de réseaux mobiles sans fil a amené les chercheurs à se renseigner sur des technologies pouvant les satisfaire. C'est dans cette optique que les technologies comme le WiMAX (Worldwide Interoperability for Microwave Access) et le LTE (Long-Term Evolution) ont vu le jour afin de fournir du haut débit, des services mobiles, et une haute efficacité. Dans le cadre de ce mémoire, on se consacre au WiMAX.

Ces technologies sont aussi utilisées par les laboratoires dans le cadre de projets de recherche. Dans ce cas de figure, l'obstacle que rencontrent les chercheurs réside dans le coût exorbitant des équipements et la difficulté de la mise en place en grandeur nature (nécessite de grands espaces).La simulation apparaît donc comme une solution salvatrice pour la recherche. Il permet alors de réaliser des scénarios, des modèles et d'optimiser des conceptions. Il est un puissant outil d'analyse et d'amélioration de technologies. Le choix du simulateur revêt une importance capitale.

La famille des simulateurs peut être subdivisée en deux : les simulateurs vendus et les simulateurs libres. Ces derniers sont souvent appréciés par les chercheurs. Cet intérêt est dû à la possibilité d'accès au code source et de pouvoir les modifier à dessein qu'ils offrent. Cependant, l'inconvénient majeur réside dans la fiabilité et l'organisation de la communauté de développeurs qui pilotent le développement du simulateur: mise à jour régulière, maintenance et correction des bogues. Dans le cadre de notre projet, le partenaire industriel ULTRA ELECTRONICS TCS et nous avons choisi de travailler avec un simulateur libre NS3 qui est géré par une communauté de développeurs active et qui propose régulièrement de nouvelles versions et des mises à jour.

Rapport-Gratuit.com

Il a le mérite de nous offrir un module qui traite le WiMAX en prenant en compte la modulation OFDM. Le module ne supporte pas l'OFDMA. Cependant, un greffon qui propose l'OFDMA a été développé et présenté dans (Furlong et Guha, 2010b). Malheureusement, il n'est pas complet. Le modèle de la couche physique et du canal reste à perfectionner (Furlong et Guha, 2010b). De plus, il ne prend pas en compte les dernières améliorations de NS3. Dans le cadre de ce projet, l'audit du modèle des couches physiques et MAC du greffon à la lumière de la norme est réalisé et les corrections nécessaires sont apportées, de sorte qu'il soit le plus près possible des standards tout en effectuant les abstractions nécessaires à la simulation. Des tests ont été effectués pour comparer le système proposé au système initial. Cela est présenté dans ce rapport en quatre chapitres.

Le premier chapitre présente nos motivations et l'évolution de la norme WiMAX. Le deuxième présente le simulateur WiMAX, ses concepts clés et comment le WiMAX est traité dans NS3. Puis, dans le chapitre trois, nous présentons l'essentiel du modèle proposé : la prise en charge de plusieurs types de modulation, l'amélioration des calculs des paramètres de la couche physique et l'allocation de ressources, le traitement et la création des DL_MAP (Downlink Media Access Protocol), l'envoie des paramètres au niveau du canal, l'envoie et la réception de paquets. Dans le chapitre 4 nous présentons les tests effectués, les résultats et les interprétons afin de vérifier les améliorations du modèle. Enfin, nous présentons, dans la dernière partie du rapport qui est la conclusion générale, un récapitulatif du travail effectué et les possibilités d'amélioration futures.

CHAPITRE 1

WiMax : Présentation générale

1.1 Introduction

La technologie WiMAX a été mise en place en vue de répondre aux besoins accrus des utilisateurs avec des débits pouvant aller jusqu'à 1 Gb/s. Ainsi, elle a subi plusieurs modifications en vue d'améliorer ses performances. Dans ce contexte, nous présentons ses diverses évolutions, son architecture, mais également les concepts clés qui sont utilisés dans le cadre de ce projet.

1.2 Évolution de la norme

Le WiMAX a été conçu avec pour but de définir une technologie de haut débit basée sur le protocole IP en remplacement du réseau filaire. Il possède des fonctionnalités semblables au Wi-Fi tout en couvrant de plus grandes zones et ce, de façon similaire aux réseaux cellulaires. Sa première version, la 802.16-2001, offrait un système point à multipoint avec un débit d'environ 70 Mb/s, une fréquence d'opération située entre 10 et 60 GHz en ligne à vue directe (Line Of Sight, LOS) et une transmission avec porteuse unique. Des modifications ont été apportées plus tard avec les versions de 2003 et 2004 à savoir : la transmission en ligne à vue non directe (No Line Of Sight, NLOS), la définition du nombre de profils, la prédéfinition de paramètres d'interopérabilité 2003 ainsi qu'une nouvelle plage de fréquences d'opérations de 2 à 11 GHz. Des amendements ont été également apportés en 2004, afin d'adopter la modulation à plusieurs porteuses et d'adapter la modulation suivant l'état du canal. Ceux-ci ont ainsi permis d'introduire l'OFDM et l'OFDMA. En 2005, la nouvelle version combine les systèmes fixes et mobiles dans la bande règlementaire. Elle intègre aussi la sécurité, les systèmes multi-porteuses et les systèmes MIMO. La dernière version en date, celle de 2011, promet un débit de 100 Mb/s pour le réseau mobile et 1 Gb/s

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

pour un réseau fixe, la *«subchannelisation»* ainsi que l'efficience dans l'utilisation des structures de pilotes (Bacioccola et al., 2010; Etemad, 2008; Pareit et al., 2012).

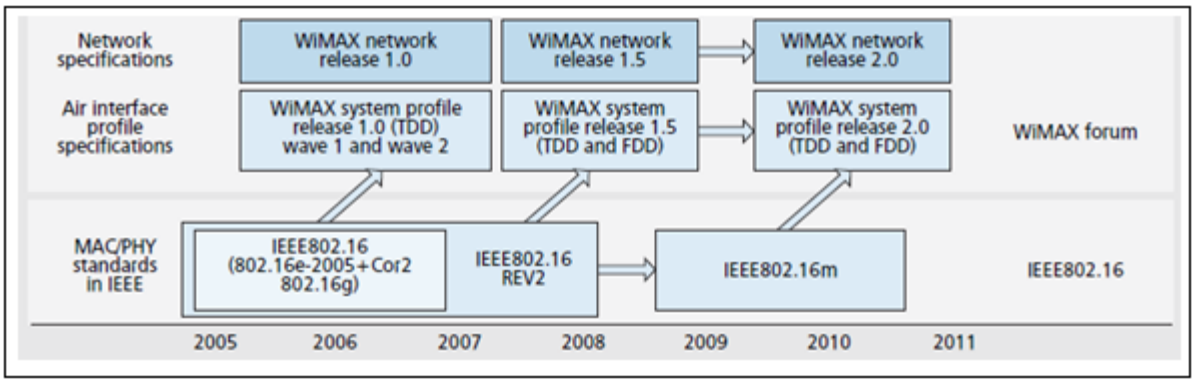


Figure 1.1 Évolution chronologique de la norme 802.16
Tirée de Etemad (2008, p. 33)

1.3 Architecture d'un réseau WiMAX

L'architecture du réseau WiMAX est conçue dans l'optique d'un trafic tout IP. À cet effet, elle doit prendre en compte la pile protocolaire du groupe de travail sur les technologies et protocoles de l'internet (Internet Engineering Task Force, IETF) et respecter les exigences du standard 802.16 ainsi que ceux d'autres standards du domaine IP. Elle doit être également flexible, pour un scénario de déploiement tant au niveau d'une architecture fixe que d'une architecture mobile. Enfin, elle doit assurer le partage de ressources facilitant la décentralisation des infrastructures ainsi que la connectivité, l'accès aux services et l'interopérabilité à travers une connexion via des infrastructures hétérogènes.

Toutes ces spécifications et fonctionnalités sont représentées dans le modèle de référence de WiMAX (Network Reference Model, NRM). Le NRM présenté dans la figure 1.2 fait la distinction entre les fournisseurs d'accès au réseau et les fournisseurs de connectivité IP et de

services WiMAX sur le réseau. Ces deux fournisseurs pourront être, dans certains cas, une même entité.

Les entités constituantes du réseau peuvent être subdivisées en plusieurs grandes catégories à savoir : les équipements d'accès, le réseau d'accès, le réseau de services de connectivité (Connectivity Service Network, CSN) et les points de références R1 à R8 servant d'interface de fonctionnement entre deux entités (Andrews, Ghosh et Muhamed, 2007c; ZeHua et al., 2012).

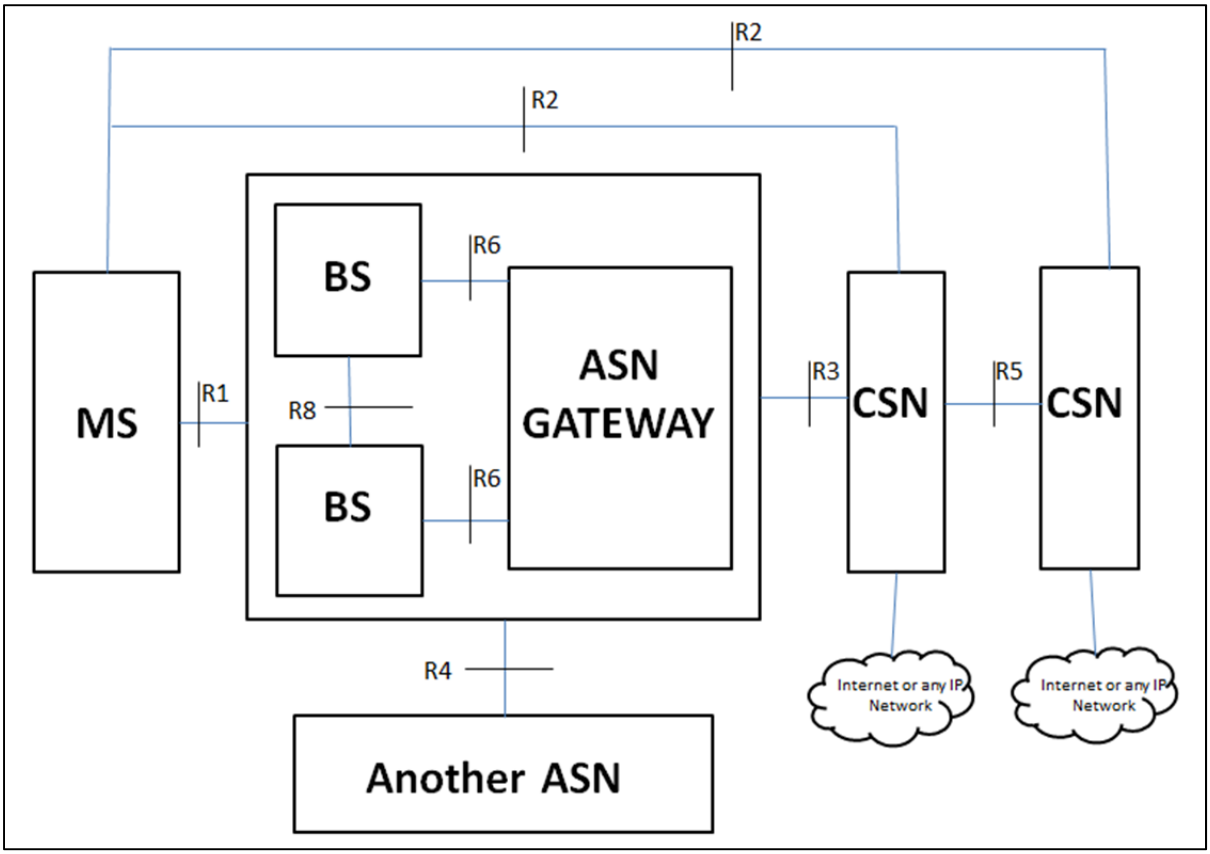


Figure 1.2 Modèle de référence du réseau
Tirée de Jeffrey G. Andrews Rias Muhamed (2007, p. 338)

Comme le montre la figure 1.2 :

- les équipements d'accès : ils représentent les équipements de l'utilisateur (terminaux fixes comme mobiles) qui permettent d'avoir une connexion avec le réseau WiMAX;
- le réseau d'accès : il représente un certain nombre d'équipements qui doivent prendre en charge les différentes fonctions nécessaires pour fournir un accès radio aux équipements d'accès. Il permet entre autres d'assurer la connectivité de niveau 2, le transfert de messages d'authentification d'autorisation et de facturation (Authentification Autorization Accounting, AAA) vers le fournisseur de service du client, la découverte et la sélection du fournisseur, l'allocation d'adresse IP, la gestion de la mobilité et des ressources radio. Le réseau d'accès peut être implémenté en une seule entité offrant toutes ses fonctionnalités ou décomposé en deux entités : la station de base et la passerelle de service d'accès (Access Service Network Gateway, ASN-GW). La version décomposée peut contenir plusieurs stations de base (Andrews, Ghosh et Muhamed, 2007c; ZeHua et al., 2012) :
  - la station de base : elle offre essentiellement des fonctionnalités radio et sert d'interface sans fil entre le réseau et les équipements d'accès. Elle peut prendre en charge optionnellement les fonctionnalités d'ordonnancement de gestion de la micro-mobilité à savoir le déclenchement du transfert et l'établissement de tunnel entre le réseau de service d'accès et le réseau de service de connectivité;
  - la passerelle : elle permet principalement de prendre en charge le trafic de la couche 2 tout en gérant également la qualité de service, la sécurité (clés de chiffrement, fonctionnalité de client AAA), la mobilité, les interactions avec d'autres réseaux d'accès, la gestion des ressources radio, le contrôle d'admission, la mise en cache des profils d'abonnés, la création et la gestion d'un tunnel de mobilité avec des stations de base, ainsi que le routage vers le réseau de service de connectivité.
- le réseau de services de connectivité (Connectivity Service Network, CSN) : il fournit à la fois des services internet et des services applicatifs aux abonnés WiMAX. Il gère la facturation, l'itinérance, la communication avec les réseaux de services de connectivité, tant publics que privés, ainsi que la gestion des adresses IP. Il est composé d'éléments de réseau, tels que les routeurs, mandataires ou serveurs d'authentification d'autorisation et

de facturation, qui prennent en charge l'authentification des appareils, des utilisateurs et des services spécifiques, les bases de données d'informations sur les abonnés, les passerelles d'interconnexion ainsi que des serveurs de réseau améliorés. Il offre aussi des services de diffusion, de multidiffusion et des services basés sur la localisation;

- les points de référence (Andrews, Ghosh et Muhamed, 2007c; ZeHua et al., 2012):
  - R1 : interface entre la station mobile et le réseau d'accès (Access Service Network, ASN). Elle joue le rôle d'interface sans fil;
  - R2 : interface entre la station mobile et le réseau de services de connectivité (Connectivity Service Network, CSN). Elle offre les fonctionnalités d'authentification d'autorisation et de facturation, la configuration d'équipement IP et de gestion de la mobilité;
  - R3 : interface entre le réseau d'accès ASN et le réseau de services de connectivité (Connectivity Service Network, CSN). Elle offre les fonctionnalités d'authentification d'autorisation et de facturation, assure l'application des politiques liées aux profils des utilisateurs et la gestion de la mobilité;
  - R4 : interface entre le réseau d'accès (Access Service Network, ASN). Elle permet de gérer la mobilité;
  - R5 : interface entre les réseaux de services de connectivité (Connectivity Service Network, CSN). Elle permet d'assurer l'interconnexion et l'itinérance;
  - R6 : interface entre la station de base et la passerelle de service d'accès (Access Service Network Gateway, ASN-GW). Elle permet d'assurer la gestion de tunnel IP afin d'établir et de libérer la connexion à la station mobile;
  - R7 : interface interne de la passerelle de service d'accès (Access Service Network, ASN);
  - R8 : interface entre stations de base, elle permet de faire les transferts.

1.4 Couche physique

1.4.1 Description

La couche physique (PHY) est la plus basse dans le modèle OSI (Open Systems Interconnection). Elle utilise l'interface sans fil entre la station de base et celle mobile. Elle transmet également les données par modulation des porteuses simples ou multiples en se basant sur différents schémas de modulation et différentes bandes de fréquence. Elle se fonde sur plusieurs normes, indépendamment du standard de télécommunication choisi. Ainsi, concernant le standard WiMAX, la couche physique a été basée principalement sur les normes IEEE 802.16-2004 à porteuses uniques pour les systèmes fixes et IEEE 802.16-2005 à porteuses multiples pour les systèmes mobiles. Cependant, au cours de son développement, le standard WiMAX définit deux couches physiques supplémentaires pouvant fonctionner avec les couches MAC existantes (Andrews, Ghosh et Muhamed, 2007b; Odhiambo et Rayolla, 2012). Les couches physiques présentes sont :

- *WirelessMAN SC* : couche PHY mono-porteuse destinée à des fréquences au-delà de 11 GHz nécessitant une ligne en vue directe LOS, elle fait partie des spécifications initiales de la 802.16;
- *WirelessMAN SCA* : couche PHY mono-porteuse dédiée aux fréquences entre 2 GHz et 11 GHz pour les systèmes point à multipoint;
- *WirelessMAN OFDM* : couche PHY OFDM avec une FFT à 256 points, elle est utilisée pour les systèmes point à multipoint opérant sans ligne en vue directe, N-LOS, à des fréquences comprises entre 2 GHz et 11 GHz. Elle a été également retenue par la norme WiMAX pour les systèmes fixes. Elle est donc aussi appelée WiMAX fixe;
- *WirelessMAN OFDMA* : couche PHY OFDMA avec une FFT à 2048 points, elle est utilisée pour les mêmes systèmes point à multipoint que la couche WirelessMAN OFDM. Par ailleurs, elle a été modifiée pour convenir à une version adaptative d'OFDMA (Scalable OFDMA, SOFDMA) dans laquelle la taille de la transformer rapide de Fourier (Fast Fourier Transform, FFT) varie entre: 128, 512, 1024 et 2048 selon la norme IEEE 802.16e -2005. La taille variable de la FFT permet une utilisation et une implémentation

optimale du système sur une large bande de canal, le rendant ainsi adaptable aux conditions radio. Cette couche PHY a été acceptée par la norme WiMAX pour les systèmes mobiles, lui conférant aussi le nom de WiMAX mobile.

Quant au fonctionnement de la couche physique, présenté à la figure 1.3, il peut être décomposé en trois étapes (Andrews, Ghosh et Muhamed, 2007b; Odhiambo et Rayolla, 2012) :

- première étape : elle consiste en la correction d'erreur directe (Forward Error Correction, FEC). Elle fait également le codage canal, l'adaptation du débit, l'entrelacement et la cartographie des symboles;
- deuxième étape : les données sont cartographiées dans les sous-canaux et les sous-porteuses appropriés afin de construire les symboles OFDM. Pour pouvoir estimer et connaitre l'état du canal, les symboles pilotes sont insérés dans les sous-porteuses correspondantes. On y réalise aussi le codage espace/temps afin d'augmenter la diversité;
- troisième étape : les symboles OFDM sont convertis du domaine fréquentiel vers le domaine temporel puis en un signal analogique qui est transmis sur l'interface sans fil.

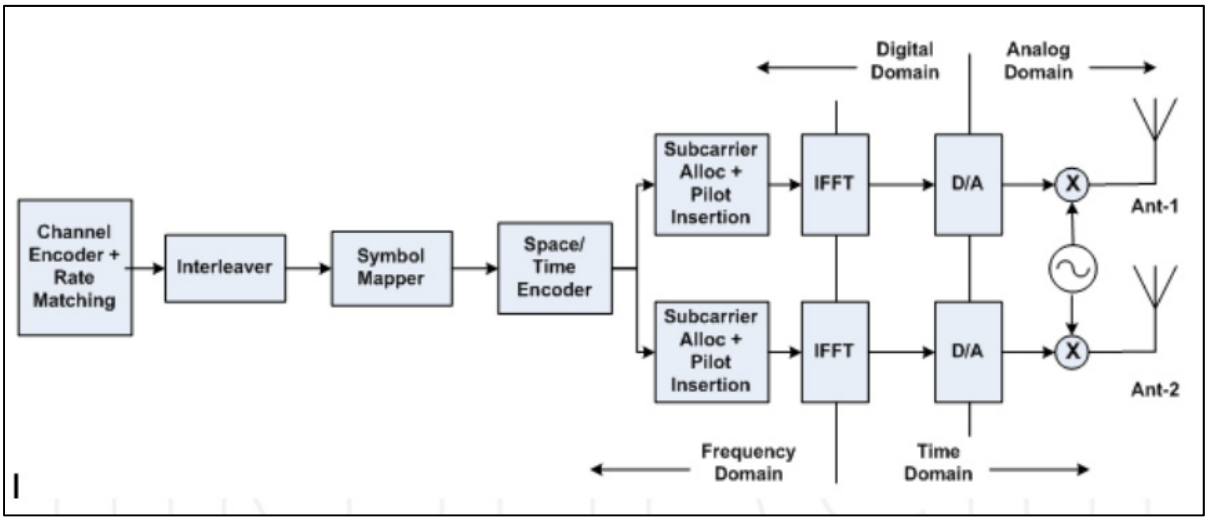


Figure 1.3 La couche physique
Tirée de Odhiambo et Rayolla (2012, p 197)

1.4.2 OFDM et OFDMA

L'OFDM est une modulation multi-porteuses avec un espacement fixe dans le domaine des fréquences. Les ressources disponibles y sont représentées soit à l'aide de symboles dans le domaine temporel, soit à l'aide des sous-porteuses dans le domaine fréquentiel (Etemad, 2008), contrairement à la modulation mono-porteuse qui consiste à allouer toute la bande (1,25 à 20 MHz) à une seule porteuse. En divisant la bande en plusieurs sous-porteuses (2048), on accroît le débit. Les sous-porteuses sont donc regroupées en sous-canaux (Odhiambo et Rayolla, 2012). Cependant, le choix du nombre de sous canaux dépend de la bande passante par canal qui doit être supérieure à la bande de cohérence. Cela rend le système moins exposé aux problèmes de canaux sélectifs en fréquence. Par ailleurs, les sous-porteuses doivent être orthogonales afin de diminuer l'interférence inter-symbole (Inter-Symbol Interference, ISI). Cette interférence est totalement éliminée grâce au préfixe cyclique (Cyclic Prefix, CP) à condition que sa durée soit supérieure à l'étalement de la réponse impulsionnelle due au multi-trajet. Quant à la durée du symbole, elle est choisie de manière à être supérieure au retard dû au multi-trajet afin d'éviter les effets du multi-trajet.

