

**Télécom Paris (ENST)  
Institut Eurécom**

**THESE**

A présenter pour obtenir le grade de docteur  
de l'Ecole Nationale Supérieure  
des Télécommunications

Spécialité: Informatique et Réseau

**Illia Racunica**

**Mécanismes de contrôle de Qualité de Service  
dans l'UMTS pour les applications sur IP**

Rapporteurs	S. Tohmé, PRISM, Université de Versailles (France) A.L. Beylot, ENSEEIHT (Toulouse, France)
Examineurs	G. Pujolle, LIP6 (Paris, France) M. Debbah, Institut Eurécom (Sophia Antipolis, France)
Directeur de thèse	C. Bonnet, Institut Eurécom (Sophia Antipolis, France)

Octobre 2004



# Remerciements

Je voudrais tout d'abord remercier mon directeur de thèse Christian Bonnet. Il m'a donné sa confiance dès le début de ces travaux, et l'a renouvelée lors des moments difficiles. Ses commentaires constructifs m'ont permis de progresser dans de très nombreux domaines, tant scientifiques que techniques.

Je voudrais aussi remercier Guy Pujolle qui a accepté de présider ce Jury. Il est à l'origine de mon intérêt pour les réseaux, ses cours et ses ouvrages ont toujours été pour moi de précieux outils de références.

Je n'oublie pas dans mes remerciements mes deux rapporteurs, Samir Tohmé et André Luc Beylot qui ont fait tout deux des commentaires très constructifs, et m'ont permis d'améliorer cet ouvrage. Je les en remercie avec gratitude.

Je voudrais également remercier Mérouane Debbah, qui a accepté d'être un examinateur de ma thèse.

Yan Moret m'a beaucoup aidé au début de mes travaux, je tiens également à le remercier. L'ensemble de l'équipe de la plate-forme d'Eurécom m'a également aidé pour ces travaux, je tiens notamment à remercier Lionel Gautier.

Aawatif Menouni a également été une aide très précieuse. Je ne crois pas que j'aurais pu finir ces travaux sans ses encouragements et ses idées pertinentes. Je la remercie donc tout particulièrement.

Enfin, je remercie l'ensemble des doctorants de l'Institut Eurécom, qui forment un groupe solidaire et sympathique. Leurs soutiens, aides et commentaires ont été très appréciés.



# Résumé

Les travaux de recherche présentés dans cette thèse portent sur la Qualité de Service (QoS) et l'allocation de ressources dans les réseaux UMTS-TDD. Les réseaux UMTS se composent de deux parties, une partie filaire qui peut être sous IP et peut utiliser Diffserv pour la gestion de la QoS, et une partie réseau d'accès radio (UTRAN) qui gère la transmission sur la partie radio. Le comportement des deux réseaux étant différent, la façon de gérer la QoS est également différente. Il est nécessaire pour chaque type d'application de connaître ses besoins sur les deux parties du réseau. Une application donnée aura donc deux descriptions de sa QoS, une dans la partie filaire et une dans l'UTRAN. Une correspondance est faite entre ces deux descriptions, ce qui permet à l'arrivée d'une application dans l'UTRAN de déterminer la QoS qui lui est maintenant nécessaire. Nous montrons pour quelques applications typiques comment nous définissons leurs descripteurs de QoS dans l'UTRAN et les associons à leurs descripteurs de QoS de la partie filaire.

Pour assurer la qualité des applications transportées, il est nécessaire de paramétrer l'UTRAN de façon appropriée. Une étude est réalisée pour déterminer le comportement de l'UTRAN en fonction de l'état du canal radio et de certains paramètres. Le nombre de paramètres pouvant être fixé est très important. Nous mettons en évidence quelques paramètres clefs, comme les codes correcteurs ou le nombre maximum autorisé de retransmissions des paquets de données particulièrement impactant sur le comportement de l'UTRAN. Cette étude permet d'évaluer les délais de transmission, les taux de pertes et l'efficacité de différents paramétrages. Il est ainsi possible de proposer un paramétrage de l'UTRAN adapté aux besoins de chaque type d'application. L'efficacité des paramétrages dépend de l'état du canal. Or, l'état du canal radio évolue. Un paramétrage qui peut être le plus efficace à un instant donné peut devenir moins efficace à un autre instant. Nous proposons donc un mécanisme de lien adaptatif qui permet d'accroître l'efficacité de notre système.

Dans une seconde étape, nous nous intéressons à l'allocation de ressources

dans le sens descendant (downlink). Après avoir défini la notion de ressources, nous proposons un algorithme d'allocation en relation avec l'algorithme de contrôle d'accès pour les applications de type Temps Réels (RT). Cet algorithme permet d'indiquer les ressources attribuées à un utilisateur et d'équilibrer la puissance de transmission sur les différents slots. Le nombre d'utilisateurs présents simultanément sur le réseau peut ainsi s'accroître tandis que le nombre d'utilisateurs refusés par le contrôle d'accès diminue.

Nous nous sommes également intéressé au trafic d'application Non Temps Réels (NRT). Les applications NRT ne subissent pas de contrôle d'accès mais peuvent être mises en attente et ordonnancées. L'algorithme d'ordonnancement communique avec le contrôle d'accès pour que le trafic RT ne soit pas perturbé par le trafic NRT. L'utilisation de notre algorithme d'allocation permet d'accroître la quantité de trafic NRT transmise. Le trafic NRT est ordonnancé en fonction de l'état des canaux radios des différents utilisateurs NRT. Cet ordonnancement adaptatif permet de réduire la quantité de paquets perdus en raison des mauvaises conditions radio. L'ordonnanceur utilisé traite cependant les différentes applications NRT de façon équitable.

# Abstract

The UMTS (Universal Mobile Telecommunications System) networks, third-generation mobile networks, will support high-speed multimedia services. This thesis concerns the UMTS-TDD with a radio access sub-system (UTRAN) directly connected to IPv6. The whole QoS is managed by the DiffServ protocol mechanisms. First, we propose a mechanism to map the DiffServ QoS to a UMTS DiffServ. Then, from this UMTS QoS, we propose a parameter tuning with respect to the QoS required by the applications that respects the delays and loss criteria. When several tuning parameters were possible, we selected the parameter that maximizes the system efficiency. Knowing that the most efficient parameter changes according to the channel characteristics, we have proposed and evaluated an adaptive tuning scheme that maximizes this efficiency.

Then, we have considered a system combining Real-Time (RT) and Non Real-Time applications (NRT). The real-time applications use dedicated resources (DCH), compared to the non real-time applications which use shared resources (DSCH). In order to maintain the current applications QoS, we have applied an access control and resources allocation mechanism to the RT applications. The access control does not accept an application if there is not enough available resources or if the use of the necessary resources for the new application will exceed a power threshold (in this case, the new application will decrease the current applications performances). The resources allocation mechanism allows by an appropriate (codes, power) combination, to decrease the amount of rejected real-time applications. Concerning the NRT applications, we propose a scheduling mechanism. This scheduling is called 'opportunistic'. It takes into account the channel state to select a flow. This decreases the amount of non real-time packets rejected. A credit mechanism allows this scheduling to be fair. Furthermore, the scheduling algorithm is linked to the access control. A DSCH can't be used except if its use doesn't decrease the current RT applications (threshold power control). We have proposed thereby some coherent mechanisms for the configuration

of multimedia applications in a UMTS-TDD network (the whole IPv6 core network)

# Table des Matières

Remerciement . . . . .	i
Résumé . . . . .	iii
Abstract . . . . .	v
Liste des Figures . . . . .	xi
Liste des Tableaux . . . . .	xv
Acronymes . . . . .	xvii
<b>1 Introduction</b>	<b>1</b>
<b>2 Architecture de Qualité de Service pour les applications multimédias</b>	<b>5</b>
2.1 Introduction . . . . .	5
2.2 Définition et critères de la Qualité de Service (QoS) . . . . .	6
2.2.1 Les Délais . . . . .	6
2.2.2 La Gigue (jitter) . . . . .	6
2.2.3 Les taux de perte . . . . .	7
2.3 Architecture générale de l'UMTS. . . . .	7
2.4 Diffserv . . . . .	9
2.4.1 Les classes de trafic Diffserv . . . . .	11
2.5 Architecture de la QoS dans le Core Network et l'UTRAN . . . . .	12
2.5.1 Quality of Service Information Element . . . . .	14
2.6 Classification des applications multimédias . . . . .	16
2.6.1 Classe conversational . . . . .	16
2.6.2 Classe Streaming . . . . .	16
2.6.3 Classe Interactive . . . . .	16
2.6.4 Classe Background . . . . .	17
2.7 Architecture de la QoS dans la plate-forme PLATON . . . . .	17
2.8 Traduction de la QoS DiffServ en QoS UMTS . . . . .	20
2.9 Exemples de traduction . . . . .	22
2.10 Conclusion . . . . .	24