Contrairement à la bande de l'OFDM, celle de l'OFDMA peut être associée à plusieurs sources de données tant dans le domaine temporel que fréquentiel. Le nombre de sous-porteuses alloué à un utilisateur est fonction soit de ses besoins, soit des conditions du canal, soit de ces deux exigences. Cela nécessite une modulation adaptative à travers l'augmentation ou la réduction de la cadence d'envoi de données. L'OFDMA requiert également la maximisation de la diversité multi-utilisateur qui est le gain obtenu, en choisissant un utilisateur ou un groupe d'utilisateurs donné. En OFDMA, il y a trois types de sous-porteuses toutes regroupées en sous-canaux :

- les porteuses de données utilisées pour la transmission de données;
- les sous-porteuses pilotes pour l'estimation et la synchronisation;
- les sous-porteuses nulles pour la bande de gardes et la sous-porteuses DC (pour identifier la fréquence centrale). Elles ne portent aucune information et ne sont pas transmises.

L'unité de base pour l'allocation de ressources dans l'OFDMA est le *slot*. Une combinaison et une séquence de *slot* constituent un sous-canal. Afin de former les sous-canaux, deux types de permutations sont utilisées : la permutation aléatoire et la permutation contigüe. La première consiste à choisir les sous-porteuses de manière aléatoire afin de former les sous-canaux. Cela offre une meilleure diversité fréquentielle et une bonne résistance aux erreurs. Quant à la permutation contigüe, elle consiste à regrouper les sous-porteuses adjacentes formant ainsi un sous-canal. En général, la permutation aléatoire est mieux adaptée pour les applications mobiles. En revanche, la permutation contigüe est meilleure pour les applications fixes à mobilité faible (Etemad, 2008).

1.4.3 Duplexage temporel et fréquentiel

Les méthodes de duplexage fréquentiel (Frequency Division Duplex, FDD) et temporel (Time Division Duplex, TDD) sont prises en charge par la norme 802.16 :

- le duplexage fréquentiel (Frequency Division Duplex, FDD) nécessite deux bandes de fréquence RF distinctes, l'une pour la transmission montante et l'autre pour la transmission descendante des données. Ceci offre la possibilité d'émettre et recevoir de manière simultanée. Il est largement utilisé dans la téléphonie cellulaire (Etemad, 2008);
- le duplexage temporel (Time Division Duplex, TDD) utilise un seul canal divisé dans le temps pour l'émission et la réception. La trame TDD est subdivisée en deux sous-trames, l'une pour la transmission montante et l'autre pour la transmission descendante, séparée chacune par un intervalle de transition réception/ émission (Receive/Transmit Transition Gap, RTG) ou un intervalle de transition émission/réception (Transmit/Receive Transition Gap, TTG) comme le montre la figure 1.4. Cela permet d'éviter les collisions. De plus, des informations de contrôle sont insérées dans la trame afin d'assurer une utilisation optimale du système. Comme le montre la figure 1.4, nous avons différents champs dans la trame à savoir :
    - *Preamble* : il sert à la synchronisation. Il contient plusieurs indices, l'une pour son identification et les autres pour les identifiants du segment et celui de la cellule;

RapportGratuit.com

- *Frame Control Header* : il fournit entre autres les informations sur la taille du message MAP, le codage utilisé et les sous-canaux utilisables. Il est suivi de l' UL_MAP et du DL_MAP qui fournissent l'information sur l'allocation des sous-canaux et des informations de contrôle, respectivement pour les liaisons montantes et descendantes.
- *UL ranging* : c'est un sous-canal qui est alloué à la station mobile afin d'effectuer des ajustements de fréquence, de puissance et de négociation de bande passante;
- *UL CQICH* : c'est un sous-canal alloué à la station mobile afin de retourner des informations sur l'état du canal;
- *UL ACK* : désigne un sous-canal est alloué à la station mobile afin d'accuser réception de la requête hybride automatique de répétitions de paquets de la liaison descendante (Down Link Hybrid Automatic Repeat request, DL HARQ).

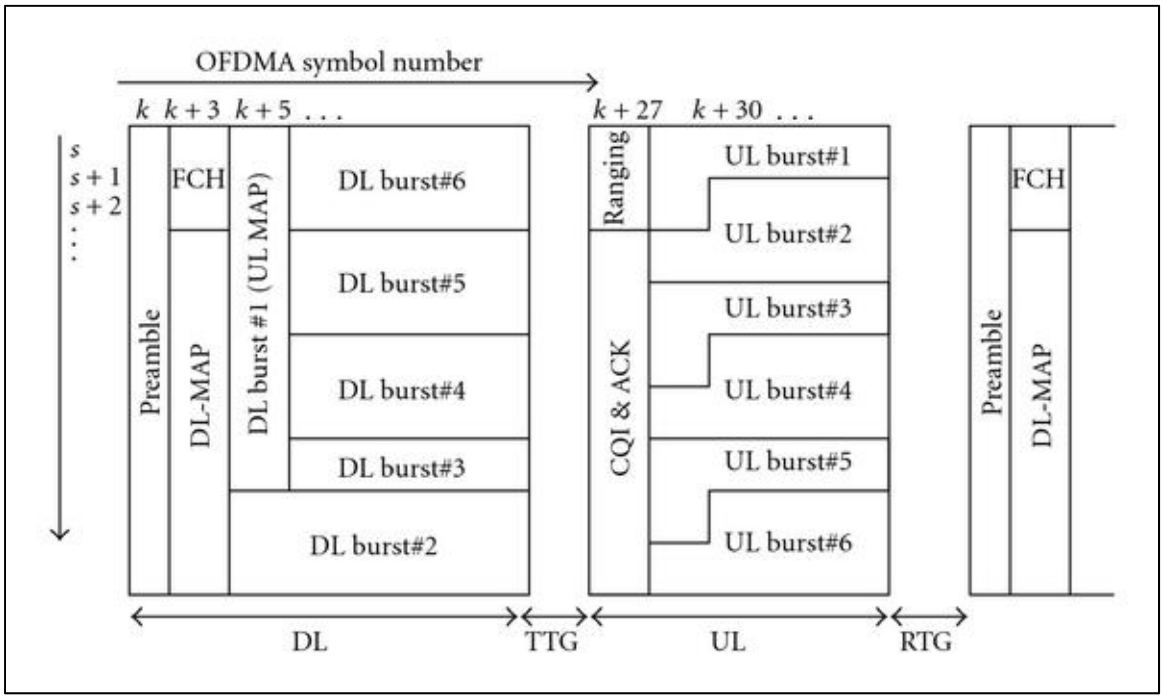


Figure 1.4 Exemple de trame OFDMA TDD
Tirée de So-In, Jain et Tamimi (2010, p 2)

Contrairement au duplexage fréquentiel où la bande passante allouée est fixe et indépendante du trafic, le duplexage temporel permet d'ajuster les proportions de bande passante allouée aux transmissions (montantes ou descendantes) dépendamment du trafic disponible. Il offre ainsi une plus grande flexibilité d'adaptation aux diverses allocations du spectre globale. La conception d'émetteur/récepteur supportant le duplexage temporel est également plus simple et moins onéreuse.

1.5 La couche MAC

La couche MAC est une partie de la couche liaison de données du mode OSI. Formée de deux sous-couches : la sous-couche de convergence (Convergence Sublayer, CS) et la sous-couche commune (Common Part Sublayer, CPS) (Etemad, 2008), elle offre une interface entre la couche physique et les couches supérieures. Elle permet de gérer de manière efficiente les ressources physiques telles que la bande passante et la puissance de la transmission. Elle transforme les SDU de la couche MAC (MAC Services Data Unit, MSDU) des couches hautes en PDU de la couche MAC (MAC Protocol Data Unit, MPDU) afin de les diriger vers la couche physique. Elle est également conçue pour prendre en charge divers protocoles de la couche supérieure. Cependant, elle ne prend en charge actuellement que l'Ethernet et l'IP, ce qui est suffisant pour un système tout IP.

1.5.1 Sous-couche de convergence (Convergence Sublayer, CS)

Elle permet de supprimer et de reconstituer les en-têtes afin d'améliorer la charge utile. Elle réalise aussi la correspondance entre les adresses de couche supérieure, telles que les adresses IP, et les identifiants de connexions (Connection ID, CID) utilisés entre la couche MAC et la couche physique. Ces identifiants sont utilisés pour combler le manque de visibilité entre la couche supérieure et les couches basses (MAC et PHY). De plus, la couche MAC étant orientée connexion, l'identifiant de connexion identifie une connexion unidirectionnelle entre la station de base et la station mobile ou un groupe de connexion réservé pour la diffusion ou

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

l'initiation de connexion au réseau. Cependant, l'identifiant de la liaison montante est différent de celui de la liaison descendante. Par ailleurs, il est dynamique et temporaire. Il peut être également assimilé à une adresse de la couche 2 assignée par la station de base, afin d'identifier la connexion entre couches MAC et PHY. Il sert au transport des données et des informations de contrôle. Une trace de la correspondance entre les adresses destination et les identifiants de connexion doit être conservée afin de transporter le SDU (service Data Unit, SDU) vers la connexion MAC adéquate. Cependant, les SDU destinés à une même adresse peuvent être transmis en utilisant différents identifiants de connexion. Le choix de l'identifiant approprié est fait suivant la qualité de service. Il dépend également de l'identifiant du flux de services (Service Flow ID, SFID) et de l'adresse source du SDU. La sous-couche de convergence peut s'interfacer avec une variété de protocoles de couche supérieure comme le mode de transfert asynchrone (Asynchronous Transfer Mode, ATM), Ethernet, IP, et tout protocole d'avenir inconnu. Il y a deux types de sous couche de convergence (Convergence Sublayer, CS) :

- les sous-couches de convergence ATM : responsables des services en mode de transfert asynchrone,
- les sous-couches de convergence qui supportent les réseaux de paquet comme IP, Ethernet et PPP.

1.5.2 La sous-couche commune (Common Part Sublayer, CPS)

La sous-couche commune est indépendante des couches supérieures. Elle est responsable du contrôle de la liaison, du maintien de la connexion, de la gestion de la demande de la bande passante, de la planification des services, du choix de la modulation et du taux de codage ainsi que de la construction des trames. Elle gère également la qualité de service (Quality Of Service, QoS).

À cet effet, les SDU reçus de la sous-couche de convergence sont assemblés pour former les PDU (Packet Data Unit,PDU), unités de base des charges utiles niveau MAC et PHY. Une

fois le PDU constitué, l'ordonnanceur vérifie son identifiant de flux de service et celui de connexion avant son envoi sur la ressource physique disponible. Cela permet de savoir la qualité de service requise et d'assurer ainsi une allocation optimale des ressources physiques. Plusieurs PDU destinés à un même récepteur peuvent être concaténés et envoyés sur une région de données représentant un ensemble de slot réservé pour les transmissions montantes et descendantes. Dépendamment de la taille de la charge utile, plusieurs SDU peuvent être soit concaténés dans un seul PDU, soit fragmentés pour être envoyés via plusieurs PDU. Dans le cas de la fragmentation du SDU, la position de chaque PDU est référencée par un numéro de séquence. Celui-ci permet à la fois d'éviter les pertes de données en cas d'erreur de transmission et de garder la synchronisation, facilitant ainsi la reconstitution du salve à la réception. Chaque PDU est composé de trois éléments :

- L'entête MAC (Generic MAC header, GMH) qui contient l'identifiant de connexion (Connection ID, CID), la longueur de la trame et le bit pour signifier si le contrôle de redondance cyclique (Cyclic Redundancy Check, CRC) a été effectué;
- la charge utile : qui est un message de transport ou de gestion identifié par le sous en-tête qui le suit immédiatement. Elle peut également contenir des requêtes de bandes passantes ou de retransmission;
- puis du contrôle de redondance cyclique (Cyclic Redundancy Check, CRC), si le message est crypté.

1.5.3 Procédure d'entrée dans le réseau

Cette procédure permet à la station mobile d'établir une communication bidirectionnelle avec la station de base. Elle se déroule en plusieurs étapes. Les étapes sont présentées à la figure 1.5 et décrites dans les sous les sous-sections suivantes.

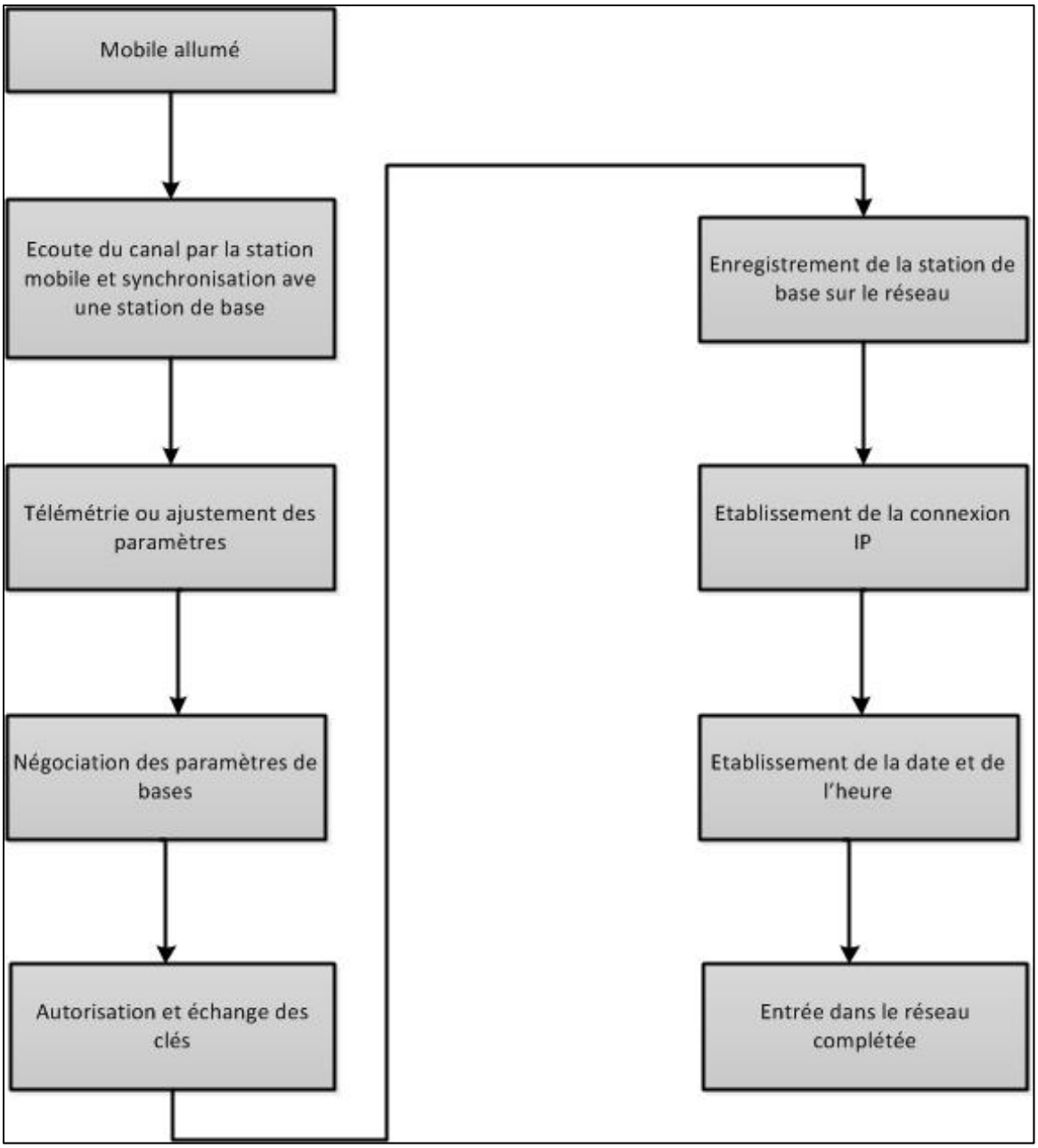


Figure 1.5 Procédure d'entrée dans le réseau
Adaptée de Jeffrey G. Andrews Rias Muhamed (2007, p. 320)

1.5.3.1 Écoute du canal par la station mobile et synchronisation avec une station de base

Lors de l'initiation de la connexion, la station mobile doit acquérir un canal descendant. Pour ce faire, dans un premier temps, il utilise les fréquences de transmissions descendantes privilégiées se trouvant dans une liste enregistrée en mémoire avec les derniers paramètres de connexion. Elles peuvent être modifiées afin de s'adapter au fournisseur de services du réseau. En cas d'échec de l'initiation avec ces précédents paramètres, la station mobile balaie les canaux disponibles de la bande de fréquences de liaison descendante jusqu'à ce qu'elle trouve un signal valide. Cela se fait par l'intermédiaire de la station mobile qui écoute les préambules de trames de transmissions descendantes. Ils sont également utilisés par la station mobile pour la synchronisation avec la station de base. Par la suite, la station mobile écoute les différents messages de contrôle, à savoir le message de contrôle d'entête de trame (Frame Control Header, FCH), le descripteur du canal descendant (Down link Channel Descriptor, DCD), le descripteur du canal montant (Up link Channel Descriptor : UCD), et les messages DL_MAP et UL_MAP, qui suivent le préambule dans la trame, afin d'obtenir les différents paramètres physiques et MAC relatifs aux transmissions montantes et descendantes. La synchronisation est maintenue tant que la station mobile reçoit des messages DL-MAP et le descripteur du canal descendant descendant (Down link Channel Descriptor, DCD). Les messages UCD et UL-MAP sont transmis périodiquement par la station de base (Andrews, Ghosh et Muhamed, 2007a).

1.5.3.2 Télémétrie ou ajustement des paramètres

L'objectif principal de cette phase est d'obtenir les ajustements d'espacement temporel et les paramètres de puissance nécessaires pour maintenir la connexion montante avec la station de base lors de la phase d'initialisation. Une fois la connexion établie, la station mobile doit réaliser un suivi périodique de la fluctuation de ces paramètres qui peut être due :

- à la mobilité : elle se traduit par la diminution de la puissance du signal due au déplacement,

- à l'évanouissement rapide : il est dû à l'addition constructive et destructive des ondes cela se traduit par une variation de l'amplitude du signal,
- aux effets d'ombrage : ils sont dus aux réflexions multiples des ondes reçues, sur des obstacles situés dans l'environnement,

ou à toute combinaison de ceux-ci. La station mobile calcule ensuite le niveau de puissance de transmission pour la phase initiale de la télémétrie en utilisant des paramètres (maximum Equivalent Isotropic Received Power : $EIRxP_{IR,\ MAX}$, BSEIRP : Base Station Equivalent Isotropic Radiated Power, Received Signal Strength : RSSI) extraits des messages de description des canaux DCD et UCD (Andrews, Ghosh et Muhamed, 2007a). Le niveau de puissance P_{TX} se calcule suivant l'équation (1.1). Elle revient à la somme de la puissance maximum reçue à la station de base et de la perte liée au chemin entre la station de base et celle mobile

$$P_{TX} = EIRxP_{IR,MAX} + BSEIRP - RSSI \tag{1.1}$$

Tirée de Jeffrey G. Andrews Rias Muhamed (2007, p. 320)

où :

- $EIRxP_{IR,\ MAX}$ est la puissance maximum d'une réception avec une antenne unique ;
- RSSI est la puissance reçue par la station mobile ;
- BSERIP la puissance de transmission de la station de base.

Une fois la puissance calculée la station mobile envoie ainsi un message *«Ranging Request»* (RNG- REQ) avec le *«Code Division Multiple Access»* (CDMA) pour identifier la phase de télémétrie ainsi que le niveau de puissance avec l'identifiant de connexion(CID) alloué à cet effet. Si la station mobile ne reçoit pas de réponse de la station de base après un certain délai

ou si le code contenu dans la réponse est *«continue»*, la station mobile estime que la précédente tentative a échoué et entre dans la phase de résolution du problème. Dans cette phase, elle transmet un message RNG- REQ avec un nouveau code CDMA et un niveau de puissance supérieur à la puissance précédente. Elle répète cette étape jusqu'à la réception d'un message *«Ranging Response»* (RNG –RSP) contenant le code de réponse, le niveau de puissance, les paramètres temporels et fréquentiels ainsi que les CID de base et les CID primaires alloués à la station mobile. Enfin, la station mobile effectue l'ajustement des paramètres comme indiqué dans le message reçu (Andrews, Ghosh et Muhamed, 2007a).

1.5.3.3 Négociation des paramètres de base

Au cours de cette phase, la station mobile et la station de base s'échangent les paramètres supportés. Le message *«SS Basic Capability Request»* (SBC-REQ) est transmis de la station mobile à la station de base afin d'informer cette dernière de ses paramètres physiques ainsi que ceux relatifs à l'allocation de la bande passante (Andrews, Ghosh et Muhamed, 2007a) :

- paramètres physiques :
  - l'intervalle de transmission : l'écart de transmission entre les trames de la liaison montantes et les trames de la liaison descendante pris en charge par la station mobile pour le duplexage temporel et le duplexage fréquentiel;
  - la puissance d'émission maximale : la puissance de transmission maximale pour chaque type de modulation (BPSK, QPSK, 16-QAM, et 64-QAM;
  - la puissance d'émission actuelle : correspondant à la puissance utilisée pour l'émission du message SBC-REQ ;
  - la taille de la FFT : les différentes tailles prises en charge (128, 512, 1024, et 2048 pour le mode et 256 pour le mode OFDM) ;
  - la prise en charge de 64 QAM afin de notifier si le 64 QAM est pris en charge par le modulateur et le démodulateur ;
  - la prise en charge FEC : permettant de notifier l'option FEC supportée (CTC, LDPC, etc.) ;

- la prise en charge de HARQ : pour vérifier la prise en charge des requêtes hybrides automatiques de répétitions de paquets;
- la prise en charge de STC (Space Time Code) et MIMO (Multiple In Multiple Out) : afin de notifier les différents modes de codages espace / temps et de codage MIMO ;
- prise en charge de l'*Advanced Antenna System* (AAS) MAP : pour notifier la prise en charge des différents AAS Media Access Protocol ;
- prise en charge du contrôle de la puissance de liaison montante : pour notifier les options de contrôle et de puissance de la liaison montante (boucle ouverte, boucle fermée, AAS préambule, contrôle de puissance) ;
- prise en charge de la permutation des sous canaux : pour notifier la prise en charge de diverses permutations des sous canaux : PUSC, FUSC, AMC et les modes TUSC.

• paramètres d'allocation de la bande passante :
  - prise en charge du duplexage fréquentiel Half-Duplex/Full-Duplex.

Après cette étape, la station de base répond avec le message SBC-RSP (Subscriber Basic Capability ReSPonse) fournissant les paramètres qui seront utilisés pour les futures transmissions. Ceux-ci peuvent être identiques aux paramètres requis par la station mobile ou un sous-ensemble.

1.5.3.4 Autorisation et échange des clés d'authentification

Après la négociation des paramètres de base, la station mobile procède à l'échange sécurisé des clés à travers le protocole *«Privacy Key Management»* (PKM). Elle envoie ainsi un message *«Privacy Key Management Request»* (PKM-REQ) à la station de base. Cette dernière répond avec un message *«Privacy Key Management Response»* (PKM-RSP). Ce processus est optionnel et négocié lors de la phase de négociation des paramètres de base (Andrews, Ghosh et Muhamed, 2007a).

1.5.3.5 Enregistrement de la station mobile sur le réseau

L'enregistrement est le processus par lequel la station mobile est autorisée à entrer dans le réseau et recevoir les identifiants de connexion secondaires. À cet effet, la station mobile envoie tout d'abord un message de requête d'enregistrement (Registration Request, REG-REQ) à la station de base. Ce message contient un code d'authentification de message haché (Hash-based message authentication code, HMAC) que la station de base utilisera pour valider l'authenticité du message. Une fois la validation effectuée, la station de base envoie un message de réponse à la requête d'enregistrement (Registration Response, REG-RSP) qui contient l'identifiant de connexion secondaire de gestion. À l'issue de cette phase, l'enregistrement de la station mobile sur le réseau est effectué (Andrews, Ghosh et Muhamed, 2007a).

1.5.3.6 Établissement de la connexion IP

Lors de la phase d'enregistrement, le message de requête d'établissement de connexion de la station mobile (REG-REQ) contient aussi les paramètres secondaires non pris en compte lors de la phase de négociation tels que la prise en charge de la version d'IP (IPv4, IPv6), les paramètres la couche de convergence et les paramètres de requêtes automatiques de répétitions de paquets. Ceux-ci permettent à la station mobile d'indiquer la version d'IP supportée. Une fois le message reçu, la station de base indiquera à son tour, via le message *«Registration Response»* (REG-RSP), la version d'IP qu'elle utilisera. Par ailleurs, dans le cas où aucun échange de version n'a été effectué, elle suppose que la version supportée par la station mobile est l'IPv4. Une fois la version d'IP connue, la station mobile utilise les mécanismes DHCP pour obtenir une adresse IP auprès du serveur DHCP ainsi que d'autres paramètres nécessaires à l'établissement de la connectivité IP (Andrews, Ghosh et Muhamed, 2007a).

Rapport-Gratuit.com

1.5.3.7 Établissement de la date et de l'heure

Afin d'avoir la date et l'heure de la station de base à la seconde près, la station mobile récupère l'heure d'un serveur de temps via la station de base. Le serveur supporte le mode connecté (Transmission Control Protocol, TCP) et le mode non connecté (User Datagram Protocol, UDP). Il écoute les connexions ou attend les datagrammes sur le port 37. Lorsque la connexion est établie, ou à l'arrivée d'un datagramme, le serveur renvoie la valeur de la date et de l'heure sur 4 octets puis ferme la connexion dans le cas du mode connecté. Si le serveur n'est pas en mesure de déterminer la valeur de la date et de l'heure, il refuse toute connexion ou ferme la connexion sans aucune réponse, en mode connecté. En mode non connecté, il rejette les datagrammes arrivant et ne répond pas à ceux reçus. (IETF, 1983).

1.5.4 Flux de service

Une fois l'enregistrement terminé, une connexion pour le flux de service est créée. Celui-ci est utilisé afin de réaliser les communications entre les couches supérieures du réseau. La connexion peut être initiée par la station mobile ou par la station de base selon que le trafic initial arrive par la liaison montante ou par la liaison descendante. Comme le monte la figure 1.6.

Ainsi, si elle est initiée par la station mobile, cette dernière envoie un message *«Dynamic Service Addition Request»* (DSA-REQ) contenant la qualité de service nécessaire pour la transmission de ce service. Dès réception de ce message, la station de base vérifie son intégrité et renvoie un message DSx- RVD dans le cas échéant. Ensuite, elle vérifie si elle peut fournir la qualité de service demandée, crée un nouveau *«Service Flow Id»* (SFID) et envoie un message *«Dynamic Service Addition Response»* DSA-RSP avec la qualité de service fournie. La station mobile met fin aux procédures en envoyant l'accusé de réception DSA-ACK.

Dans le cas où l'initiation est faite par la station de base, elle vérifie si la station mobile est autorisée à accéder à ce service et si elle peut supporter la qualité de service demandée. Si elle est autorisée, la station de base crée un nouveau *«Service Flow Id»* (SFID) et envoie le message DSA-REQ avec la qualité de service appropriée et l'identifiant de connexion sera utilisé. La station mobile notifie son accord en envoyant le message *«Dynamic Service Addition Response»* DSA-RSP. Une fois la notification effectuée, la station de base renvoie l'acquittement de ce message via le message *«Dynamic Service Addition ACKnoledge»* DSA-ACK (Andrews, Ghosh et Muhamed, 2007a).

Une fois le flux de service créé, la station de base et celle du mobile disposent d'une paire de connexions pouvant servir à échanger des données et des messages de gestion.

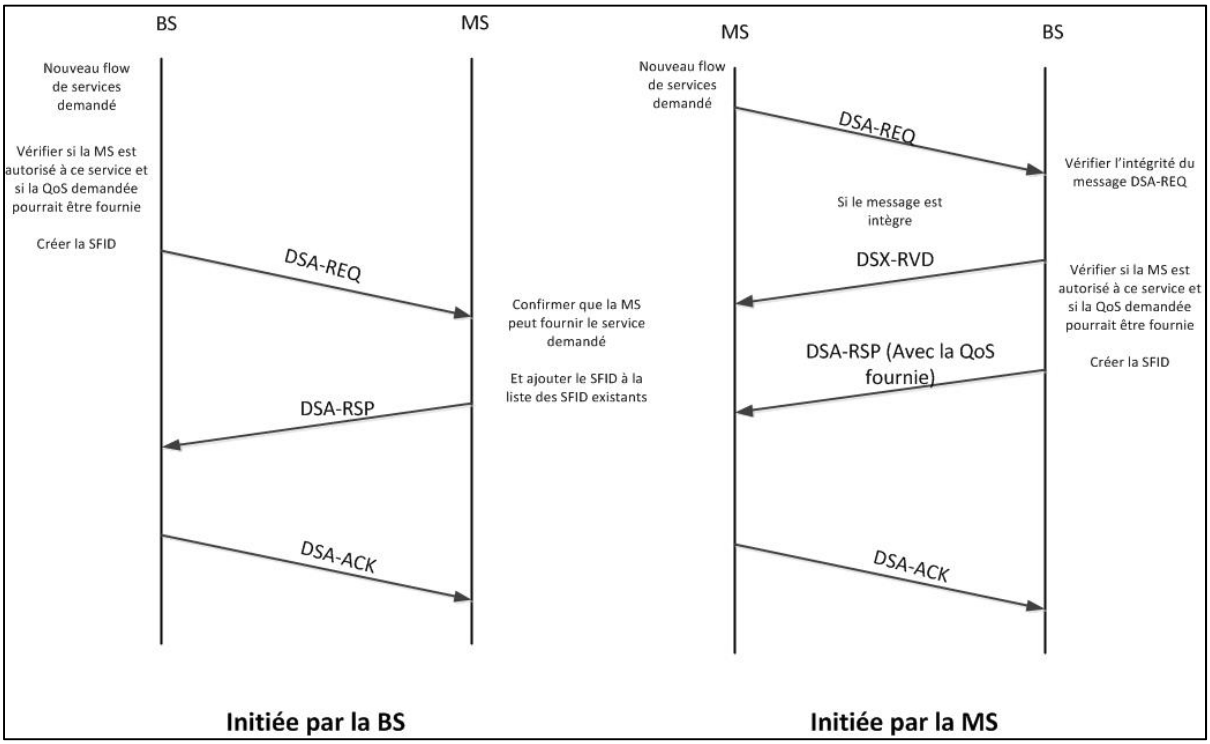


Figure 1.6 Procédure de création de flux de service
Adaptée de Jeffrey G. Andrews Rias Muhamed (2007)

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

1.5.5 Qualité de service

Dans la couche MAC, la qualité de service est fournie par les flux de service. Ces derniers sont fonction des classes de service. On distingue 5 classes de services (Jeffrey G. Andrews Rias Muhamed, 2007):

- *«Unsollicited Grant Service»* (UGS)
- *«Real Time Polling Service»* (RTPS)
- *«Extended Real Time Polling Service»* (ERTPS)
- *«Non-Real Time Polling Service»* (NRTPS)
- *«Best Effort»* (BE)

1.6 Fiabilité de la transmission WiMAX

Dans des transmissions sans fil, les conditions de réception peuvent varier à cause de plusieurs facteurs tels que l'évanouissement dû à la mobilité et l'interférence. Aussi, une façon traditionnelle d'obtenir la fiabilité de la transmission est la répétition de la transmission (Cipriano et al., 2010). Dans le WiMAX, deux mécanismes sont utilisés afin d'assurer la fiabilité, les requêtes automatiques de répétitions de paquets (Automatic Repeat-request, ARQ) et les requêtes hybrides automatiques de répétitions de paquets (Hybrid Automatic Repeat-request,HARQ), tous basés sur le contrôle de parité pour vérifier l'intégrité de données et la retransmission si celles-ci sont perdues ou corrompues.

1.6.1 Requêtes automatiques de répétitions de paquets (Automatic Repeat-request, ARQ)

Il y a trois types de procédures de requêtes automatiques de répétitions de paquets qui sont le plus implémentés *: stop-and-wait, go-back-N*, et la retransmission sélective. L'ARQ est utilisé pour assurer la transmission de données fiables en retransmettant les données corrompues. Les *«Service Data Unit»* (SDU-MAC) sont subdivisés en bloc ARQ dont la

taille est négociée lors de la connexion. Ces blocs ne peuvent pas être fragmentés. Une fois au niveau du destinataire, le paquet reçu est vérifié grâce au code *«Cyclic Redundancy Check»* (CRC) qui est contenu dans chaque bloc. Le destinataire envoie un accusé de réception (Acknowlegement, ACK) en cas de succès de la vérification. Dans le cas contraire, il n'accuse pas réception (Negative Acknowlegement, NACK). L'émetteur, quant à lui, retransmet un bloc seulement s'il reçoit un NACK ou si le délai de retransmission est écoulé (Dias et al., 2010).

1.6.2 Requête hybride automatique de répétitions de paquets (Hybrid Automatic Repeat-request, HARQ)

La requête hybride automatique de répétitions de paquets (Hybrid Automatic Repeat-request, HARQ) combine le code correcteur d'erreur FEC et l'ARQ. Il est implémenté seulement pour la couche physique OFDMA. Cela permet de réduire le nombre de retransmissions par rapport à la requête automatique de répétitions. Cependant, cette combinaison accroît la complexité de son implémentation et augmente le débit du système par rapport à la requête automatique de répétitions. Il existe deux types de requête hybride automatique de répétitions de paquets (Dias et al., 2010):

- le premier est appelé de *«type 1 chase combining»*. Avant la transmission, chaque bloc de données et le code CRC associé sont codés utilisant un codeur FEC. La retransmission est demandée si le décodeur ne peut pas décoder correctement le bloc reçu. Ainsi, quand un bloc codé retransmis est reçu, il est combiné avec le bloc codé précédemment détecté et utilisé à l'entrée du décodeur FEC. La combinaison des deux versions du bloc de code permet d'améliorer les possibilités de réussite du décodage ;
- le deuxième, de type 2, est conçu pour améliorer la fiabilité de retransmission. Il est appelé redondance incrémentale. Contrairement au type 1, chaque retransmission est codée différemment pour obtenir de meilleures performances. Le taux de codage diminue à chaque retransmission.

1.7 Conclusion

Les différentes versions du WiMAX développées ont permis la convergence vers un réseau haut débit, plus fiable et robuste. Ces principales caractéristiques ont conduit à une augmentation de l'utilisation des réseaux WiMAX. Il est donc important de disposer de techniques permettant d'analyser et de simuler ce réseau.

CHAPITRE 2

NS3 : Description et contexte

2.1 Introduction

Dans ce chapitre, nous présenterons brièvement les diverses méthodes d'analyses de réseau puis nous mettrons plus l'accent sur celles choisies. Nous décrivons ensuite l'outil qui sera utilisé pour cette méthode et la représentation de WiMAX dans ce dernier. Cette description portera aussi sur les limites du système choisi par notre partenaire industriel ULTRA ELECTRONICS TCS.

2.2 Méthodes d'analyses de réseau

Il y a plusieurs méthodes d'analyse de réseau à savoir :

- la méthode analytique : elle consiste en une modélisation de système et en des analyses mathématiques ;
- la simulation : elle consiste en la modélisation du système à un niveau d'abstraction à travers des logiciels ;
- l'émulation des équipements qui agissent comme un système réel ;
- les mesures actives ou passives de paramètres prédéfinis ;
- l'expérimentation à travers un banc de tests.

La simulation constitue sans nul doute l'une des méthodes prédominantes d'évaluation dans le domaine des réseaux (Weingartner, vom Lehn et Wehrle, 2009). Elle est moins onéreuse et nécessite un temps moindre comparé à l'implémentation réelle. Elle permet également d'étudier des concepts clés en simplifiant les systèmes plus complexes (Henderson et al., 2006).

Dans la famille des simulateurs nous avons ceux à évènements discrets. Ils sont des outils puissants pour la recherche sur les protocoles, leurs interactions et la mesure de performance (Henderson et al., 2006). Un simulateur est dit à évènements discrets quand ces simulations consistent en une série d'évènements (envois de paquets, ajout de nœuds, etc.) indépendants qui modifient l'état de la simulation. Chaque évènement planifié s'exécute jusqu'à sa fin sans avoir d'effet sur la durée de la simulation. Cependant, à la fin de chaque évènement, la durée s'incrémente de la date planifiée du début du prochain évènement (Thomas, 2011).

Les simulateurs de la famille NS (*Network Simulator*) sont des simulateurs à évènements discrets. Leur développement a commencé avec le simulateur NS-11 en 1995 par *Berkeley National Laboratory* suivi de la première version de la suite NS-2 en 1996 (Henderson et al., 2006). Le dernier NS3 a commencé en 2006 (NSNAM, 2013). Nous présentons dans la prochaine section le simulateur NS3.

2.3 Le simulateur NS3

NS3 est formé d'un noyau testé et validé et de différents modules. Le noyau est composé de l'ordonnanceur, des classes définissant les objets clés à savoir : les paquets, les nœuds, les canaux, les applications, les éléments de réseaux. Tous ces éléments sont utilisés dans des scripts afin de réaliser les simulations. La figure 2.1 qui montre les composants de base de NS3 de NS3. Les langages supportés par NS3 sont C++ et Python, contrairement au NS-2 qui utilise du TCL. Il est conçu en mettant une emphase sur l'ensemble des couches 2 à 4 du modèle OSI, lui permettant ainsi de supporter les adresses IP (Henderson et al., 2006). Il dispose d'une architecture modulable, extensible et réaliste, avec de meilleures performances (NSNAM, 2011). Cette particularité lui permet de supporter la simulation de réseau de grande échelle de manière efficiente avec de meilleure performance en comparaison aux autres simulateurs existants (Weingartner, vom Lehn et Wehrle, 2009). Il génère en sortie des fichiers de type capture de paquet (Packet Capture, *PCAP,* des traces et des fichiers journaux.

Pour la visualisation et l'analyse des résultats. Le NS3 offre aussi la possibilité d'y intégrer d'autres logiciels libres, on peut citer comme exemple le logiciel de capture et d'analyse de paquets Wireshark. Il est fait par et pour la communauté de recherche (NSNAM, 2013). Cette communauté est formée d'environ 963 utilisateurs et de 1176 développeurs (NSNAM, 2011).

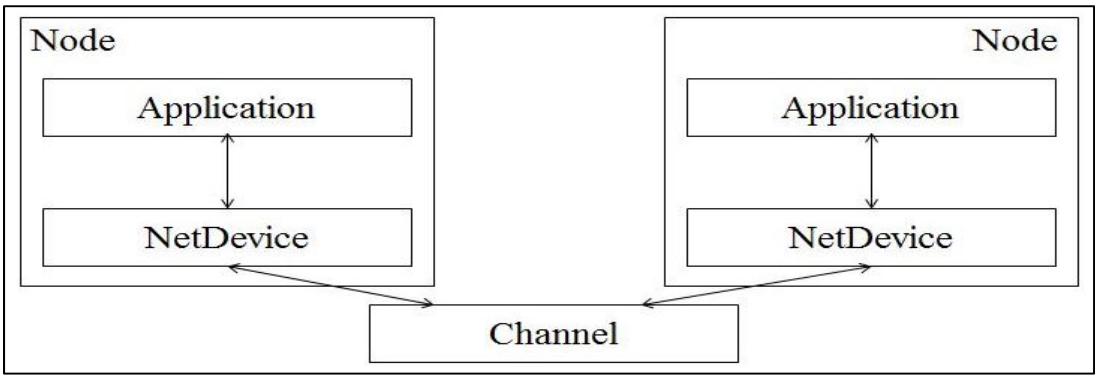


Figure 2.1 Composants de base de NS3

2.3.1 Environnement de développement

La gestion du code source dans NS3 regroupe la communication entre les membres de la communauté, la gestion des divers changements apportés au code source et la gestion de la documentation. Elle est réalisée grâce au gestionnaire de code source *Mercurial*. Un environnement de développement peut être utilisé pour l'implémentation. Celui-ci permet de compiler directement le programme écrit, de réaliser des exécutions pas à pas en cas d'erreur, et de conserver la synchronisation avec *Mercurial* afin de partager le travail avec les autres membres de la communauté ou tout autre groupe. Avec un environnement de développement comme Éclipse, des mots clés comme *TODO* et *FIXME* sont utilisés afin de notifier dans le code les parties à développer et à corriger. Le marqueur *FIXME* sert à notifier les taches dont la non-réalisation entrave le fonctionnement du système et *TODO* les taches dont la réalisation représente une amélioration du système. Cela a pour but de rendre plus aisée la communication entre les membres de la communauté qui travaillent sur le même projet.

2.3.2 Fonctionnement et concepts clés

2.3.2.1 Fonctionnement

NS3 peut être installé sur la plupart des systèmes d'exploitation comme Linux, Mac OS, Windows suivant trois étapes : le téléchargement, la compilation et la validation. Son fonctionnement est assez simple. En effet, il consiste en la réalisation d'un script contenant la topologie définie à travers : les nœuds, le canal et les interfaces réseau, la configuration des piles de protocoles et des applications ainsi que les attributs choisis. Ainsi, une fois le script mis en place, son exécution est effectuée. On pourra alors procéder à l'analyse des résultats.

2.3.2.2 Concepts clés

Afin de mener à bien les simulations et d'avoir à la fois un système robuste et moins complexe, plusieurs simplifications du système à étudier ont été faites. Celles-ci sont représentées par des classes C++ fournissant les méthodes à utiliser.

2.3.2.3 Nœud

Un nœud est utilisé pour représenter un ordinateur ou un équipement terminal utilisateur. Il représente une unité de base. Tout comme l'ordinateur, on peut y ajouter des applications, des périphériques, des pilotes comme le montre la figure 2.1. De plus on peut également y implémenter des piles protocolaires des exemples de protocoles supportés sont présentés dans la figure 2.2. Il est géré par la classe *Node* qui représente ses diverses fonctionnalités (NSNAM, 2013).

2.3.2.4 L'application

L'application est la partie logicielle du nœud, elle utilise les ressources physiques disponibles au niveau des nœuds afin de réaliser les tâches qui lui sont demandées. Dans NS3, l'application est utilisée afin de générer des flux de données au niveau des nœuds. Elle est représentée par la classe du même nom servant de base à la création de nouvelles applications (NSNAM, 2013).

2.3.2.5 Le canal

Le canal comme présenté à la figure 2.1, représente le média utilisé pour la communication entre deux nœuds. Il permet de gérer les éléments de communication du sous-réseau et la connexion des nœuds à ces derniers. Il peut aussi être spécialisé par le développeur afin de définir d'autres médias tels que les canaux filaires, sans fil, etc. (NSNAM, 2013).

2.3.2.6 Éléments de réseau

Pour la connexion de deux nœuds, nous avons besoin de périphériques réseau, composés de matériel et de logiciel, et/ou de câbles. Dans NS3, l'élément de réseau représente à la fois les deux éléments du périphérique réseau : logiciel et le matériel. Il est installé dans le nœud, comme le montre la figure 2.1 facilitant ainsi la communication avec d'autres nœuds via le canal. Un nœud peut être connecté à plusieurs autres nœuds via plusieurs éléments de réseau. Il est représenté par la classe *NetDevice* (NSNAM, 2013).