<b>3</b>	<b>Paramétrage de l'UTRAN</b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	Architecture de l'UTRAN . . . . .	28
3.2.1	Radio Resource Control . . . . .	28
3.2.2	Packet Data Convergence Protocol . . . . .	29
3.2.3	Radio Link Control . . . . .	29
3.2.4	Medium Access Control . . . . .	36
3.2.5	Couche Physique . . . . .	36
3.2.6	Hybrid ARQ . . . . .	38
3.3	Choix des paramètres . . . . .	39
3.3.1	Conditions expérimentales . . . . .	40
3.3.2	Mesures sur les pertes . . . . .	40
3.3.3	Mesures sur les délais . . . . .	42
3.3.4	Calcul de l'efficacité des paramétrages . . . . .	52
3.4	Configuration d'un service . . . . .	56
3.4.1	Format de Transport . . . . .	57
3.4.2	Transport Format Set . . . . .	58
3.4.3	Transport Format Combination . . . . .	58
3.5	Paramétrage des Formats de Transport et des Combinaisons de Formats de Transport . . . . .	58
3.6	Conclusion . . . . .	60
<b>4</b>	<b>Lien adaptatif</b>	<b>61</b>
4.1	Introduction . . . . .	61
4.2	Aperçu des techniques de lien adaptatif dans les réseaux sans fils . . . . .	61
4.2.1	GPRS . . . . .	61
4.2.2	ATM sur satellite . . . . .	63
4.2.3	Hiperlan2 . . . . .	64
4.3	Proposition de lien adaptatif dans l'UMTS . . . . .	65
4.3.1	Evaluation de l'algorithme de lien adaptatif . . . . .	67
4.3.2	Application Non Real Time . . . . .	67
4.3.3	Application Real Time Interactive . . . . .	70
4.3.4	Application Real Time Non Interactive . . . . .	71
4.4	Conclusion . . . . .	72
<b>5</b>	<b>Contrôle d'accès et Allocation de Ressources</b>	<b>73</b>
5.1	Introduction . . . . .	73
5.2	Notion de Ressources Radio . . . . .	74
5.2.1	Principe du WCDMA . . . . .	74

---

5.2.2	Burst Format . . . . .	78
5.2.3	Time Division Duplex . . . . .	79
5.2.4	Notion de ressource . . . . .	79
5.3	Canaux de Transport . . . . .	79
5.3.1	Dedicated Channel . . . . .	80
5.3.2	Broadcast Channel . . . . .	80
5.3.3	Forward Access Channel . . . . .	81
5.3.4	Paging Channel . . . . .	81
5.3.5	Random Access Channel . . . . .	81
5.3.6	Downlink Shared Channel . . . . .	81
5.3.7	Uplink Shared Channel . . . . .	81
5.4	Coded Composite Transport Channel . . . . .	82
5.5	Contrôles d'accès . . . . .	82
5.6	Algorithmes d'allocation . . . . .	83
5.6.1	Max in slot . . . . .	83
5.6.2	Min in slot 1 . . . . .	83
5.6.3	Min in slot 2 . . . . .	83
5.6.4	Power Based . . . . .	85
5.7	Comparaisons des algorithmes . . . . .	86
5.8	Conclusion . . . . .	91
<b>6</b>	<b>Ordonnancement des services NRT en présence de services RT</b>	<b>93</b>
6.1	Introduction . . . . .	93
6.2	Rappel sur les ordonnancements . . . . .	94
6.2.1	Round Robin . . . . .	94
6.2.2	Weighted Round Robin . . . . .	94
6.3	WCFQ, Un algorithme d'ordonnancement opportuniste . . . . .	95
6.4	Notion d'équité . . . . .	96
6.5	Adaptation de WCQF à l'UMTS . . . . .	96
6.6	Expériences . . . . .	97
6.6.1	RT sur DCH, NRT sur DSCH . . . . .	99
6.6.2	RT sur DCH ou DSCH, NRT sur DSCH . . . . .	104
6.6.3	Conclusion . . . . .	107
<b>7</b>	<b>Intégration sur la plate-forme UMTS d'Eurécom</b>	<b>109</b>
7.1	Introduction . . . . .	109
7.2	Le serveur de canal . . . . .	110
7.3	Principes de la Propagation . . . . .	111
7.3.1	Effet Doppler et canaux à évanouissement . . . . .	112

7.3.2	Path loss . . . . .	114
7.3.3	Effet “Shadowing” . . . . .	116
7.3.4	Expression Générale . . . . .	116
7.4	Calcul du nombre de ressources d’un CCTRCH . . . . .	116
<b>8</b>	<b>Conclusion</b>	<b>119</b>
<b>9</b>	<b>Publications</b>	<b>121</b>
9.1	Publications concernant cette thèse . . . . .	121
9.2	Présentation de Poster et atelier . . . . .	122
9.3	Autres Travaux . . . . .	122
<b>10</b>	<b>Bibliographie</b>	<b>125</b>