2.3.2.7 Classe d'aide à la création de topologie (Topologyhelpers)

Afin de constituer un réseau, on aura besoin de nœuds et de canaux. Comme vu précédemment, chaque nœud devra contenir un élément de réseau. Cet ensemble devra être

Rapport-Gratuit.com

connecté au canal et devra être configuré adéquatement. Dans le cadre d'un réseau avec des nombres de nœuds élevés, cette tâche peut devenir fastidieuse (NSNAM, 2013). NS3 offre des classes facilitant la création de topologie qui rendent la tâche plus simple grâce à quelques lignes de code dans le script de la simulation. Cette fonctionnalité s'étend aussi à la création d'autres éléments du système. Chaque module de la figure 2.2 en possède une classe d'aide à la création de ces divers éléments (topologie, nœuds canaux, etc.).

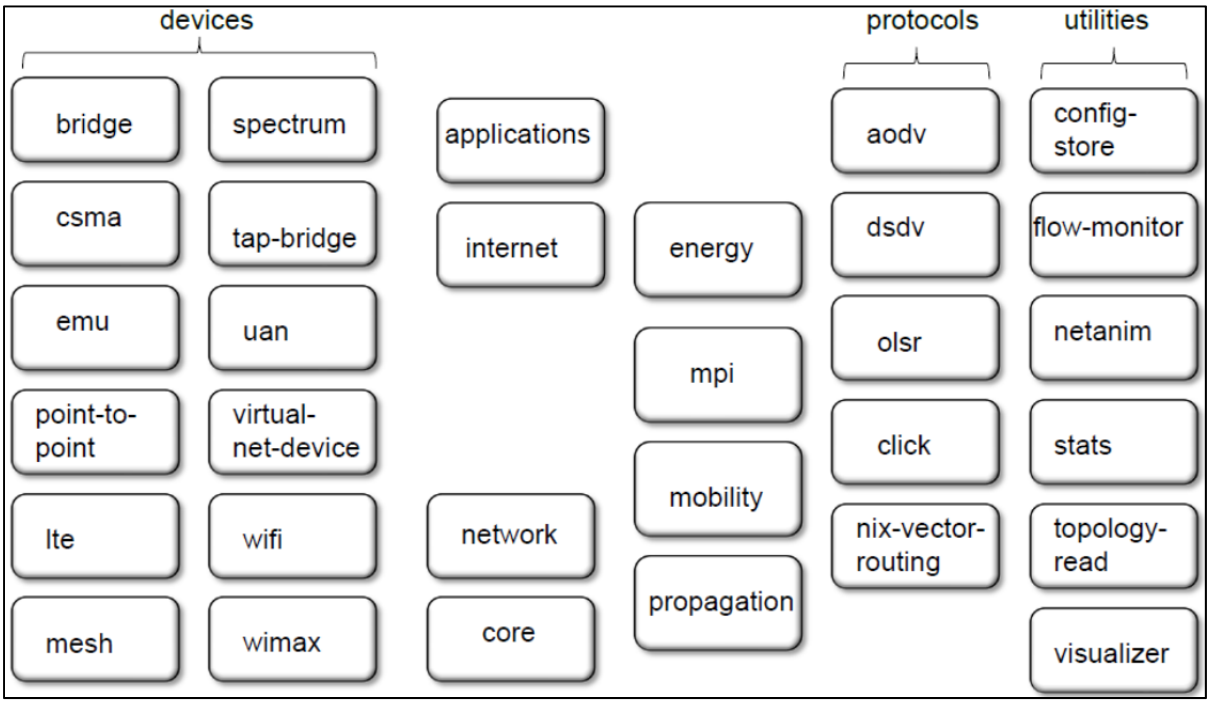


Figure 2.2 Les modules disponibles dans NS3
Tirée de NSNAM (2011, p 8)

NS3 est formé de plusieurs modules comme on peut le voir sur la figure 2.2. Dans le cas de nos travaux, nous nous attarderons plus sur le module WiMAX.

2.3.3 WiMAX dans NS3

2.3.3.1 Modèle de couche physique

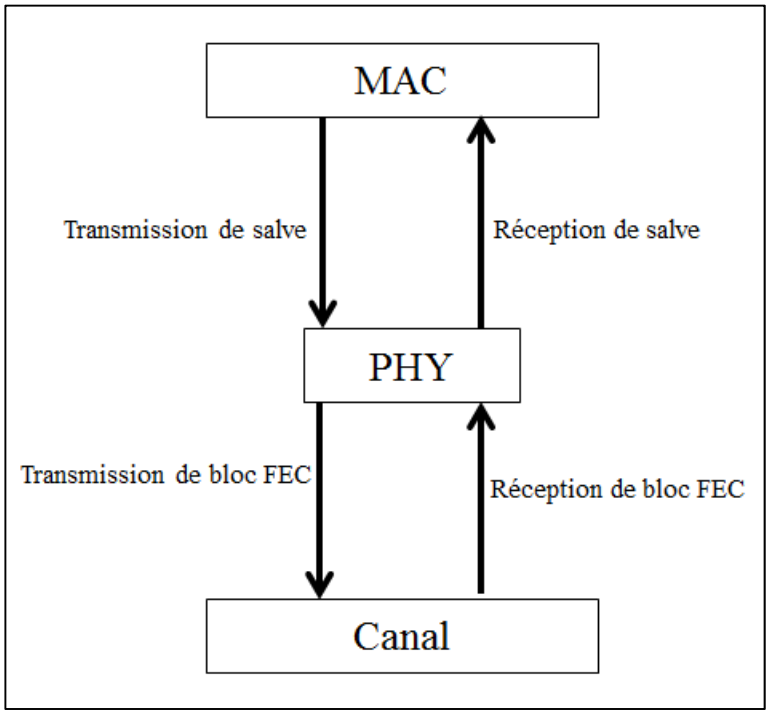


Figure 2.3 Modèle physique

La couche physique gère principalement l'échange de trafic entre la couche MAC et le canal comme décrit dans la figure 2.3. Cette gestion est implémentée dans le modèle de la manière suivante :

- transmission : la salve qui est formée par un ensemble de PDU est convertie en un flux de bits subdivisé en petits blocs FEC qui sont à leur tour envoyés au canal avec la même puissance de transmission;
- réception : une fois les blocs FEC reçus du canal, un calcul du niveau de bruit, avec l'équation (2.1). Le niveau de bruit est fonction de la figure de mérite et de la bande passante. Une fois le niveau de bruit connu on réalise un estimé du SNR, en utilisant l'équation (2.2).

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

$$\mathrm{N} = -114 + N_F + 10log_{10}(\mathrm{BW}) \quad (2.1)$$

Tirée de (Ismail et al., 2010)

où N_F est la figure de mérite du récepteur et BW la bande de passante

$$SNR_{RX} = P_{tx} - L - N \quad (2.2)$$

Tirée de (Ismail et al., 2010)

où L est la perte due au chemin.

Cela permet d'évaluer si le bloc FEC sera correctement décodé. Après cette étape vient le regroupement des blocs FEC pour la reconstruction de la salve originale afin de l'acheminer vers les couches supérieures (Ismail et al., 2010).

2.3.3.2 Modèle du canal

Le modèle est implémenté dans une classe qui hérite de la classe *WimaxChannel*. Cette classe est celle qui implémente le canal WiMAX. Il possède une structure *m_phylist* qui contient la liste de tous les équipements connectés à un canal. À chaque bloc FEC reçu, il calcule la distance entre deux entités physiques grâce à la méthode *GetDistanceFrom(),* la perte liée au chemin à travers les modèles de propagation pour un mode de propagation donné, ainsi que le délai de propagation. Une fois tous ces calculs effectués, le canal transmet le FEC au destinataire(Ismail et al., 2010).

2.3.3.3 Modèle MAC

La couche MAC est subdivisée en deux sous couches : la sous-couche de convergence (Convergence Sublayer, CS) et la sous-couche commune (Common Part Sublayer, CPS) (Etemad, 2008). Les prochaines sections décriront les deux couches présentées à la figure2.4.

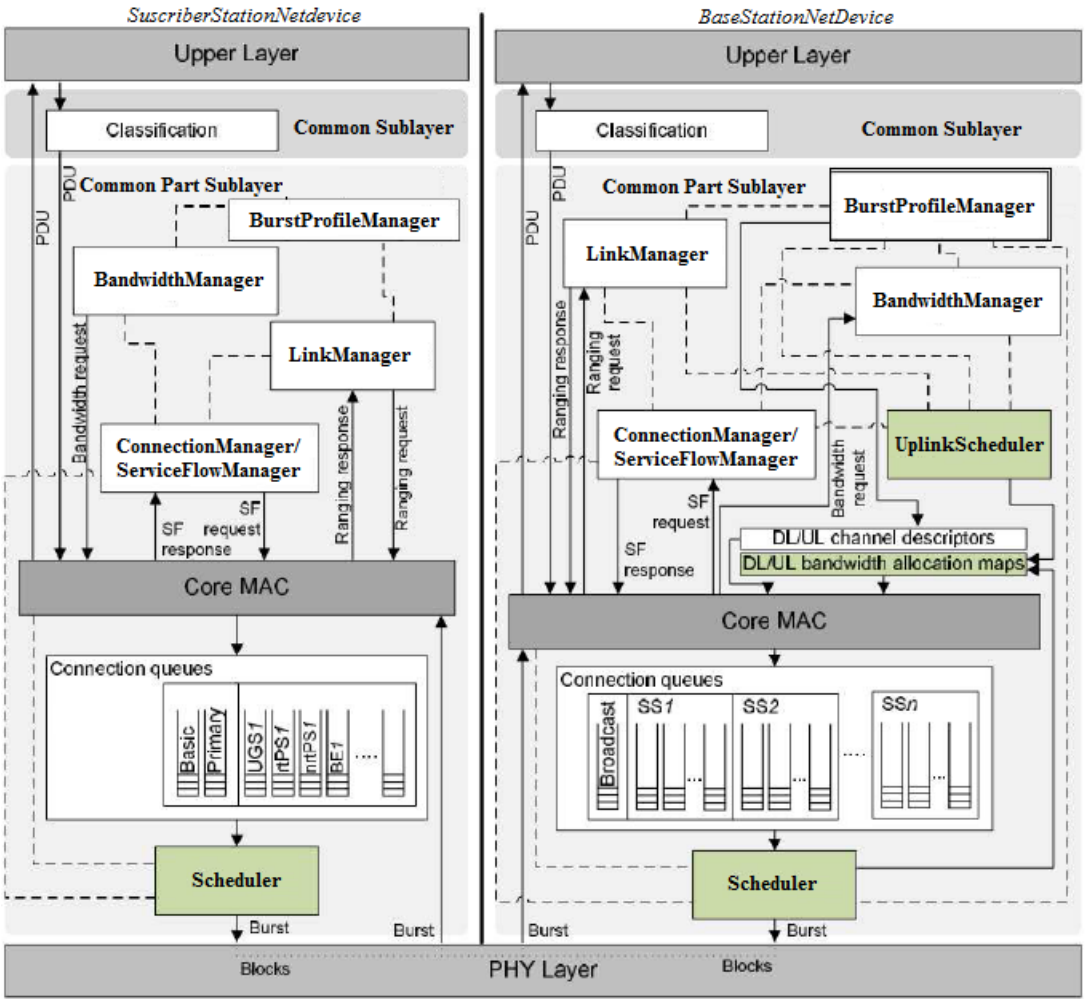


Figure 2.4 Modèle MAC
Adaptée de Farooq et Turletti (2009, p 4)

2.3.3.4 La sous-couche de convergence (Convergence Sublayer, CS)

Deux types de sous-couche de convergence sont définis par la norme, IEEE 802.16, l'une pour le traitement des services en mode de transfert asynchrone (Asynchronous Transfer

Mode, ATM) et l'autre pour le traitement des paquets. C'est le traitement de paquets qui est implémenté dans le module. Comme le monte la figure 2.4 la sous-couche de convergence (Convergence Sublayer, CS) reçoit les paquets des couches supérieures ou d'autres nœuds et les classe dans la connexion de services appropriée (Farooq et Turletti, 2009). Chaque connexion de services est identifiée pas un identifiant de lux de service (Service Flow ID, SFID). Afin de réaliser le classement, la sous-couche de convergence utilise un classeur IP. Le module en implémente deux, un pour la station mobile qui classe les flux montants et le deuxième pour la station de base qui classe les flux descendants (Ismail et al., 2010) :

- classeur de la station mobile (Subscriber Station, SS) : il associe les paquets IP à la connexion appropriée. À cet effet, il contient une liste de règles d'association. Chacun des éléments de cette liste représente une règle d'association entre un flux IP et un identifiant de flux de service précis. À la réception des paquets de la couche IP, le classeur de la station mobile en extrait les adresses IP source et destination, les ports source et destination ainsi que le protocole. Avec ces informations, il cherche s'il y a une règle qui s'applique au paquet traité. Dans le cas échéant, l'identifiant de flux de service correspondant est extrait de la liste et le paquet est alors envoyé dans la file de ce flux de service. Sinon, il est envoyé dans le flux de service par défaut dans le script de la simulation (Ismail, Piro et al. 2010);
- classeur de la station de base (Base Station, BS): il implémente les mêmes fonctionnalités que celui de la station mobile. La figure 2.5 montre son diagramme de classe. Il conserve une liste des stations mobiles connectées aux stations de base. Chaque élément de la liste contient un ensemble de règles de traitement qui lui est associé. À la réception d'un paquet des couches supérieures, le classeur extrait l'adresse IP destination et vérifie si elle correspond à celle d'une des stations mobiles connectées. Il extrait ainsi l'adresse IP source, les ports source et destination ainsi que le protocole. Le paquet est alors traité de la même manière que le classeur station mobile le fait. Dans le cas contraire, le paquet est relâché (Ismail, Piro et al. 2010).

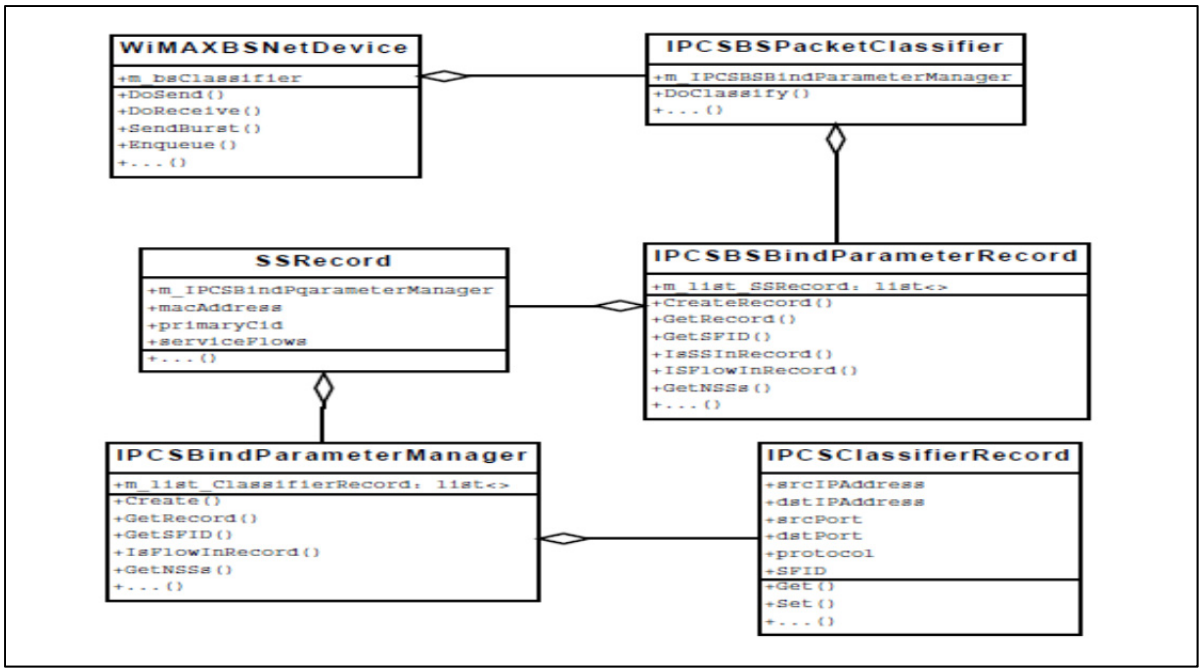


Figure 2.5 Diagramme de classe du classeur de la BS
Tirée de Ismail, Piro et al (2010, p 7)

2.3.3.5 La sous-couche commune (Common Part Sublayer, CPS)

Elle est la sous-couche principale de la sous-couche MAC. Elle en effectue les fonctions fondamentales. Le module implémente le mode PMP Point à Multi Point. Dans ce mode, la station de base gère la communication avec plusieurs stations mobiles. La sous-couche commune implémente principalement les fonctionnalités de gestion de trame, d'adressage, de génération de messages de gestion MAC, d'initialisation et d'enregistrement de station mobile, de gestion de flux de service, de gestion de bande passante et de service d'ordonnancement. Elle est représentée par les classes *BaseStationNetDevice* et *SubscriberStationNetDevice* qui représentent respectivement les couches MAC de la station de base et de la station mobile comme le montre la figure 2.5. Les deux héritent de la classe *WimaxNetDevice* (Ismail, Piro et al. 2010).

D'autres classes pour station mobile et station de base sont aussi utilisées afin de pouvoir implémenter toutes les fonctionnalités de la CPS telles que :

- le gestionnaire de lien implémenté par la classe *Link Manager,*
- le gestionnaire de connexions implémenté par la classe *ConnectionManager,*
- le gestionnaire de flux de services implémenté par la classe *ServiceFlowManager*,
- le gestionnaire de profiles de paquets implémenté par la classe *BurstProfileManager,*
- le gestionnaire de bande passante implémenté par la classe *BandwidthManager,*
- l'ordonnancement des paquets implémenté par les classes *UplinkScheduler* et *Scheduler*.

Les interactions entre ses différentes classes sont présentées dans la figure 2.4.

2.3.3.6 Connexion et adressage

La couche MAC de WiMAX est orientée connexion. Une connexion désigne une association unidirectionnelle entre la station de base et la station mobile. Elle est identifiée par un identifiant de connexion appelé CID. Le standard définit deux types de connexion :

- la connexion de gestion : elle gère les messages de contrôle de transmission et doit être bidirectionnelle (une paire de connexions montante et descendante avec le même CID). Les deux principales connexions de gestion sont nommées de base (*Basic CID*) et primaire (*Primary CID*) ;
- la connexion de transmission : elle sert à transmettre les données.

Chaque connexion gère sa propre file de transmission. La file contient les données reçues à travers la connexion comme le montre la figure 2.4.

Toutes ces fonctionnalités définies par la norme IEE 802.16 sont prises en charge par le simulateur. La classe *WimaxConnection* implémente les connexions de base et la classe CID les identifiants de connexion. La structure *ConnectionManager* est responsable de la création et de la gestion des connexions. On en a une à chaque nœud. Ainsi, celle de la station de base s'occupe des connexions des stations mobiles qui lui sont associées et celle de la station mobile s'occupe de toutes les connexions qui sont associées à la station mobile. Afin de

réaliser la bidirectionnalité de la connexion de gestion, après la création de celle de la liaison descendante la station de base, son CID est envoyé à la station mobile. Cette dernière crée à son tour la connexion de gestion de la liaison montante avec le même identifiant (Ismail, Piro et al. 2010).

2.3.3.7 Entrée dans le réseau

La phase d'entrée dans le réseau au niveau du simulateur se scinde en deux étapes (Ismail, Piro et al. 2010) :

- balayage et synchronisation : dans cette partie, la station mobile balaie la liaison descendante d'un canal précis à la recherche de trame de la couche physique. Une fois celle-ci détectée, la synchronisation débute. Elle est établie dès la réception d'un message DL_MAP et reste maintenue tant que les messages de demandes d'allocation de bande passante et de description de la liaison descendante (DL_MAP et DCD) sont reçus. Par ailleurs, la réception du message de description de la liaison descendante (UCD) permet d'avoir les paramètres du canal montant offert par la station de base et marque ainsi la fin de cette partie ;
- télémétrie initiale : après l'étape précédente, la station mobile reste en attente de l'UL_MAP qui contient les demandes en bandes et d'autres informations sur la liaison montante. En effet, c'est à partir de ce message qu'elle pourra recueillir l'intervalle de télémétrie. Cet intervalle est fixé à 0,5 ms dans le simulateur, mais peut y être modifié dans le script. Par ailleurs, l'intervalle de télémétrie est alloué de façon régulière par l'ordonnanceur de la station de base.

2.3.3.8 Le service d'ordonnancement

L'ordonnancement dans WiMAX consiste principalement en la gestion de la bande passante et de la qualité de service. Il est implémenté dans le module WiMAX de NS3 par (Ismail, Piro et al. 2010) :

- l'ordonnanceur de liaison montante : implémenté par la classe *UplinkScheduler*, il permet de déterminer la liaison montante qui sera attribuée à une station mobile. Pour ce faire, il se base sur des paramètres de qualité de service du flot de service ou du service d'ordonnancement ainsi que sur les demandes en bande passante de la station mobile ;
- l'ordonnanceur de trafics sortants : formé par les classes *BSScheduler* pour celui de la station de base et *SSScheduler* pour celui de la SS. Il permet de déterminer quel paquet sera transmis. La *BSScheduler* détermine quel paquet sera transmis dans la trame de la liaison montante et le *SSScheduler* détermine quel paquet sera transmis dans la trame de la liaison descendante. Ils sont tous basés sur le principe de premier arrivé premier servi. L'allocation des bandes passantes se fait par priorité la priorité la plus élevée re vient au paquet de types UGS les paquets de types RTPS suivis des paquets de types NRTPS et en dernier lieu les paquets de types BE);
- le gestionnaire de bande passante (*bandwidthManager*).

Avec ces différentes classes, le modèle WiMAX implémente tous les quatre services d'ordonnancement définis par la norme IEEE 802.16. Chaque service d'ordonnancement a des paramètres de qualité de service et de bandes passantes précises. Ces paramètres sont spécifiés dans les tableaux 111a à 110d de la 802.16. C'est en se basant sur la qualité de service offert par un service d'ordonnancement qu'on lui attribue un flux de service. Un flux de service est associé à un seul service d'ordonnancement et à une seule connexion de transport. À sa création, l'ordonnanceur de la liaison montante calcule les paramètres nécessaires (*grantinterval* et *grantsize*) à son assignation à un service d'ordonnancement. Le modèle propose trois différentes versions d'ordonnanceur (Ismail, Piro et al. 2010):

- l'ordonnanceur simple : il est implémenté par les classes *BSSchedulerSimple* et *UplinkSchedulerSimple*. Il est basé sur le principe premier arrivé premier servi. Il prend en charge les services temps réels (UGS et RTPS) et ceux qui ne le sont pas (NRTPS et BE). Aux services temps réels, il alloue la bande passante garantie à un intervalle de temps précis. Pour les autres, il vérifie si la bande passante demandée est disponible après qu'on ait satisfait les requêtes des services en temps réel. Si oui, il alloue la bande

passante. Si non il alloue une bande passante minimum. De plus, il ne prend en charge que des paquets de taille fixe;

- l'ordonnanceur RTPS (*Real Time Polling Service,* RTPS) : il est implémenté dans la classe *UplinkSchedulerRTPS* et *BSSchedulerRTPS*. Il est similaire à l'ordonnanceur simple exception faite des connexions RTPS. Ces dernières ont la plus grande priorité. Elles transmettent tous leurs paquets tant qu'il y a de la bande passante disponible. Afin d'éviter la saturation, un système de contrôle a été implémenté. Ce dernier redistribue la bande passante disponible aux connexions RTPS qui ont au moins un paquet à transmettre. Le reste est alors loué aux connexions de types RTPS et BE;
- *Migration* : cet ordonnanceur est basé sur la migration suivant la qualité de service. L'ordonnancement se fait en respectant de manière stricte les priorités. Il réalise l'ordonnancement en se basant sur trois files d'attente à priorité basse, intermédiaire et élevée :
  - file à priorité basse : elle contient les requêtes de bandes passantes des connexions BE;
  - file à priorité intermédiaire : elle contient les requêtes de bandes passantes des connexions de types RTPS et RNTPS. Ces connexions pourront être transférées de cette file à celle de priorité élevée afin de garantir la qualité de service. À cet effet, un délai de séjour dans cette file est associé à chaque connexion qui s'y trouve par la station de base. Une fois ce délai écoulé, le trafic est transféré vers la file à priorité élevée. Cela permet également de satisfaire les exigences de délai maximum. Quant aux exigences d'allocation de bandes passantes minimum, elles sont satisfaites durant une fenêtre de temps;
  - file à priorité élevée : outre les demandes migrées de la file intermédiaire, elle stocke les requêtes périodiques de garantie de bande passante et les opportunités d'envoi de paquets dans la prochaine trame.

Rapport-Gratuit.com

2.3.3.9 Trames et messages de gestion

Les trames dans NS3 sont représentées en terme de durées elles ont donc un début et une fin dans le temps.
Chaque trame est subdivisée en sous-trames, une pour la liaison montante et l'autre pour la liaison descendante. Le mode de duplexage temporel étant celui implémenté, les liaisons montantes et descendantes utilisent la même fréquence, elles sont juste séparées dans le temps. Chaque sous-trame est constituée de rafale de paquet. Cette dernière constitue l'unité de transmission au niveau MAC. Elle est implémentée par la classe *PacketBurst* et est constituée d'une liste de paquet. La génération des sous-trames des liaisons montantes et descendantes est implantée par les classes *BSScheduler* et *UplinkScheduler*. Celle de la liaison descendante est simulée en transmettant des instances de la classe *PacketBurst*. La sous-trame de la liaison montante est divisée en slot. Chaque slot étant composé de rafales transmises par la station mobile. Une trame est alors divisée en un nombre entier de symboles et de slot physique (Physical Slot, PS) afin de gérer de manière efficiente la bande passante. Le nombre de symboles par trame dépend des paramètres de la couche physique comme la modulation, le taux de codage. La taille des rafales est alors spécifiée en termes de symbole (Ismail, Piro et al. 2010).

2.3.3.10 Message de gestion

Le message de gestion joue un rôle fondamental au niveau MAC il transporte des informations de contrôles et sont transmis en via les connexions de gestion. Le module implémente plusieurs messages de gestions nous présentons dans les sections suivantes les plus importantes à savoir : DL-MAP, UL-MAP, DCD et UCD (Ismail, Piro et al. 2010).Leurs positions dans la trame et leurs contenues sont montrées dans la figure 2.6.

Les messages DL-MAP et UL-MAP définissent l'accès aux sous-trames des liaisons montantes et descendantes dans la trame MAC. Ils sont constitués d'élément d'information

(Information Element, IE) et d'autres champs comme le montre la figure 2.3. Le message DL-MAP comporte un champ notifiant le temps de début permettant aux stations mobiles de passer en mode veille et d'en sortir une fois le temps écoulé ce champ n'est pas utilisé, car le module n'implémente pas la gestion de la puissance. Le même champ existe dans le message UL-MAP et sert à notifier le début du service accordé. Ce dernier contient aussi le champ qui indique la durée du service accordé. Quant à l'élément d'information (Information Element, IE), il contient un identifiant de connexion indiquant la station mobile ou le groupe de station mobile qui recevra la prochaine rafale, un code UIUC (*UplinkInterval User Code*) ou un code DIUC (*DownlinkInterval User Code*) indiquant le profil à utiliser afin de recevoir ou d'envoyer les paquets. Le profil est un ensemble de paramètres de la couche physique. Les profils des liaisons descendantes et montantes sont définis dans les messages qui décrivent les liaisons montantes et descendantes (DCD et UCD) (Ismail, Piro et al. 2010).

En plus des profils, les messages de description des liaisons (DCD et UCD) comportent des informations sur le canal comme ont peut le voir à la figure 2.6. La gestion et la création de ces messages sont implémentées par la structure *BurstProfileManager*. Le message UCD contient la limite du backoff et la taille des opportunités pour la télémétrie et pour la requête de bande passante. Le message DCD contient entre autres l'intervalle de transition réception/ émission (Receive/Transmit Transition Gap, RTG), l'intervalle de transition émission/réception (Transmit/Receive Transition Gap, TTG), l'identifiant de la trame courante et sa durée. Le profile contient une combinaison de code de modulation de code FEC et le DIUC ou UIUC servant d'identifiant unique pour le profile comme le montre la figure 2.6 (Ismail, Piro et al. 2010).

Le DL-MAP et UL-MAP sont envoyés dans chaque trame et le DCD et l'UCD sont envoyés de manière aléatoire. À la réception d'un DL-MAP, la station mobile vérifie à travers les IE si la trame lui est destinée, si oui elle le décode avec les paramètres du profile correspondant dans le cas où elle reçoit le UL-MAP lui indiquant une opportunité de service accordé elle envoie alors ses données en attente jusqu'à écoulement de la durée du service accordé (Ismail, Piro et al. 2010).

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

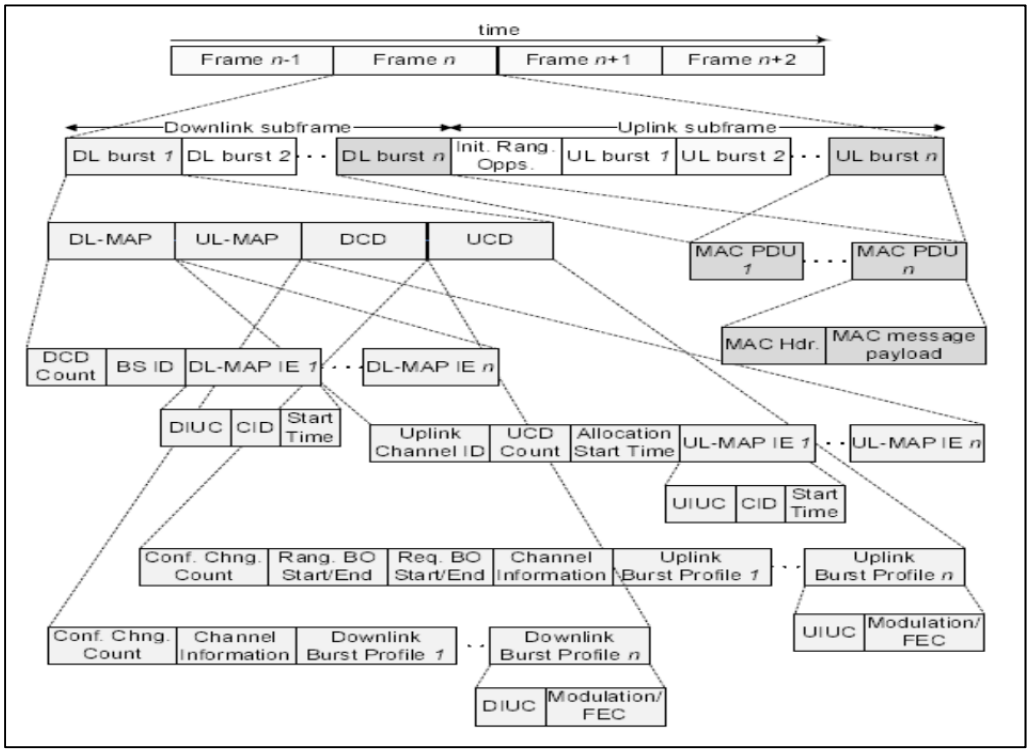


Figure 2.6 Format des trames MAC et des messages de gestions
Tirée de Farooq et Turletti (2009, p 6)

2.3.3.11 Flux de service

Le flux de service définit une connexion unidirectionnelle de transport de flux de PDU MAC assujetti à une même qualité de service. Il est implémenté par les classes *ServiceFlow* et *QoSParameterSet*. La structure de donnée *ServiceFlowManager* de la station de base est responsable de sa création et de sa gestion. Le standard prévoit deux mécanismes pour sa création, celui initié par la station de base et celui initié par la station mobile. C'est ce dernier qui est implémenté par le module WiMAX de NS3 (Farooq et Turletti, 2009; Ismail et al., 2010).

Malgré toutes ces implémentations, le module reste incomplet. La fragmentation et la défragmentation des PDU ainsi que la suppression des entêtes des paquets, la prise en charge de l'OFDMA restent des fonctionnalités à améliorer et à implémenter pour ce module. De plus, il ne prend pas en charge la topologie maillée, l'ARQ et l'adaptation des liens OFDMA(Farooq et Turletti, 2009; Ismail et al., 2010).

2.3.3.12 OFDMA dans NS3

Dans le cadre de ce mémoire, nous nous attelons plus sur la prise en charge de l'OFDMA. On constate dans la littérature que peu de choses ont été faites concernant l'OFDMA dans NS3. Aucune version de NS3 ne comprend un module prenant en charge l'OFDMA.

On retrouve néanmoins un greffon implémenté par Furlong et Guha, 2010. Ce dernier comporte 34 nouveaux fichiers, 41 nouvelles classes et une douzaine de classes modifiées (Furlong et Guha, 2010a). Nous présentons dans les prochaines sections les principaux éléments de cette implémentation.

2.3.3.13 Élément de réseaux (*Devices*)

Les classes des éléments de réseaux (*devices*) ont été étendues afin d'implémenter les fonctionnalités requises par l'OFDMA. Par ailleurs, des modifications ont été aussi apportées aux classes du modèle afin qu'elles prennent en charge la logique de l'OFDMA.

2.3.3.14 Phase de télémétrie

Contrairement à l'OFDM, l'OFDMA utilise les codes CDMA pour la télémétrie initiale, bien que celle-ci possède des étapes similaires à l'OFDM. Aussi, toutes les différences sont ajoutées au niveau de la station mobile. Par ailleurs, concernant la télémétrie périodique, elle

peut se faire soit en utilisant les codes CDMA soit en utilisant les messages présents dans l'OFDM.

Dans le modèle implémenté, un paquet unique d'un octet est ajouté. Celui-ci emprunte le même chemin que le reste de la salve et se substitue au code CDMA. (Furlong et Guha, 2010a)

2.3.3.15 Message de gestion et ordonnancement

De nouvelles classes ont été créées afin de prendre en charge les messages de gestion tandis que des modifications ont été également effectuées sur les anciennes. Celles-ci se traduisent par la création d'une classe mère englobant les messages communs à l'OFDM et l'OFDMA ainsi que la création de deux autres classes filles. Elles contiennent respectivement les parties non communes de l'OFDM et l'OFDMA (Furlong et Guha, 2010a).

Par ailleurs, de nouvelles structures *OfdmaTddMap*, *OfdmaTddDlMap* et *OfdmaTddUlMap* ont été créées pour la prise en charge des notions de symboles et de slots conférant ainsi un aspect bidirectionnel aux messages UL_MAP et DL_MAP. Elles implémentent aussi d'une carte des régions en termes d'identifiant de connexion. Celle-ci se compose d'une matrice bidimensionnelle qui représente l'ensemble des slots qui définisse les régions de données. À cet effet, en plus des identifiants de connexion (CID) un autre identifiant a été ajouté permettant ainsi une représentation des régions de données en deux dimensions. Le modèle comporte également des fonctions pour l'allocation par octet des slots, le comptage des slots ainsi que la possibilité de rechercher des régions utilisées suivant les différents identifiants (Furlong et Guha, 2010a).

Tous les services d'ordonnancement ont été modifiés afin de supporter le paradigme de l'OFDMA. En effet, deux ordonnanceurs ont été implémentés : *UplinkSchedulerSimpleOfdma* et *UplinkSchedulerRtpsOfdma*. Elles héritent des fonctionnalités de l'OFDM. Ainsi, le premier analyse les requêtes en bande passante de chaque station mobile et les sert par ordre d'enregistrement de flux de service correspondant au niveau de la station de base. Le second, quant à lui, prend en compte les conditions du canal et essaie de partager la bande passante entre les stations mobiles à servir (Furlong et Guha, 2010a)

La gestion de la puissance, la création d'une sous-couche physique pouvant modéliser les effets de collision au niveau sous-canal, la sous-couche de sécurité, la gestion du transfert intercellulaire et les possibilités d'avoir des trames avec une proportion de liaisons montantes et descendantes dynamiques sont des fonctionnalités qui n'ont pas été prises en charge par cette extension OFDMA (Furlong et Guha, 2010a)

Le modèle proposé utilise TDD OFDMA qui permet de diviser la bande de fréquence entre plusieurs connexions. Le nombre de symboles y est fonction de la taille de la trame, du taux d'échantillonnage et de la permutation partielle (Partial Use of Sub-Carriers, PUSC) qui est utilisée. L'allocation des symboles à la liaison montante et descendante est faite fixe. Les paramètres de la couche physique y sont initialisés pour un cas fixe: la fréquence est fixée à 10 MHz avec une transformée de Fourier rapide de 1024 points (Furlong et Guha, 2010a). Le temps de transmission y est calculé en se basant sur la quantité de données à envoyer et la capacité des slots de données. Cependant, le calcul de la capacité des slots de données ne prend en compte qu'un seul type de modulation et ne prend pas en charge la direction de communication. Aussi, les auteurs suggèrent de perfectionner la couche physique (Furlong et Guha, 2010a).

De plus dans le code du module proposé, des marques ont été mises indiquant des fonctions à parfaire, des traitements à valider. On peut citer entre autres le choix des profils, le calcul du

nombre de symboles disponibles pour l'allocation, la vérification de l'utilisation des symboles ou des slots dans le calcul du nombre de régions disponible et/ou utilisé sur la carte des régions de données, l'initialisation des paramètres physiques. Ainsi, dans le prochain chapitre nous présentons les principales modifications effectuées afin d'atteindre le modèle proposé. Ce dernier prend en compte toutes les modifications suggérées et aussi celles que nous avons détectées.

2.4 Conclusion

La présentation de l'outil NS3 nous permet d'en connaitre les concepts clés, les modules disponibles et les possibilités qu'il nous offre en matière de simulation. On constate donc que malgré la panoplie de modules proposés, du travail reste à faire afin de disposer d'un module réaliste qui permet de simuler le WiMAX avec la modulation OFDMA. Ainsi, nous présentons dans le prochain chapitre le modèle que nous proposons à cet effet qui se base sur le modèle de (Furlong et Guha, 2010a).

CHAPITRE 3

MODÈLE PROPOSÉ

3.1 Introduction

Après avoir effectué une présentation de la norme WiMAX ainsi qu'une description du simulateur NS3, la présentation du modèle implémenté est faite. Ainsi, cette section aborde premièrement les objectifs du projet ainsi que la méthodologie mise en place pour les atteindre. Les problèmes rencontrés y sont ensuite identifiés et classifiés dans deux sous-groupes distincts. Finalement, les divers algorithmes, améliorations et extensions de fonctionnalités ajoutées dans le modèle y sont décrits.

3.2 Objectifs et méthodologie

3.2.1 Objectifs

L'objectif principal de ce travail est d'apporter des améliorations et de possibles extensions de fonctionnalité au simulateur NS3 dans le but d'obtenir un simulateur supportant l'OFDMA offrant un débit supérieur ou égal à celui l'OFDM ainsi que de meilleurs délais de traitements lors de la simulation de réseaux WiMAX supportant l'OFDMA. Pour y parvenir, une version de NS3et greffon pour OFDMA, ont été fournis par le partenaire industriel. Le greffon est présenté dans (Furlong et Guha, 2010a). Il n'a pas encore été intégré dans une version officielle de NS3. À cet effet, cet article fait office de la seule documentation disponible.
L'article, le greffon et une version de NS3 propre à notre partenaire industriel ULTRA ELECTRONICS TCS nous ont été fournis par ce dernier.

Pour réaliser le modèle, le greffon a été intégré à une machine Linux distribution UBUNTU où a été installée au préalable la version du simulateur fourni par le partenaire industriel. Du

fait de la différence entre la version pour laquelle le correctif a été conçu et celle fournie par le partenaire, des changements ont dû y être apportés afin de permettre au greffon de fonctionner. Une fois cette étape franchie, l'environnement de développement ÉCLIPSE a été installé et configuré dans le but de pourvoir accéder aisément au code source, le modifier et d'y faire des tests de point de vue de la programmation. Une copie du système a été effectuée ainsi nous disposons de deux versions une appelé initiale et l'autre à laquelle nous apportons des modifications. Le système initial sert de référence de comparaison lors des diverses simulations.

3.2.2 Méthodologie

Une fois la plateforme de travail mise en place, il a été nécessaire d'adopter une méthodologie. Celle-ci s'établit en quatre grandes étapes :

- dans un premier temps, toutes les erreurs et fonctions à développer et celles notifiées dans le code lui-même grâce aux marqueurs TODO et FIXME sont accumulées et analysées. On y note aussi les possibilités d'améliorations constatées en auditant le simulateur à la lumière du standard IEEE 802.16-2009;
- en second lieu, toutes les possibilités d'améliorations sont regroupées suivant qu'elles soient reliées à la couche physique ou à la couche MAC;
- ensuite, leur impact sur la simulation est vérifié, des solutions sont proposées le cas échéant;
- enfin, les solutions proposées sont implémentées et des tests sont effectués pour vérifier l'impact des améliorations apportées par le modèle en comparaison au modèle initial.

3.3 Identifications et classifications des améliorations

Les projets de développement de logiciel, où plusieurs acteurs interviennent, utilisent le gestionnaire de code source afin de gérer les différentes versions et les différents apports de

tout un chacun au code. À cela s'ajoutent aussi des marqueurs spécifiques qui peuvent être insérés par le biais de l'environnement de développement. Ces derniers permettent de notifier dans le code source les différentes tâches et améliorations à effectuer. Ainsi, les marqueurs ont servi à identifier les possibles améliorations du modèle initial. Ils ont été recensés au nombre de cinquante. La liste a été épurée en y soustrayant les tâches générées automatiquement par l'environnement de développement lui-même ainsi que les tâches répétitives. Leur impact sur la simulation a ensuite été vérifié.

Les défauts constatés ont également été notés. Pour le constat de ces derniers, les diverses fonctions du modèle OFDMA proposé par le greffon ont d'abord été exécutées et des traces effectuées afin de s'assurer que les enchainements suivis respectent bien le standard IEEE 802.16-2009. Ensuite, le traitement opéré par les fonctions elles-mêmes a été analysé permettant de faire apparaître certains dysfonctionnements par rapport à ce que propose le standard. Onze fichiers doivent être modifiés afin de régler les problèmes recensés. Ces derniers ont été regroupés suivant l'influence que le problème a sur la couche physique ou sur la couche MAC. Le résultat est consigné dans le tableau 3.1.

Le tableau contient le nom de chaque fichier qui sera modifié. Ces noms sont représentatifs des fonctionnalités qui sont traitées par ces fichiers et donc des fonctionnalités qui seront améliorées par nos travaux. Ainsi le fonctionnement de la couche physique est revu afin de permettre de supporter différents types de modulations et différentes fréquences. Au niveau de la couche MAC la gestion des profils de paquets la création et la gestion des messages de gestion (DL_MAP) et des modifications mineures au niveau des ordonnanceurs afin de refléter la vision bidimensionnelle des ressources au niveau OFDMA,

Rapport-Gratuit.com

Tableau 3.1 Regroupement des corrections par fichier et par couche

Couche physique	**Couche MAC**
bs-net-device-ofdma.cc simple-ofdma-Wimax-phy.cc ss-net-device-ofdma.cc	bs-scheduler-ofdma-simple.cc bs-uplink-scheduler-cdrr-ofdm.cc bs-uplink-scheduler-cdrr-ofdma.cc bs-uplink-scheduler-rtps-ofdma.cc bs-uplink-scheduler-simple-ofdma.cc ofdma-tdd-dl-map.cc ofdma-tdd-map.cc ss-scheduler-ofdma.cc

3.4 Présentation des améliorations apportées

3.4.1 Amélioration de la couche physique

La couche physique dans NS3 sert à initialiser les paramètres physiques ainsi qu'a recevoir et envoyer des salves. À cette fin, plusieurs paramètres sont calculés. Cependant, force est de constater que les fonctions ont été conçues avec des valeurs fixes comme le nombre de points de la transformée de Fourier rapide (Fast Fourier Transform, FFT) ou le type de modulation. Ces valeurs étant amenées à changer lors d'une simulation, des extensions de fonctionnalité ont été développées et des améliorations ont été apportées. Celles-ci prennent en compte, dans les calculs, la direction de la liaison, la modulation et le mode de permutation. Une fois effectuées, elles permettent d'avoir plusieurs possibilités de simulation.