# Liste des Figures

1.1	Architecture générale d'un réseau UMTS . . . . .	2
2.1	Architecture d'un réseau UMTS . . . . .	8
2.2	Classification et conditionnement du trafic dans Diffserv . . .	10
2.3	Architecture de la QoS dans l'UMTS . . . . .	12
2.4	Mécanisme de gestion de la QoS . . . . .	13
2.5	Quality of Service Information Element . . . . .	15
2.6	Architecture Platon : Plan Utilisateur et Plan Contrôle . . .	18
2.7	Différence entre l'Architecture Platon et l'Architecture 3GPP	19
3.1	Architecture de l'UTRAN . . . . .	28
3.2	Communication entre deux entités RLC . . . . .	32
3.3	Communication entre deux entités RLC, perte d'un SDU . .	34
3.4	Codes Convolutifs utilisés dans l'UMTS . . . . .	37
3.5	Turbo Code utilisé dans l'UMTS . . . . .	38
3.6	SDU Error Rate pour des PDU de 80 bits avec codes correcteurs.	41
3.7	SDU Error Rate pour des PDU de 80 bits sans codes correcteurs.	42
3.8	SDU Error Rate pour des PDU de 160 bits avec codes cor- recteurs. . . . .	43
3.9	SDU Error Rate pour des PDU de 160 bits sans codes cor- recteurs. . . . .	44
3.10	SDU Error Rate pour des PDU de 320 bits avec codes cor- recteurs. . . . .	45
3.11	SDU Error Rate pour des PDU de 320 bits sans codes cor- recteurs. . . . .	46
3.12	Délais moyens de transfert, PDU = 80 bits, maxdat = 0. . . .	47
3.13	Délais moyens de transfert, PDU = 160 bits, maxdat = 0. . .	47
3.14	Délais moyens de transfert, PDU = 80 bits, maxdat = 1. . . .	48
3.15	Délais moyens de transfert, PDU = 160 bits, maxdat = 1. . .	48
3.16	Délais moyens de transfert, PDU = 80 bits, maxdat = 2. . . .	49

3.17	Délais moyens de transfert, PDU = 160 bits, maxdat = 2. . .	49
3.18	Délais moyens de transfert, PDU = 80 bits, maxdat = infini.	50
3.19	Délais moyens de transfert, PDU = 160 bits, maxdat = infini.	50
3.20	Délais pour recevoir 99 % des SDU. . . . .	51
3.21	Délais pour recevoir 90 % des SDU. . . . .	51
3.22	efficacité du paramétrage, PDU de 80 bits. . . . .	53
3.23	efficacité du paramétrage, PDU de 160 bits. . . . .	54
3.24	efficacité du paramétrage, PDU de 320 bits. . . . .	54
3.25	efficacité du paramétrage, PDU de 320 bits. . . . .	55
3.26	efficacité des codes correcteurs en fonction de la taille du PDU.	56
4.1	débit en fonction du code correcteur et de l'état du canal . . .	62
4.2	Graphe des Transitions pour le lien adaptatif dans GPRS . .	63
4.3	Efficacité des taux des RS code dans ATM sur Satellite . . .	64
4.4	Mécanisme adaptatif . . . . .	66
4.5	Code à sélectionner en fonction de l'application et de l'état du canal . . . . .	68
4.6	Code à sélectionner en fonction de l'application et de l'état du canal . . . . .	72
5.1	Structure d'une Trame Radio . . . . .	75
5.2	Etalement et désétalement en CDMA . . . . .	76
5.3	Désétalement avec un OVSF qui ne coresspond pas. . . . .	76
5.4	Arbre des OVSF codes. . . . .	77
5.5	Ressource en UMTS TDD, SF = 16 . . . . .	80
5.6	Max in slot . . . . .	84
5.7	Min in slot 1 . . . . .	84
5.8	Power Based . . . . .	85
5.9	Les ressources allouées par les différents algorithmes. . . . .	86
5.10	puissance moyenne par slot, charge forte . . . . .	87
5.11	puissance moyenne par slot, charge moyenne . . . . .	88
5.12	puissance moyenne par slot, charge faible . . . . .	88
5.13	cdf charge forte . . . . .	89
5.14	Echec du Max in slot . . . . .	90
6.1	Configuration des ressources . . . . .	98
6.2	Modèle du canal . . . . .	98
6.3	Puissance moyenne par slot en fonction des algorithmes DCH	99
6.4	$\beta = 0$ . . . . .	101
6.5	$\beta = 10\ 000$ . . . . .	102