3.4.2 Détermination de la ressource nécessaire pour l'envoi d'une slave

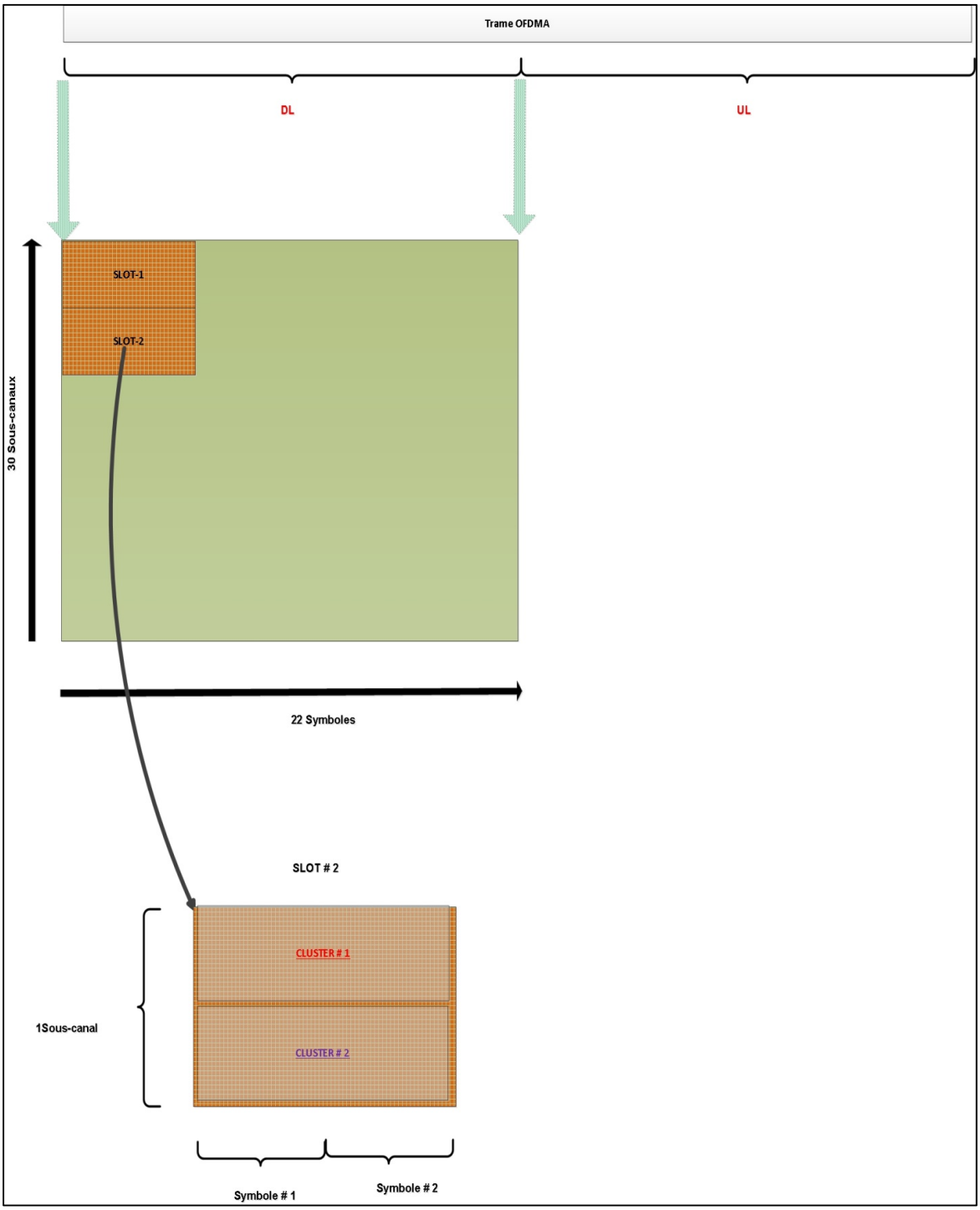


Figure 3.1 Décomposition de la trame OFDMA
Tirée d'une communication privée avec le partenaire industriel

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

La communication sur un lien se résume à l'envoi et à la réception des trames dans la direction montante et/ou descendante. À cet effet, les paquets sont traités par l'ordonnanceur afin de déterminer leur ordre d'envoi. Aussi, pour être envoyés, ils sont regroupés dans une trame, elle-même constituée par des salves. Les ressources disponibles sont donc allouées à chaque salve. Il devient alors primordial de savoir la quantité de ressources nécessaires pour transmettre une salve.

Par ailleurs, le slot est l'unité d'allocation en OFDMA. Comme le montre la figure 3.1, la portion liaison descendante de trame est allouée à une zone de données bidimensionnelles formées par un ensemble de slot. Déterminer la quantité de ressource nécessaire pour traiter cette trame revient à déterminer la taille de la zone qui lui sera allouée, donc le nombre de slot qui la constitue. Cette donnée est nécessaire pour une salve d'une taille donnée. Pour ce faire, il faut aussi connaitre la capacité d'un slot. Celle-ci dépend du nombre de symboles OFDMA, du nombre de sous-canaux qui le forme ainsi que du type de modulation utilisée. Le nombre de symboles et de sous-canaux formant un slot dépend quant à lui de la direction de la liaison du type de permutation utilisée. Dans notre cas le type de permutation convenu avec le partenaire industriel est le PUSC.

Dans la version actuelle de NS3, le nombre de slots à utiliser pour envoyer un paquet est calculé indépendamment du type de modulation, de la direction de la transmission et du schéma de permutation utilisé. Le calcul effectué ne répond qu'à la modulation QPSK-1/2 pour une liaison montante. Elle divise la taille de la salve en octet par 6 (le nombre d'octets de deux symboles en liaison montante avec le mode PUSC) comme le montre l'équation (3.1).

$$Nombre\ de\ slots = \frac{taille\ de\ la\ salve}{6} \tag{3.1}$$

Tirée du code source du greffon de Guha et Furlong

Une solution a été proposée afin d'y remédier. Elle passe par un nouvel algorithme pour le calcul du nombre de slots qui reflète les exigences de la norme. Ainsi, selon le standard 802.16-2009, pour une permutation PUSC, un slot est représenté pas un sous canal et deux symboles OFDMA pour la liaison descendante voir figure 3.1. Pour la liaison montante quant à elle, le slot est représenté par un sous-canal et trois symboles OFDMA.

De plus la section 8.4.6.1.2.1 du standard IEEE 802.16-2009 à la table 443, 24 sous-porteuses de données par symbole OFDMA ainsi que 30 sous-canaux par symbole OFDMA sont utilisés pour le PUSC en liaison descendante avec une FFT de 1024. La table 456 dans la section 8.4.6.2 du standard 802.16-2009 définit aussi pour le PUSC en liaison montante 24 sous-porteuses de données par symbole OFDMA et 35 sous-canaux par symbole OFDMA pour une FFT de 1024. Ces paramètres ont servi de base pour la réalisation de l'algorithme de détermination de la ressource nécessaire pour une salve. Mais l'algorithme est adapté pour supporter d'autres paramètres.

L'algorithme 3.1 reçoit en entrée la direction, le type de modulation et la taille du paquet en octet. Grâce à la direction, le nombre de sous-porteuses par symbole ainsi que le nombre de symboles par slot sont déterminés suivant la norme. La modulation quant à elle permet de déterminer le nombre de bits par sous-porteuses de données ainsi que le taux de codage utilisé. La multiplication de toutes ces valeurs entre elles permet d'avoir la charge utile en bits d'un slot. Une fois cette charge connue, le nombre de slots est déterminé par un rapport de la taille de la salve par la charge utile.

```
DÉBUT
        /* Soient nb_Dc_parSymbole et nb_SymboleparSlot le nombre de porteuses de données par symbole et le nombre de symboles par slot.

        Soient nb_BitparDc et taux_Codage respectivement le nombre de bits par porteuses de données et le taux de codage.

        Soient nb_bitparSlot et nb_Slot respectivement le nombre de bits d'un slot et le nombre
        de slots.
        Soient :
        Direction la direction de la transmission
        Modulation la modulation utilisée
        Taille la taille du paquet en octet */
        FAIRE:

                SELON Direction

                        Déterminer nb_Dc_parSymbole          /*le nombre de sous porteuses de données par symbole*/

                        Déterminer nb_SymboleparSlot          /*le nombre de symboles par slot*/

                FIN SELON

                SELON Modulation

                        Déterminer nb_BitparDc                /*le nombre de bits par porteuses de données*/

                        Déterminer taux_Codage                /*le taux de codage*/

                FIN SELON

                nb_bitparSlot ← nb_Dc_parSymbole * nb_SymboleparSlot *  nb_BitparDc
        * taux_Codage

                nb_slot ← taille / (nb_bitparSlot*8)
          FIN FAIRE

FIN
```

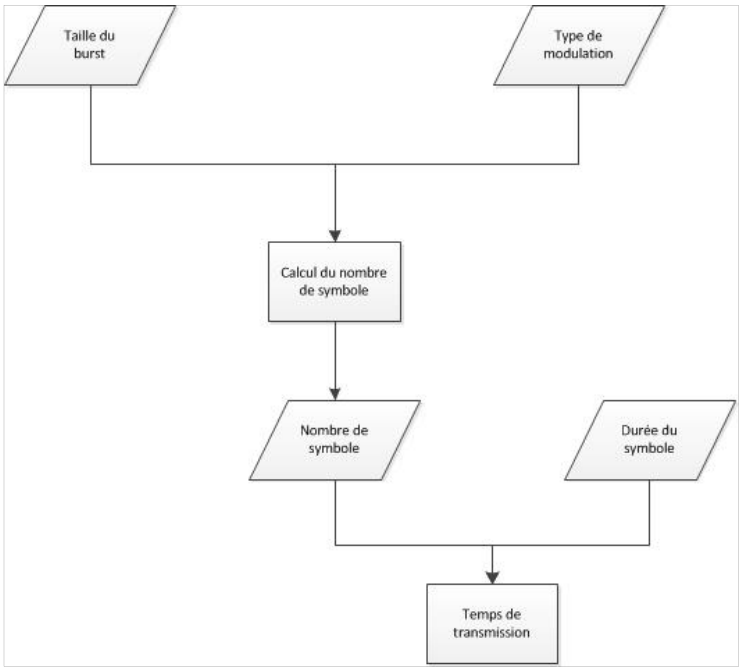
Algorithme 3.1 Algorithme de calcul du nombre de slots

Une fois la quantité de ressources à utiliser déterminée, le calcul de la durée de la transmission, du débit de transmission et du nombre de bits envoyé, valeurs importantes au niveau du simulateur, a été revu. Des possibilités d'améliorations furent détectées. Dans les prochaines sections, les solutions apportées sont présentées.

3.4.3 Détermination de la durée de la transmission

La durée de transmission représente le temps que prend une salve pour être transmise. Dans la version actuelle du modèle OFDMA, elle n'est pas calculée à partir des paramètres de la couche physique. Elle a une valeur indépendamment de la taille du paquet et de la modulation utilisée. Une solution est donc proposée à travers l'algorithme 3.2 de calcul.

Ainsi, dans le but de calculer la durée de transmission, la taille de la salve est utilisée, ainsi que le type de modulation et la durée du symbole. Par ailleurs, la taille de la salve et le type de modulation ont servi à déterminer le nombre de slots à utiliser pour la transmission de la salve. Une fois le nombre de symboles formant le slot déterminé, il est multiplié par la durée du symbole afin d'obtenir la durée de la transmission.



Algorithme 3.2 Algorithme de détermination du temps de transmission

3.4.4 Détermination de la charge utile de symbole

La charge utile est une valeur utilisée directement pour le calcul d'autres paramètres. Elle sert à réaliser l'allocation et d'autres fonctions dans le simulateur. Afin de la déterminer, l'algorithme 3.3 de calcul de la charge utile d'un symbole en fonction de la modulation est utilisé. Comme dit précédemment, dans le cadre de ce projet, le type de permutation convenue est le PUSC et le nombre de sous-porteuses de données (24) ne change pas, que ce

soit en liaison montante ou en liaison descendante. De ce fait, la direction n'influence pas nos calculs.

```
Nom de la méthode : Calcul_Charge_utile

    Description : Calcule la charge utile en fonction du nombre de symboles et de la
modulation

    En-tête : Calcul_Charge_utile (nb_Symbole, modulation)

    /* Soient nb_Dc_parSymbole et nb_Octet respectivement le nombre de porteuses de
    données par symbole et le nombre d'octets (la charge utile).

    Soient nb_BitparDc et taux_Codage respectivement le nombre de bits par porteuses de
    données et le taux de codage.
    Soit nb_Symbole le nombre de symboles */
    DEBUT

      SELON modulation

         Déterminer nb_BitparDc

         Déterminer taux_Codage

      FIN SELON

      nb_Octet ← nb_Symbole * nb_Dc_parSymbole * nb_BitparDc * taux_Codage / 8

     RETOURNER nb_Octet

    FIN
```

Algorithme 3.3 Algorithme de calcul de la charge utile de symbole

3.4.5 Détermination du débit de la transmission

Le débit de la transmission permet d'avoir la capacité transmise en fonction de la durée. Il est utilisé dans le simulateur afin de déterminer la taille d'un bloc en fonction de la durée de transmission. Dans la version actuelle de NS3, il a une valeur fixe et ne fait pas l'objet d'un

calcul. Il retourne en effet la valeur zéro comme on peut le voir sur la figure 3.2. Ceci a été corrigé par l'équation (3.2) proposée. Elle représente le rapport entre la charge en bits d'un symbole et sa durée.

$$Debit_transmision_{(b/s)} = \frac{nb_bit_parSymbole}{duree_Symbole\ _{(s)}} \tag{3.2}$$

où :

- nb_bit_parSymbole est calculé avec l'algorithme de la charge utile qui prend en compte la modulation et le taux de codage;
- duree_Symbole est la durée d'un symbole en seconde.

```
uint32_t SimpleOfdmaWimaxPhy::DoGetDataRate (ModulationType modulationType) const
{
    return 0;

 }
```

Figure 3.2 Capture d'écran de la fonction de calcul du débit de la transmission

3.4.6 La création et le traitement du DL_MAP

Le message DL_MAP est défini dans la table 40 du standard IEEE 802.16-2009. Il décrit les paramètres à utiliser pour la liaison descendante. Il contient plusieurs valeurs, dont l'identifiant de la station de base et des valeurs qui décrivent le canal et servent à spécifier le profil de salve à utiliser ainsi que le champ de synchronisation. Il comporte aussi les éléments d'informations dont un en marque le début et un autre la fin. Dans NS3, afin de traiter le message DL_MAP, il est parcouru du début jusqu'à la fin dans le but d'en extraire toute l'information. Ce parcours est basé sur les éléments d'informations qui le délimitent.

Cependant, la fin du message DL_MAP n'est pas spécifiée, une modification a été faite en ajoutant un élément d'information à la fin du message DL_MAP avec un code DIUC (*Download Interval Usage Code*) qui correspond à celui qui marque sa fin.

3.4.7 Détermination des paramètres initiaux de la couche physique

Afin de réaliser la simulation, plusieurs configurations des paramètres de la couche physique doivent être réalisées en particulier le nombre de slots physiques par symbole, la fréquence d'échantillonnage, la taille du slot et sa durée.

Le nombre de slots physiques par symbole est déterminé par l'équation (3.3) :

$$PS_{par\ symbole} = \frac{durée\ d'un\ symbole(s)}{d_PS\ (s)} \tag{3.3}$$

Tirée du standard IEEE 802.16-2009

où d_ PS est la durée du PS en seconde est calculée par l'équation (3.4);

$$d_PS\ (en\ seconde) = \frac{4}{Fe_{(MHz)}} \tag{3.4}$$

Tirée du standard IEEE 802.16-2009

où Fe est la fréquence d'échantillonnage.

Cependant, le résultat obtenu à l'issue de ce calcul est donné sous une forme décimale. Dans NS3, l'arrondi de celui-ci se fait à la borne supérieure de la valeur. Cela représente une

source d'erreur notamment dans l'allocation de ressources de la ressource disponible. Aussi, afin de remédier à cela, la valeur a été arrondie à la borne inférieure. Il en est de même pour la fréquence d'échantillonnage dont l'équation de calcul est la (3.5). L'arrondi effectué influence aussi sur la valeur de la durée du slot physique donc du calcul du nombre de slot physique par symbole

$$Fe = \left[\frac{n * Bw}{8000}\right] * 8000 \tag{3.5}$$

Tirée du standard IEEE 802.16-2009

où n est le facteur d'échantillonnage et Bw la largeur de la bande passante en Hz.

Quant au calcul du nombre total de symboles, de la taille d'un slot en symbole et du nombre de sous-canaux, il est effectué indépendamment de la direction. Toutes ces valeurs sont initialisées suivant celles des tables 456 et 443 du standard IEEE 802.16-2009 corrigeant ainsi la situation.

3.5 Amélioration de la couche MAC

Les principales améliorations apportées à la couche MAC ont été effectuées au niveau des ordonnanceurs. Parmi les valeurs reçues en entrée par l'ordonnanceur dans NS3, il y a la liste des connexions, le nombre de sous-canaux et le nombre de symboles. Cela permet de calculer d'autres valeurs nécessaires à l'allocation des ressources à savoir : le nombre de symboles requis pour une transmission et le nombre total de symboles disponibles. Ce calcul prend en compte le sens de la liaison ainsi que le type de modulation utilisé.

Rapport-Gratuit.com

3.6 Détermination de la ressource requise et de la ressource disponible

Le nombre de slots à utiliser était initialement fixé au niveau de la couche physique dans NS3. Il a été rendu dynamique en fournissant le nombre de slots requis dépendamment de la direction dans le cas de la permutation PUSC. Ainsi, l'algorithme de calcul du nombre de slots nécessaire pour une salve a été utilisé. Il permet de déterminer le nombre de slots requis au niveau de l'ordonnanceur. Cela a un effet direct sur le calcul du nombre de slots disponibles. On a ainsi un ordonnancement plus précis dû à la précision dans le calcul de la ressource requise et celle disponible.

3.7 Détermination de la taille des régions de la DL_MAP

La région de données selon la section 8.4.3.1 du standard 802.16-2009 représente une zone d'allocation en OFDMA. Elle est représentée en sous-canaux et en nombre de symboles OFDM contigus. La figure 3.3 représente un exemple de régions de données formées pas 3 sous-canaux et 4 symboles.

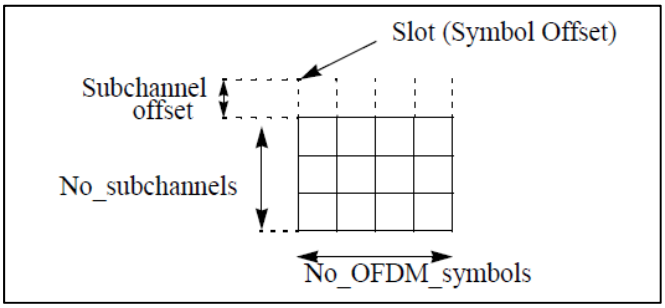


Figure 3.3 Exemple de région de données
Tirée du standard 802.16-2009

Dans NS3, la taille de la région de données est calculée avec des valeurs de nombre de symboles fixes. Le nombre de symboles qui forme une zone est initialement fixé à deux. Ce qui a été corrigé en proposant un calcul basé sur la direction et la modulation. À cette fin, la fonction proposée dans la couche physique pour le calcul du nombre de slots et le calcul de charge utile est utilisée. Cela permet de déterminer le nombre de symboles et de sous-canaux qui forment la région de données à allouer à une salve. Cette technique est réalisée dans l'optique d'obtenir une allocation précise.

3.8 Conclusion

Ce chapitre présente les améliorations apportées au simulateur. Mis à part celles-ci, d'autres améliorations mineures ont été effectuées comme l'utilisation des bons profils de slave et la suppression des variables locales après utilisation afin de libérer de la mémoire. Tout ceci permet d'obtenir un modèle proposé plus réaliste et plus adéquat en termes de flexibilité d'utilisation.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

CHAPITRE 4

SIMULATION ET ANALYSE DES RÉSULTATS

4.1 Introduction

Le modèle proposé regroupe les améliorations et les extensions de fonctionnalités. Aussi, afin de comparer le modèle proposé à l'initial, plusieurs simulations ont été effectuées. Ce chapitre présente donc l'architecture de test et les différentes techniques mises en place à cet effet, et fournit également une analyse des résultats des simulations.

4.2 Architecture de test

La simulation avec NS3 débute par la modélisation du test à effectuer. Ainsi, une fois celle-ci réalisée, le système modélisé est transformé en script utilisé pour sa transmission au simulateur. La simulation peut y être alors exécutée afin d'en extraire les résultats comme le montre la figure 4.1. Le script est réalisé à partir de deux langages : le C++ ou le Python. Pour des considérations de flexibilité, notre choix s'est porté sur le langage C++.

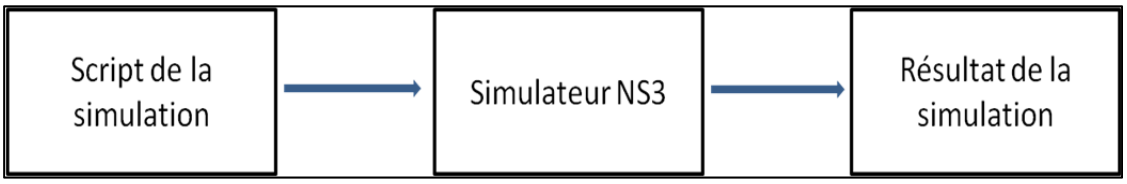


Figure 4.1 La réalisation d'une simulation dans NS3

4.2.1 Présentation de l'architecture réseau

Cette section présente le réseau utilisé afin de réaliser la simulation. Celui-ci comporte une station de base et une station mobile comme le montre la figure 4.2.

Figure 4.2 Architecture du réseau

La configuration et l'initialisation des divers nœuds de l'architecture reviennent à spécifier :

- le type de couche physique : il peut être du WI-FI, du WiMAX, du LTE, etc. Un nœud ne supporte qu'un seul type de couche physique;
- le type de nœud : il est choisi entre la station de base et la station mobile suivant les besoins de la simulation. Un nœud ne peut avoir qu'un seul type;
- l'adresse IP : elle est attribuée suivant une plage d'adresse et un masque réseau;
- le type du canal de communication: il peut être filaire ou sans fil (WI-FI, WiMAX, LTE etc.). Il doit être le même pour les deux nœuds désirant communiquer;
- l'ordonnanceur : le choix est possible entre quatre ordonnanceurs dans NS3;
- le type de modulation.

Une fois le réseau mis en place, la simulation peut être effectuée. Celle-ci fournit des résultats à analyser et interpréter. La méthode d'extraction de ces derniers est présentée dans la section suivante.

4.2.2 Méthodes d'extraction des résultats

Plusieurs informations sont générées après la simulation. Parmi celles-ci, des métriques doivent être extraites pour la validation du modèle. Dans NS3, ces dernières peuvent être obtenues à partir des informations recueillies à travers les fichiers traces, les fonctions post-traitement ou les fichiers journaux ou PCAP (*Packet CAPture*). Cependant, les fichiers traces nécessitent beaucoup d'espace de stockage, quand bien même une optimisation est faite au niveau des échantillons stockés. En effet, les fichiers traces sont des fichiers textes contenant un évènement par ligne, décrit avec des marquages temporels et d'autres propriétés extraites de la simulation. Chaque fichier trace est associé à un objet spécifique et identifié par un nom. Par ailleurs, les fichiers traces influencent sur la performance de la simulation à cause des multiples opérations d'entrées et de sorties. Leurs utilisations allongent la durée des simulations de 6 à 7 fois contrairement à celles où les statistiques sont collectées durant la simulation. Quant aux fichiers PCAP (*Packet CAPture*), ils contiennent des informations brutes comme les captures de paquets transmis et reçus. En extraire des métriques nécessite un effort de programmation supplémentaire. Il en est de même pour le système de fonctions post-traitement (Carneiro, Fortuna et Ricardo, 2009). Mettre en place une méthode d'extraction permettant d'extraire les données durant la simulation avec un minimum d'effort de programmation serait donc la panacée.

Le module *FLOWMONITOR* peut être utilisé afin de faciliter l'extraction des informations durant la simulation. Dans ce dernier, la liste des flux à contrôler est détectée par le contrôleur de flux lors de la simulation sans intervention du programmeur. Il est subdivisé en trois parties principales comme le montre la figure 4.3 (Carneiro, Fortuna et Ricardo, 2009) :

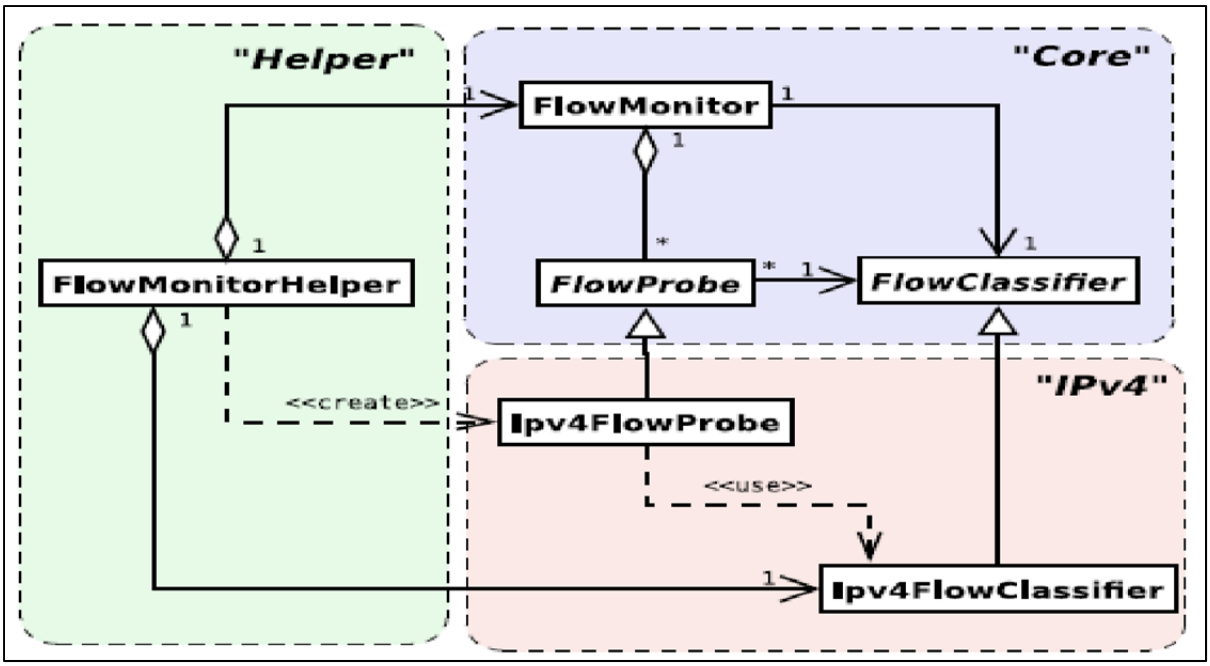


Figure 4.3 Architecture du contrôleur de flux
Tirée de Carneiro Fortuna et Ricardo (2009)

- le groupe Core contient :
  - *FlowMonitor* : il permet de collecter les statistiques d'un flux de bout en bout;
  - *FlowProbe* : il collecte les statistiques pour un flux à un moment particulier (le passage par un nœud par exemple);
  - *FlowClassifier* : il permet de créer un identifiant unique de flux et de paquet qui sert à les identifier partout dans le simulateur.
- le groupe IPv4 : il est une spécialisation du groupe Core. Il fournit les classes *Ipv4FlowProbe* et *Ipv4FlowClassifier* qui, à la différence des classes *FlowProbe* et *FlowClassifier*, sont basées sur les entêtes IP TCP/ UDP des paquets (adresse IP source, adresse IP destination, le protocole, le port source, le port destination);
- le groupe *Helper* : il contient la classe *FlowMonitorHelper* qui permet de faciliter la construction du *FlowMonitor*.

L'extraction de données d'une simulation grâce au module *FlowMonitor* revient à créer une instance de *FlowMonitorHelper*, de *FlowMonitor*, de *Ipv4FlowClassifier* et plusieurs instances d'*Ipv4FlowProbe* à raison d'un par nœud. Ainsi l'*Ipv4FlowProbe* capture les paquets, assigne un identifiant à chaque paquet et rapporte les divers événements au FlowMonitor qui les collecte afin de calculer les statistiques. Les différentes valeurs calculées ainsi que leurs descriptions sont consignées dans le tableau 4.1.

Tableau 4.1 Données des statistiques
Tiré de Carneiro, Fortuna et Ricardo(2009)

Nom	**Description**
Timefirsttxpacket	Représente le temps absolu où le premier paquet du flux a été transmis.
Timelasttxpacket	Représente le temps absolu où le dernier paquet du flux a été transmis.
Timefirstrxpacket	Représente le temps absolu où le premier paquet du flux a été reçu par le nœud destinataire.
Timelastrxpacket	Représente le temps absolu où le dernier paquet du flux a été reçu par le nœud destinataire.
Delaysum	Représente la somme des délais de bout en bout pour les paquets du flux reçu.
Jittersum	Représente la somme des variations de délai de bout en bout pour les paquets du flux reçu.
Txbytes, txpackets	Représente respectivement le nombre total d'octets et de paquets transmis pour un flux
Rxbytes, rxpackets	Représente respectivement le nombre total d'octets et de paquets reçu pour un flux
Lostpackets	Représente le nombre de paquets perdus : un paquet est supposé perdu s'il n'a jamais été reçu ou s'il a été envoyé depuis plus de 10 s.
Timesforwarded	Il représente la somme du nombre de fois que chaque paquet du flux a été envoyé.
Delayhistogram, jitterhistogram, packetsizehistogram	Ils représentent respectivement la version en histogramme du délai de la variation du délai et de la taille des paquets.
Packetsdropped, bytesdropped	Ils représentent le nombre de paquet et d'octet perdus suivant la cause de la perte (pas de route, expiration du TTL ou mauvais checksum)

Les données sont contenues dans les diverses statistiques collectées par l'instance de *Flow Monitor* et celle de *FlowProbe*. Chacune d'elles dispose d'une structure *FlowStat* dans laquelle sont consignées les statistiques. Les différents paramètres contenus dans *FlowStat* sont présentés dans le tableau 4.3. Ils représentent l'ensemble de paramètres que l'on peut extraire.

À partir de ces statistiques, d'autres métriques sont calculées selon nos besoins. En ce qui concerne le présent travail, trois principales métriques ont été choisies pour analyser le modèle proposé :

- le délai moyen par paquet,
- la variation moyenne du délai,
- le débit réel de simulation.

Par ailleurs, à chaque simulation le débit moyen de transmission, le débit moyen de réception le taux de paquets perdus sont aussi extraits. Les équations sont utilisées pour calculer toutes ces métriques à partir des données statistiques extraites.

Le délai moyen est calculé grâce à l'équation (4.1). Il est obtenu en faisant le rapport du délai moyen par le nombre de paquets reçus. Les différentes variables utilisées à savoir *delaySum* en seconde et *rxpackets* sont présentées dans le tableau 4.1.

$$Délai\ moyen = \frac{delaySum}{rxPackets} \tag{4.1}$$

Tirée de Carneiro Fortuna et Ricardo (2009)

La variation de délai moyen est calculée grâce à l'équation (4.2). Elle est obtenue en faisant le rapport de la somme des variations de délai de bout en bout par le nombre d'intervalles entre paquets reçus. Les différentes variables utilisées à savoir *jitterSum* en seconde et *rxpackets* sont présentées dans le tableau 4.1

$$Variation\ de\ délai\ moyen\ = \frac{jitterSum}{(rxPackets - 1)} \tag{4.2}$$

Tirée de Carneiro Fortuna et Ricardo (2009)

Le débit moyen de la transmission est calculé grâce à l'équation (4.3) où Btx est le débit moyen transmis en bit par seconde. Il est obtenu en faisant le rapport du nombre total d'octets transmis par le délai entre le premier paquet et le dernier paquet transmis. Les différentes variables utilisées à savoir *txBytes*, *timeLastTxPacket* en seconde et *timeFisrstTxPackets* en seconde sont présentées dans le tableau 4.1

$$Btx\ = \frac{8 * txBytes}{(timeLastTxPacket\ -\ timeFirstTxPacket)} \tag{4.3}$$

Tirée de Carneiro Fortuna et Ricardo (2009)

Le débit moyen de la réception est calculé grâce à l'équation (4.4) où Brx est le débit moyen reçu en bit par seconde. Il est obtenu en faisant le rapport du nombre total d'octets reçus par le délai entre le premier paquet et le dernier paquet reçu. Les différentes variables utilisées à savoir *rxBytes*, *timeLastRxPacket* en seconde et *timeFisrstRxPackets* en seconde sont présentées dans le tableau 4.1

$$Brx\ = \frac{8 * rxBytes}{(timeLastRxPacket\ -\ timeFirstRxPacket)} \tag{4.4}$$

Tirée de Carneiro Fortuna et Ricardo (2009)

Le taux de paquet perdu est calculé grâce à l'équation (4.5) où q est taux de paquets perdu. Il est obtenu en faisant le rapport du nombre total de paquets perdus par la somme du nombre de paquets reçus et du nombre de paquets perdus. Les différentes variables utilisées à savoir *lostPackets*, *rxPacket* et *lostPackets* sont présentées dans le tableau 4.1

$$q = \frac{\text{lostPackets}}{(\text{rxPackets} + \text{lostPackets}} \tag{4.5}$$

Tirée de Carneiro Fortuna et Ricardo (2009)

4.2.3 Paramètres et critères de comparaison

Le scénario retenu pour le test consiste à réaliser une application UDP client-serveur. Le serveur et le client sont configurés et installés sur deux nœuds différents. Un flux de données peut-être ainsi généré et envoyé d'un nœud à un autre dépendamment du sens de la liaison (montante ou descendante) qu'on veut simuler.

À cet effet, les nœuds doivent tout d'abord être configurés comme le montre le tableau 4.2.

Tableau 4.2 Paramètres de configuration des nœuds et du client UDP

Paramètres	**Valeurs**
Type de la couche physique	`WimaxHelper::SIMPLE_PHY_TYPE_OFDMA,` `WimaxHelper::SIMPLE_PHY_TYPE_OFDM`
Type de nœud	`WimaxHelper::DEVICE_TYPE_SUBSCRIBER_STATION` `WimaxHelper::DEVICE_TYPE_BASE_STATION`
Adresse IP	`10.1.1.1` `10.1.1.2`
Type du canal de communication	OFDM, OFDMA
Ordonnanceur	`WimaxHelper::SCHED_TYPE_SIMPLE`
Type de modulation	`WimaxPhy::MODULATION_TYPE_QPSK_12`

Rapport-Gratuit.com

Par la suite, le client et le serveur UDP sont créés et le port sur lequel ce dernier communique est spécifié avant de les installer sur les nœuds appropriés. L'adresse IP du serveur, le port de communication, la taille des paquets, l'intervalle de temps entre deux paquets, le temps de mise en service et d'arrêt du client sont fournis au client UDP. L'intervalle de temps entre deux paquets permet de spécifier la cadence d'envoi des paquets. Dans la simulation il est incrémenté de 0.1 seconde à chaque fois. Tous ces paramètres sont consignés dans le tableau 4.3.

Tableau 4.3 Paramètres de configuration du client UDP

Paramètres	**Valeurs**
Taille des paquets	1024 octets
Intervalle de temps entre deux paquets	$0.12 + 0.1 * j$, j : nombre d'itérations
Temps de mise en service du client	Durée de la simulation
Temps d'arrêt du client	Durée de la simulation +0.1

Une fois l'application prête à être utilisée, il faut procéder à la création des règles de classification des flux de service. Elles servent à traiter les paquets. À cet effet, la spécification des adresses IP source et destination du flux, des intervalles de port source et destination, du numéro du protocole de communication utilisé (dans notre cas c'est le port 17 pour l'UDP) et la priorité du flux de communication est nécessaire. Les valeurs utilisées de tous ces paramètres ainsi que de la durée de la simulation sont dans le tableau 4.4.

Tableau 4.4 Paramètres des règles de classification et durée de la simulation

Paramètres		**Valeurs**
Règle de classification	Adresse IP source	10.1.1.0
	Adresse IP de destination	10.1.1.1
	Intervalle de port source	0 à 65000
	Intervalle de port destination	100 à 100
	Protocole de communication	17
	Priorité du flux	1
Durée de la simulation	30 secondes	

Une fois la règle de traitement de flux de service mise en place, le flux de services est créé en spécifiant la direction du flux (montante ou descendante), le type de paquet supporté (UGS, RTPS. NRTPS, BE) et la règle de classification utilisée. Le flux de service est alors ajouté aux nœuds, achevant ainsi la partie du script concernant le scénario.

La partie du script modélisant le scénario écrit, la partie d'extraction de données a été implémentée. Celle-ci a consisté en la création d'une instance de Flow Monitor et à son installation sur tous les nœuds ainsi qu'à des instructions de calcul et d'affichage des paramètres calculé sur la sortie standard comme le montre la portion de script présentée sur la figure 4.4.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

```
/* Configuration de l'objet flowmonitor */
FlowMonitorHelper flowmon_helper;
Ptr<FlowMonitor> monitor = flowmon_helper.InstallAll();//activation du contrôle du flux sur tous les noeuds
monitor->SetAttribute("DelayBinWidth", ns3::DoubleValue(0.00001));
monitor->SetAttribute("JitterBinWidth", ns3::DoubleValue(0.00001));
monitor->SetAttribute("PacketSizeBinWidth", ns3::DoubleValue(20));
Simulator::Stop (Seconds (duration + 0.1));
monitor->SerializeToXmlFile("resultat_wimax.xml", true, true);

/* Affichage des résultats sur la sortie standard */
Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier>(flowmon_helper.GetClassifier ());
std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats();

for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator i =stats.begin (); i != stats.end (); i++)
{
Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow (i->first);
std::cout << "Flux " << i->first << " ( " << t.sourceAddress << " ->" << t.destinationAddress << " )\n";
std::cout << "    Nombre de paquet transmis:   " << i->second.txPackets << "\n";
std::cout << "    Nombre de paquet reçus:   " << i->second.rxPackets << "\n";
std::cout << "    Nombre d'octets transmis:   " << i->second.txBytes << "\n";
std::cout << "    Nombre d'octets reçus:   " << i->second.rxBytes << "\n";
std::cout << "    Heure de la premiere reception:    " << (i->second.timeFirstRxPacket) << "\n";
std::cout << "    Heure de la derniere reception:    " << (i->second.timeLastRxPacket)<< "\n";
std::cout << "    Debit de l'emmission: " << (i->second.txBytes * 8)/(double ((i->second.timeLastTxPacket -i->second.timeFirstTxPacket).GetSeconds())) << " bps\n";
std::cout << "    Debit de la reception:   " << (i->second.rxBytes * 8)/(double((i->second.timeLastRxPacket - i->second.timeFirstRxPacket).GetSeconds())) << " bps\n";
std::cout << "    Délai moyen:  " << (i->second.delaySum).GetSeconds()/(double (i->second.rxPackets)) << " s\n";
std::cout << "    Variation moyen du délai:  " << (i->second.jitterSum).GetSeconds()/(double (i->second.rxPackets - 1)) <<" s\n";
std::cout << "    Taille moyen du paquet reçus:  " << (i->second.rxBytes)/(double (i->second.rxPackets)) << " octets\n";
std::cout << "    Nombre de paquet perdu pas le moniteur de flux Packets lost:  " << (i->second.lostPackets) << " paquets\n";
std::cout << "    Nombre de paquet perdu:  " << (i->second.txPackets -i->second.rxPackets) << " paquets\n";
std::cout << "    Taux de paquet perdu :  " << (i->second.txPackets - i->second.rxPackets)/(double (i->second.txPackets))*100 << "%\n\n";
}
```

Figure 4.4 Capture d'écran montrant la portion de script d'extraction de données.

La prochaine section présente les résultats des différentes simulations effectuées afin de comparer le modèle proposé à celui initial. Ces résultats sont analysés et interprétés.

4.3 Résultats

Les résultats présentés ci-dessous sont ceux des diverses simulations effectuées. Ils peuvent être classés en deux groupes : celles effectuées avant les améliorations et extensions et celles effectuées après. Aussi, le premier groupe est constitué de simulations effectuées avec une couche physique supportant l'OFDM et une autre supportant l'OFDMA. Les courbes obtenues portent la légende «OFDM» et «OFDM.Avant». Le deuxième groupe, quant à lui,

contient des simulations effectuées juste avec une couche physique supportant l'OFDMA les courbes obtenues portent la légende «OFDMA.Avant».

Pour chaque type de couche physique, une simulation est réalisée pour chaque type de paquet (UGS, NRTPS, RTPS, BE). De plus, pour chaque type de paquet, l'intervalle de temps entre paquets varie par pas de 0.10 seconde. La direction utilisée pour la simulation est la descendante. Une bande de fréquence de 10 MHz est utilisée afin de pouvoir comparer nos résultats à ceux du système initial qui ne supporte que cette bande. Par ailleurs, pour chaque simulation, le débit, le délai moyen par paquet et la variation moyenne du délai ont été calculés. Les courbes obtenues sont présentées dans la section suivante.

4.3.1 Analyse des résultats

Dans cette section, les valeurs obtenues pour le modèle proposé seront comparées à ceux du système initial et à ceux du module WiMAX de la version officielle de NS3 qui supporte l'OFDM. Ainsi si le débit du modèle proposé est supérieur à celui du modèle initial et les délais sont moindres ou égaux le modèle est alors dit plus réaliste.

4.3.1.1 Délai moyen

La figure 4.5 présente les délais moyens obtenus pour les transmissions en OFDM ET OFDMA dans les simulations.

Les valeurs du délai moyen en fonction de l'intervalle entre paquets gardent une similarité relative pour les paquets de types BE et NRTPS. Ainsi, on constate que le délai moyen obtenu avec le modèle avant les améliorations avoisine les 97 ms, soit plus de 6 fois les valeurs obtenues pour le modèle officiel de NS3 supportant l'OFDM qui est de 15 ms.

Cependant, on observe un bond de 25 ms quand l'intervalle entre paquets atteint 52 ms. Le modèle après amélioration quant à lui a un délai moyen, pour ces deux types de paquets, de 2.5 ms indépendamment de l'intervalle entre les paquets. Il est 40 fois plus bas que celui du modèle initial et 10 fois plus bas que celui de l'OFDM pour ces deux types de paquets. **Le modèle proposé est donc plus réaliste, car on constate une grande amélioration des délais qui se comparent bien à ceux de l'OFDM et au modèle initial**.

Pour les paquets RTPS, la valeur du délai moyen varie entre 7 et 3 ms avec le modèle après les améliorations. Avec le modèle de base, les valeurs obtenues restent constantes (environ 200 ms) avec un pic à 4.95 s pour un intervalle entre paquets de 0.12 s. En ce qui concerne l'OFDM les valeurs sont plus ou moins stables. En effet, elles varient autour de 15ms. On constate donc que le délai moyen avec les améliorations apportées prend des valeurs de 2 à 5 fois plus basses que ceux du système OFDM et plus de 25 fois plus basses que celui avec l'OFDMA dans le système initial pour ce type de paquet. **Le modèle proposé est donc plus réaliste, car on constate une grande amélioration des délais qui se comparent bien à ceux de l'OFDM et au modèle initial**.