---

6.6	$\beta = 100\ 000$ . . . . .	102
6.7	Quantité de blocs cumulé d'un utilisateur en fonction de $\beta$ . . .	103
6.8	Evolution des quantités servies pour 2 services de meme pri- orités au cours du temps. . . . .	103
6.9	Allocation de RT sur DCH et DSCH . . . . .	106
7.1	Utilisation et intégration de la plate-forme . . . . .	110
7.2	Le serveur de canal . . . . .	111
7.3	La chaîne Radio Fréquence . . . . .	111
7.4	Propagation directe avec un terminal mobile. . . . .	112
7.5	Deux chemins de propagation. . . . .	113



# Liste des Tableaux

2.1	Les classes de trafic 3 GPP . . . . .	17
2.2	Proposition de transcription de classe de trafic Diffserv en classe de trafic UMTS . . . . .	21
2.3	Description des services IP . . . . .	22
2.4	QoS IE des services IP 1 . . . . .	23
2.5	QoS IE des services IP 2 . . . . .	25
3.1	Paramètres variant dans nos expériences . . . . .	40
3.2	Taille du CRC en fonction du BER requis . . . . .	57
3.3	Transport Format Set . . . . .	59
4.1	Codes correcteurs utilisés dans GPRS . . . . .	62
4.2	Débit moyen dans GPRS . . . . .	63
4.3	Mode de la couche physique d'Hiperlan 2 . . . . .	64
4.4	Exemple de lien adaptatifs dans les réseaux mobiles . . . . .	65
4.5	Exemples de critères de QoS . . . . .	65
4.6	Caractéristiques des scénarios. . . . .	67
4.7	Gain du codage adaptatif pour une application Non Real Time	69
4.8	Gain du codage adaptatif pour une application Real Time Interactive . . . . .	70
4.9	Gain du codage adaptatif pour une application Real Time Non Interactive . . . . .	71
5.1	Structure des bursts . . . . .	78
5.2	durée moyenne des interarrivées et des services . . . . .	86
5.3	Taux de rejet en fonction de la charge et de l'algorithme . . . . .	89
6.1	Taux de rejet des applications RT et charges des DSCH . . . . .	100
6.2	Block Error Rate . . . . .	100
6.3	Proportion de ressources allouées par service . . . . .	104

6.4	Taux de rejet des services RT et taux d'utilisation des DSCH pour les services NRT, seuil de 10 % de rejet pour le trafic RT 106
6.5	Taux de rejet des services RT et taux d'utilisation des DSCH pour les services NRT, les RT utilise les DSCH librement. . . 107
7.1	Valeurs typiques de l'exposant $\beta$ du path loss . . . . . 115

# Acronymes

AF	Assured Forwarding
AM	Acknowledged Mode
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BCH	Broadcast Channel
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CCTRCH	Coded Composite Transport Channel
CDMA	Code-Division Multiple Access
CN	Core Network
CRC	Cyclic Redundancy CheckSum
DCH	Dedicated Channel
DSCH	Downlink Shared Channel
DL	Downlink
EDF	Earliest Deadline First
EF	Expedited Forwarding
FDD	Frequency Division Duplex
FACH	Forward Access Channel
FER	Frame Erasure Ratio
FTP	File Transfer Protocol
GBR	Guaranteed Bit Rate
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
HARQ	Hybride Automatic Repeat Request
HPF	Highest Priority First
IETF	Internet Engineering Task Force

---

IP	Internet Protocol
LLF	Least Laxity First
MAC	Medium Access Control
MBR	Maximum Bit Rate
MT	Mobile Terminal
OVSF	Orthogonal Variable Spreading Factor
PCH	Paging Channel
PDP	Packet Data Protocol
PDU	Protocol Data Unit
QPSK	Quaternary Phase Shift Keying
RAB	Radio Access Bearer
RACH	Random Access Channel
RAN	Radio Access Network
RG	Radio Gateway
RLC	Radio Link Control
RNC	Radio Network Controller
RNRT	Réseau National de Recherche en Télécommunications
RR	Round Robin
RRC	Radio Resource Control
RRM	Radio Resource Management
RSVP	Resource Reservation Protocol
RT	Real Time
SDU	Service Data Unit
SGSN	Serving GPRS Support Node
SNR	Signal-to-Noise Ratio
TDD	Time Division Duplex
TM	Transparent Mode
TPC	Transmission Power Control
TRCH	Transport Channel
TF	Transport Format
TFC	Transport Format Combination
TFCI	Transport Format Combination Indicator
TFS	Transport Format Set
TTI	Transmission Time Interval
UE	User Equipment
UL	Uplink
UM	Unacknowledged Mode
UMTS	Universal Mobile Telecommunication System
USCH	Uplink Shared Channel
UTRA	UMTS Terrestrial Radio Access

UTRAN	UMTS Terrestrial Radio Access Network
WCDMA	Wideband Code Division Multiple Access
WRR	Weighed Round Robin



# Chapitre 1

---

## Introduction

---

Les réseaux UMTS (Universal Mobile Telecommunication System) sont des réseaux mobiles dits de troisième génération. Les réseaux mobiles de première génération, pendant les années 1970-1980, étaient des réseaux analogiques (AMPS aux USA, Radiocom 2000 en France) [1]. Puis, dans les années 1990 apparurent les réseaux mobiles de seconde génération, qui étaient des réseaux numériques (GSM en Europe) [2]. Ces deux types de réseaux transportaient essentiellement de la conversation téléphonique. Les réseaux de troisième génération commencent à transporter des applications très diverses : la conversation téléphonique bien sur, mais également des applications vidéo, ou des fichiers variés allant de la consultation de pages Web à des applications bancaires. Ces applications doivent, pour convenir aux utilisateurs, avoir un certain niveau de qualité. On parle de Qualité de Service (QoS). La qualité perçue par un utilisateur doit être indépendante du protocole utilisé pour transporter ces applications. La QoS doit être transcrite en critères utilisables par le réseau. Cette transcription dépend de l'application elle-même, mais également de la nature du réseau.

Le thème de cette thèse est de proposer d'abord un moyen de transporter des applications multimédias dans le réseau UMTS en respectant les critères de QoS de ces applications.

Le réseau UMTS se décompose en deux parties :  
d'une part, le Réseau d'Accès Radio (UTRAN, UMTS Radio Access Net-

work) qui gère l'accès au lien radio, et, d'autre part, le Core Network, réseau filaire. La figure 1.1 montre l'architecture générale d'un réseau UMTS.

Le comportement et la capacité de ces deux parties du réseau sont très

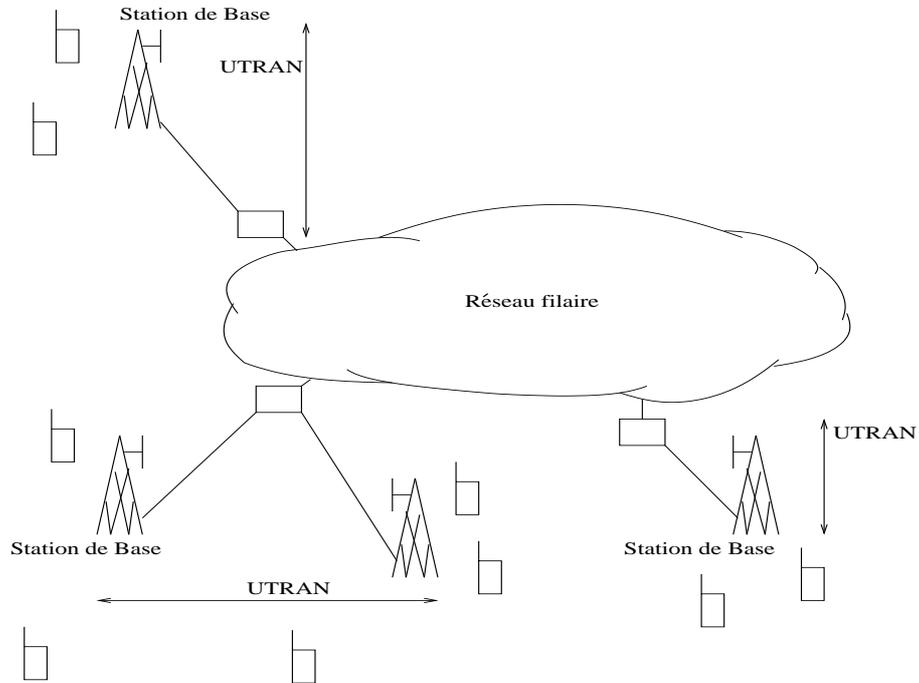


Figure 1.1: Architecture générale d'un réseau UMTS

différents.