Concernant les paquets UGS dans le système amélioré, le délai moyen tourne autour de 6ms avec un pic à 11 ms pour un intervalle entre paquets de 0.42 ms. Une baisse à 4 ms est observée à la même valeur de l'intervalle avec l'OFDMA dans le système initiale. De plus, les valeurs oscillent autour de 7 ms, sauf la valeur de 12ms observée pour un intervalle entre paquets de 0.12 ms. On note aussi une constance pour les valeurs de l'OFDM. Elles oscillent entre 19 ms et 24 ms avec une valeur moyenne de 22 ms. Elles connaissent également un pic de 24 ms à un intervalle entre paquet de 0.12 ms et une baisse de 19 ms pour un intervalle de 0.42 s. Ainsi, on constate que le délai moyen obtenu pour ce type de paquet est d'environ 6ms ce qui est plus bas que celui du système initial qui est de 7 ms.

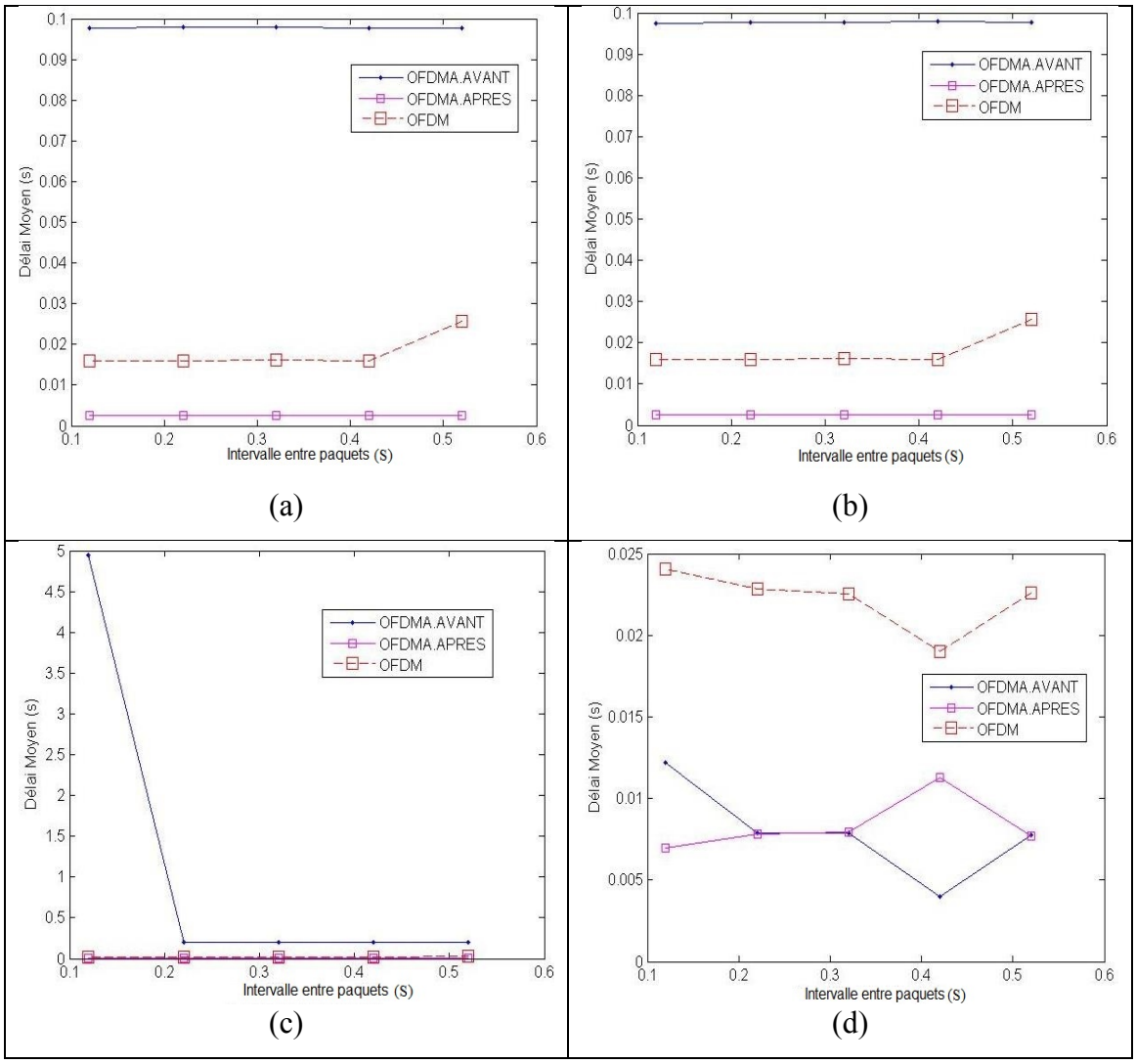


Figure 4.5 Délai moyen par intervalle entre paquets de types :
(a) BE; (b) NRTPS; (c) RTPS; (d) UGS

4.3.1.2 Variation moyenne du délai

Les courbes des résultats des simulations pour la variation moyenne du délai sont présentées à la figure 4.6.

Pour les paquets de types BE avec l'OFDMA dans le système proposé ou dans le système initial, la variation moyenne du délai croît avec l'intervalle entre les paquets. Cette augmentation de valeur varie de 0.05 ms à 0.2 ms.

La tendance pour les paquets de types NRTPS est la même que celle des paquets de types BE à la différence que la courbe de l'OFDMA avec le modèle amélioré croit plus vite au début et décroit à l'intervalle entre les paquets de 0.42 ms avant de remonter.

En ce qui concerne l'OFDM, les courbes sur les figures 4.6 (a) et 4.6 (b) évoluent en dents de scie. En effet, elles croissent jusqu'à un intervalle entre paquets de 0.42 s puis décroissent à cette valeur avant de croitre à nouveau. Elles prennent des valeurs entre 0.4 ms et 0.7 ms pour les deux types de paquets. La variation moyenne du délai du modèle proposé varie dans le même intervalle que celle du modèle initial et est inférieure à celle obtenue dans le modèle officiel de NS3 supportant l'OFDM d'envions 10 fois.

Pour les paquets de types UGS, la variation moyenne du délai évolue également en dent de scie que ce soit pour le modèle initial ou pour le modèle proposé : elle croit de 0.02 à 6.7 ms avec la valeur de l'intervalle entre paquets, puis décroit à 0.09 ms avant de remonter jusqu'à 6.5 ms. Il en est de même pour l'OFDM. Elle évolue entre 0.4 et 0.72 ms. On constate que pour ce type de paquet le système malgré les améliorations à la même valeur de variation moyenne de délai même si l'intervalle entre paquets varie. Cette valeur est supérieure à celle de l'OFDM sauf pour un intervalle entre paquets de 0.42 ms.
Pour les paquets de types NRTPS, BE et UGS, **le modèle proposé est donc plus réaliste, car on constate une grande amélioration des délais qui se comparent bien à ceux de l'OFDM et au modèle initial**.

Pour les paquets de types RTPS avec le modèle amélioré, la variation moyenne croit avec l'intervalle entre paquets de 0.07 jusqu'à la valeur de 0.3 ms. Quant au système de base utilisant OFDM, elle croit de 0.04 ms à 0.6 ms, décroit à 0.5 puis remonte vers 0.7 ms. Par ailleurs, avec le système initial utilisant OFDMA, les valeurs augmentent de 0.02 à 6.7 ms puis descendent à 0.09 ms avant de remonter jusqu'à 6.5 ms. On constate que pour ce type de paquets les variations moyennes du délai, mis à part pour l'intervalle 0.12 ms, ont quasiment les mêmes valeurs qui sont inférieures à celle du système OFDM.

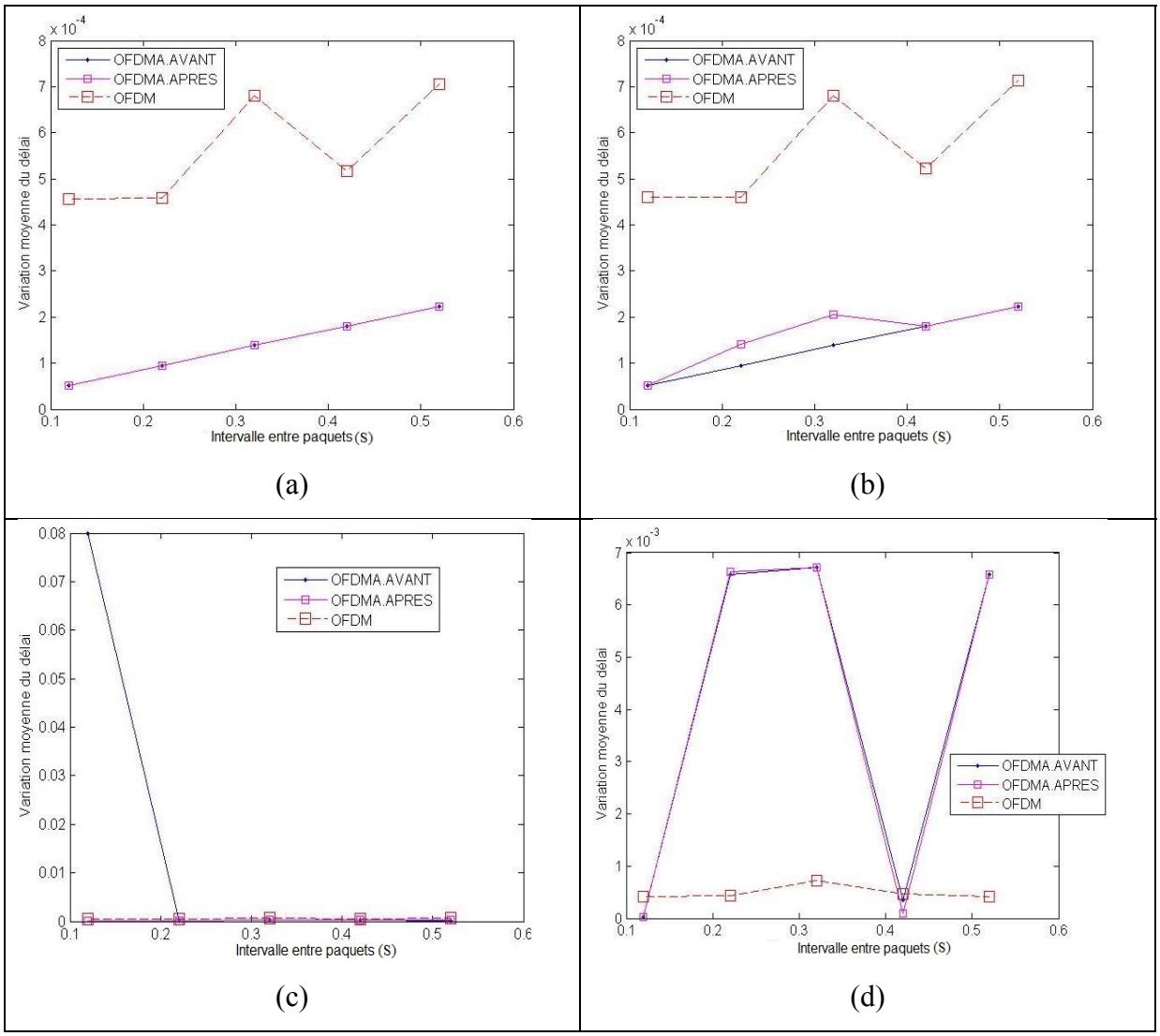


Figure 4.6 Variation moyenne du délai par intervalle entre paquets de types :
(a) BE; (b) NRTPS; (c) RTPS; (d) UGS

4.3.1.3 Débit réel de simulation

Les différents débits obtenus en fonction de l'intervalle entre les paquets pour les quatre types de services sont présentés dans la figure 4.7. L'ensemble des courbes montre que le débit décroit quand l'intervalle entre les paquets croit. Ainsi, pour les systèmes OFDM et le module amélioré OFDMA, le débit décroit d'environ 70 500 à 16 533 b/s, quelques soit le type de paquet. Pour le système de base OFDMA, il varie de 21 786 à 8494 b/s pour les paquets RTPS et d'environ 36 188 à 8 448 b/s pour les autres paquets. **Le modèle proposé est donc plus réaliste, car on constate une grande amélioration des débits qui se comparent bien à ceux de l'OFDM et au modèle initial**.

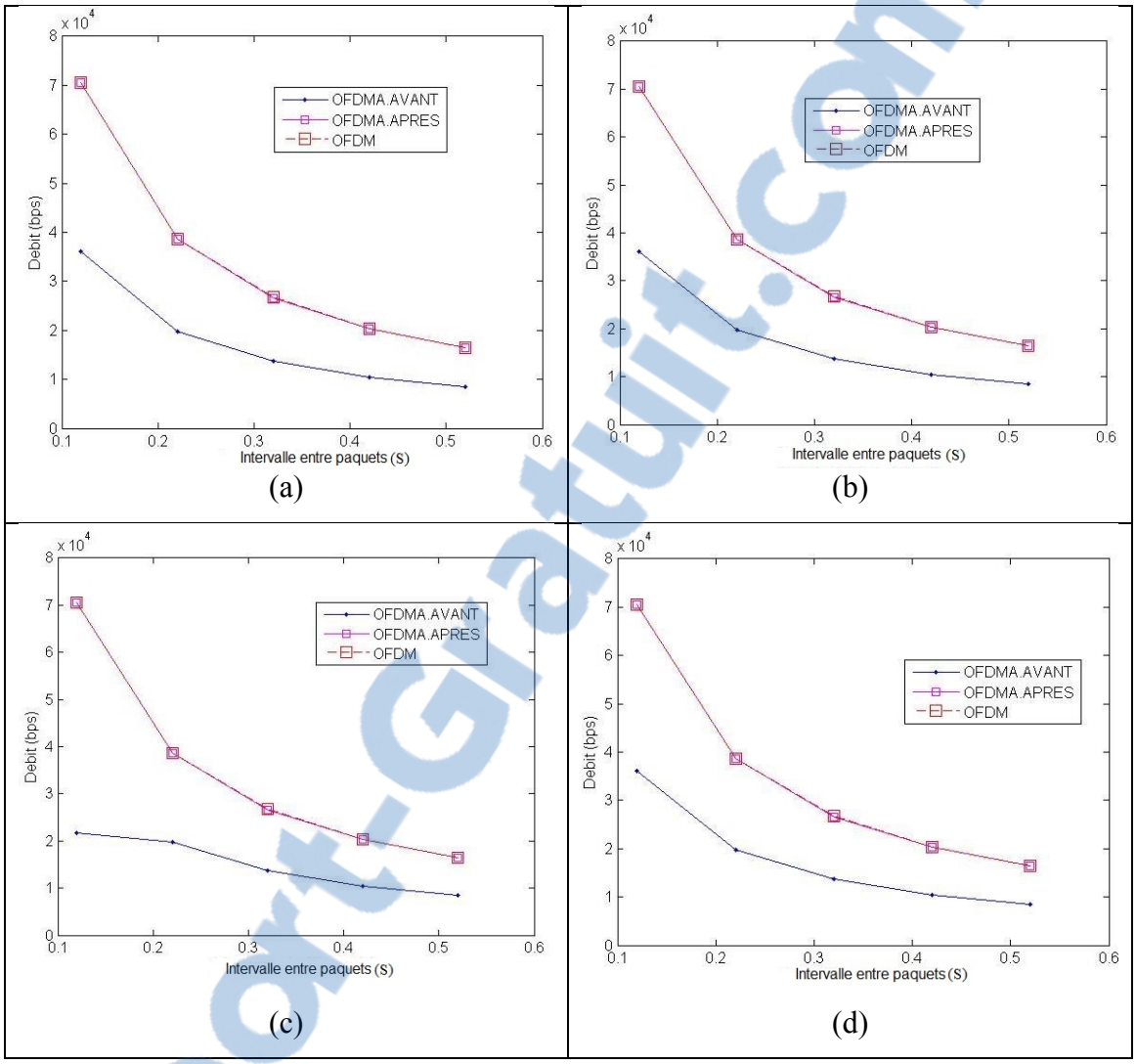


Figure 4.7 Débit par intervalle entre paquets de types :
(a) BE; (b) NRTPS; (c) RTPS; (d) UGS

4.4 Conclusion

Des simulations ont été réalisées pour l'OFDM et l'OFDMA sur le modèle initial et celui proposé. Ces valeurs ont servi de valeur témoin afin de pouvoir évaluer notre modèle. Aussi, après plusieurs observations, on constate que le modèle proposé offre une grande amélioration des débits et des délais qui se compare bien à ceux du système officiel de NS3. Ces résultats viennent donc renforcer le réalisme de notre modèle.

Rapport-Gratuit.com

CONCLUSION

Le présent mémoire a été un cadre d'apprentissage et de mise en pratique des notions engrangées tout au long de la formation ainsi qu'un creuset pour apprendre et comprendre des systèmes comme le WiMAX et NS3.

Dans le premier chapitre du mémoire, les différentes évolutions de la norme ainsi que les fonctionnalités qu'elles offrent dans les couches physiques et MAC sont présentées. Le fonctionnement et les traitements du WiMAX effectués dans NS3 sont décrits dans le chapitre 2. Après cela, le traitement effectué dans NS3 a été analysé à la lumière de ce que propose le standard 802.16. De plus, les possibilités d'améliorations ont été notées et des solutions proposées conduisant à un modèle amélioré. Ce modèle est l'objet du chapitre 3. Les simulations effectuées montrent de meilleurs débits et délais par rapport au modèle initial permettant ainsi d'affirmer qu'il est plus réaliste.

Cependant, le modèle proposé ne prend pas en compte la gestion des puissances et ne propose pas d'algorithme sophistiqué d'ordonnancement. Ces points pourront faire l'objet de futurs travaux.

BIBLIOGRAPHIE

Andrews, Jeffrey G., Arunabha Ghosh et Rias Muhamed. 2007a. « MAC Layer of WiMAX ». In *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*.

Andrews, Jeffrey G., Arunabha Ghosh et Rias Muhamed. 2007b. « PHY Layer of WiMAX ». In *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*.

Andrews, Jeffrey G., Arunabha Ghosh et Rias Muhamed. 2007c. « WiMAX Network Architecture ». In *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*.

Bacioccola, A., C. Cicconetti, C. Eklund, L. Lenzini, Z. Li et E. Mingozzi. 2010. « IEEE 802.16: History, status and future trends ». *Computer Communications,* vol. 33, n° 2, p. 113-123.

Cipriano, A. M., P. Gagneur, G. Vivier et S. Sezginer. 2010. « Overview of ARQ and HARQ in beyond 3G systems ». In *Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), 2010 IEEE 21st International Symposium on*. (26-30 Sept. 2010), p. 424-429.

Dias, Ricardo V., Jo˜ao P. Leite, Rafhael M. Amorim, Andr´e N. Barreto, Paulo Henrique P. de Carvalho et Robson D. Vieira. 2010. « Performance Analysis of HARQ in WiMAX Networks Considering Imperfect Channel Estimation ». In.

Etemad, K. 2008. « Overview of mobile WiMAX technology and evolution ». *Communications Magazine, IEEE,* vol. 46, n° 10, p. 31-40.

Farooq, Jahanzeb, et Thierry Turletti. 2009. « An IEEE 802.16 WiMAX Module for the NS-3 Simulator ». In.

Furlong, W. P., et R. Guha. 2010a. « OFDMA Extension of NS-3 WiMAX Module ». In *Computer Modeling and Simulation (EMS), 2010 Fourth UKSim European Symposium on*. (17-19 Nov. 2010), p. 426-431.

Furlong, William P., et Ratan Guha. 2010b. « OFDMA Extension of NS-3 WiMAX Module ». p. 426-431.

Henderson, Thomas R., Sumit Roy, Sally Floyd et George F. Riley. 2006. « ns-3 project goals ». In *Proceeding from the 2006 workshop on ns-2: the IP network simulator*. (Pisa, Italy), p. 13. 1190468: ACM.

IETF. 1983. *Time Protocol*.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Ismail, M. A., G. Piro, L. A. Grieco et T. Turletti. 2010. « 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), p. 1-10. 1808223: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

NSNAM. 2011. « ns-3 Overview ». En ligne. p. 40. < http://www.nsnam.org/docs/ns-3-overview.pdf >. Consulté le 25 juillet 2012.

NSNAM. 2013. « ns-3 Tutorial ». En ligne. p. 115. < http://www.nsnam.org/docs/release/3.18/tutorial/ns-3-tutorial.pdf >. Consulté le 17 décembre 2013.

Odhiambo, Marcel O., et Amimo P.O. Rayolla. 2012. « The WiMAX PHY Layer ». In *Digital Communication*, sous la dir. de (Ed.), Prof. C Palanisamy.

Pareit, Daan, Bart Lannoo, Ingrid Moerman et Piet Demeester. 2012. « The History of WiMAX: A Complete Survey of the Evolution in Certification and Standardization for IEEE 802.16 and WiMAX ». *Communications Surveys & Tutorials, IEEE,* vol. 14, n° 4, p. 1183-1211.

So-In, Chakchai, Raj Jain et Abdel-Karim Tamimi. 2010. « Capacity Evaluation for IEEE 802.16e Mobile WiMAX ». *Journal of Computer Systems, Networks, and Communications,* vol. 2010.

Thomas, Christopher. 2011. « NS-3 SIMULATION OF WIMAX NETWORKS ». MS thesis. Washington, University in St. Louis, 66 p.

Weingartner, E., H. vom Lehn et K. Wehrle. 2009. « A Performance Comparison of Recent Network Simulators ». In *Communications, 2009. ICC '09. IEEE International Conference on*. (14-18 June 2009), p. 1-5.

ZeHua, Gao, Gao Feng, Zhang Bing et Wu ZhenYu. 2012. « Wimax core Network ». In *Digital communication*, sous la dir. de Palanisamy, C.: InTech.