Nous avons utilisé pour cette thèse la plate-forme expérimentale d'Eurécom, développée dans le cadre du projet RNRT PLATON [3]. Le code source de cette plate-forme a été intégré dans un simulateur auquel était adjoint un simulateur du canal radio. Les couches de protocoles radios RLC, MAC et Physique (détaillé aux chapitres 3 et 5) sont conformes à la Release 5 des spécifications de 3 GPP. La couche physique en particulier, utilise la technique d'accès radio TDD (Time Division Duplex). Cette thèse se place donc dans le cadre de l'UMTS-TDD. La couche réseau, quant à elle, interface avec IPv6 [4] et le réseau filaire.

Dans la partie filaire, sous IPv6, la bande passante est généralement constante et les données d'un utilisateur sont transportées sous forme de

---

paquets. La probabilité de perdre un paquet est très faible, excepté faite des nœuds du réseau (en raison notamment des problèmes de congestions). Par conséquent, la QoS dépend essentiellement du traitement des priorités des différents flux de paquets sur les nœuds du réseau. Le mécanisme utilisé pour gérer ces priorités s'appelle Diffserv.

Dans le réseau d'accès radio, la bande passante est variable (des interférences, peuvent la réduire) et les données d'un utilisateur sont transportées sous forme de blocs. La probabilité de perdre un paquet est non négligeable. La principale cause de perte est due aux erreurs de transmissions sur la voie radio.

3 GPP, l'organisme de spécification de l'UMTS, propose des outils spécifiques pour la gestion de la QoS dans l'UTRAN. Cependant, 3GPP ne propose pas de méthodes pour traduire la transcription de la QoS du réseau filaire en QoS UTRAN. Dans le chapitre 2 de cette thèse nous proposerons donc une méthode pour réaliser cette traduction et nous donnerons des exemples de traduction de QoS.

Ensuite, dans le chapitre 3, nous proposerons un paramétrage de l'UTRAN pour transporter ces applications. Le paramétrage devra permettre le respect des critères de Qualité et il se fera en fonction de la QoS signalée. Une étude du comportement de l'UTRAN en fonction de l'état du canal et du paramétrage sélectionnée y sera réalisée.

Le comportement et l'efficacité d'un paramétrage sont très dépendants de l'état du canal radio, qui est variable dans le temps. Pour améliorer le paramétrage fait au chapitre précédent, nous proposerons dans le chapitre 4 un paramétrage adaptatif en fonction de l'état du canal radio. Ce paramétrage adaptatif est donc une amélioration de celui réalisé au chapitre 3, car il en accroît l'efficacité .

Le comportement du canal dépend également du nombre d'utilisateurs simultanément présents. Un trop grand nombre d'utilisateurs peut énormément dégrader la qualité du canal radio. Pour les applications dites Real Time (RT), des applications à débit garanti et ayant des contraintes fortes en termes de délais, nous proposerons une politique de Contrôle d'Admission (CAC) et d'allocation de ressources. Le contrôle d'admission permettra qu'un nouvel utilisateur ne perturbe pas les applications en cours, et assurera ainsi la QoS. L'allocation de ressources permettra de réserver à un utilisateur des ressources pour la durée de son application. Nous comparerons, dans le chapitre 5, différents algorithmes d'allocation, et proposerons un algorithme qui permettra d'avoir le maximum d'utilisateurs présents simultanément.

Ensuite, dans le chapitre 6, nous chercherons à transporter des appli-

cations dites Non Real Time (NRT) en même temps que des applications Real Time. Les applications Non Real Time ont un débit non garanti et n'ont pas de contraintes en termes de délais. Elles peuvent par contre avoir différents niveaux de priorité. Etant à débit variables, les applications NRT n'auront pas de ressources qui leurs seront réservées. Elles partageront donc des ressources avec les applications d'autres usagers. L'accès à ces ressources leur sera indiqué par un algorithme d'ordonnancement. Cet algorithme aura plusieurs caractéristiques : il tiendra compte de la priorité des différentes applications, il tiendra compte de l'état du canal radio, il sera équitable, et surtout, il permettra que les applications NRT transportées ne perturbent pas les applications RT en cours. En tenant compte de l'état du canal, cet algorithme réduira le taux de perte des paquets des applications NRT transportées.

Cette thèse aura ainsi couvert la plupart des points nécessaires aux transports des applications multimédia dans l'UTRAN. Les choix qui auront été faits permettront de respecter les critères de QoS des applications transportées et en optimisant l'utilisation des ressources.

## Chapitre 2

---

# Architecture de Qualité de Service pour les applications multimédias

---

### 2.1 Introduction

Nous nous intéressons dans cette thèse au transport de services multimédias dans un réseau UMTS. Or, les services transportés concernent des applications très variées qui auront des besoins très différents. Si une application est transportée dans des conditions correctes, alors un utilisateur en sera satisfait, sinon, il en aura une mauvaise perception. Si la perception d'un utilisateur est un concept assez intuitif, elle peut tout de même être traduite en critères utilisables par le réseau. Ces critères vont définir la Qualité de Service (QoS).

Le réseau UMTS est composé de plusieurs parties : des Stations de Bases (BS), qui gère la transmission sur le lien radio, et une partie filaire qui assure le transport entre les différentes BS. Le comportement de la partie radio et de la partie filaire sont très différents, la façon de gérer et signaler la QoS va également être différente. Il sera donc nécessaire d'avoir un outil à l'intersection entre les deux parties du réseau qui réalise la traduction entre les deux méthodes de signalisation de la QoS.

Dans ce chapitre, nous présenterons successivement :

- les critères de QoS dans un réseau de télécommunication,
  - l'architecture générale d'un réseau UMTS,
  - la façon dont la QoS peut être gérée dans la partie filaire,
  - la façon dont la QoS est gérée dans la partie UMTS suivant les spécifications 3GPP,
  - la façon dont la QoS est gérée dans l'UTRAN dans la plate-forme UMTS PLATON d'Eurécom,
  - notre mécanisme de traduction entre la QoS dans la partie filaire et la QoS dans l'UTRAN,
- enfin nous présenterons quelques exemples de traduction de QoS.

### 2.2 Définition et critères de la Qualité de Service (QoS)

La notion de Qualité de Service (QoS) est un concept qui permet d'exprimer les besoins d'applications multimédias. L'IUT-TS [5] donne cette définition de la QoS : "Effet global produit par les caractéristiques d'un service fourni à un usager qui déterminent le degré de satisfaction que cet usager retire du service". La QoS représente l'ensemble des caractéristiques quantitatives et qualitatives d'un système multimédia nécessaire à l'accomplissement des fonctionnalités requises par une application. Pour assurer le bon déroulement de l'application, il est nécessaire de satisfaire aux besoins en terme de QoS de chacun des médias échangés par l'application.

Quels critères allons nous retenir pour assurer la QoS des applications que nous allons transporter ? Les standards 3GPP [6] proposent de retenir les critères suivants : les délais de transmission, la gigue et les taux de pertes [17].

#### 2.2.1 Les Délais

Le délai d'un service est le temps nécessaire à un paquet pour traverser le réseau de la source de ce paquet à la destination. Les causes du délai sont notamment le temps du traitement du paquet, les temps de transmission, de propagation et d'attente dans le réseau.

#### 2.2.2 La Gigue (jitter)

La gigue représente la variation du délai de transfert de deux paquets consécutifs (valeur absolue du délai de transfert d'un paquet en réception moins le délai de transfert du paquet précédemment reçu).

### 2.2.3 Les taux de perte

Le but principal de la télécommunication est de transporter des données. Les critères sur les pertes de données concernent donc toutes les applications transportées. Deux types de pertes seront considérés. Le Bit Error Rate (BER) qui indique la proportion de bits erronés sur une trame, et le BLER (BLock Error Rate) qui mesure le taux de paquets qui ne sont pas reçus par les utilisateurs. Le BER peut être dû aux conditions radios (bruits gaussiens, multi trajets, évanouissements) qui modifient la valeur des bits lors de leurs transmissions. Le BLER peut être dû à des pertes de paquets en raison du manque d'espace en mémoire à une étape du trajet, ou au rejet du paquet par le récepteur si il y detecte une erreur.

Les valeurs de ces critères sont définies pour chaque type d'application. Des exemples en seront donnés dans ce chapitre. La QoS des applications sera respectée si le réseau est capable de respecter ces critères. Nous allons présenter maintenant l'architecture de la QoS dans le réseau UMTS (partie filaire et UTRAN), et montrer comme le réseau va s'efforcer de respecter ces critères.

## 2.3 Architecture générale de l'UMTS.

L'architecture générale d'un réseau UMTS est constituée de quatre parties [7] [24] :

1. l'UTRAN (UMTS Terrestrial Radio Access Network) correspond au réseau d'accès radio;
2. le Réseau Cœur (Core Network) est l'interface responsable de la commutation et du routage des communications entre l'UTRAN et la partie filaire;
3. la partie filaire, parfois appelée réseau externe, transporte les communications entre les différentes Stations de Bases (BS);
4. et enfin le terminal mobile, ou User Equipement(UE), permet à un utilisateur d'accéder aux services du réseau.

La figure 2.1 présente les différents éléments du réseau, ainsi que les différentes interfaces.

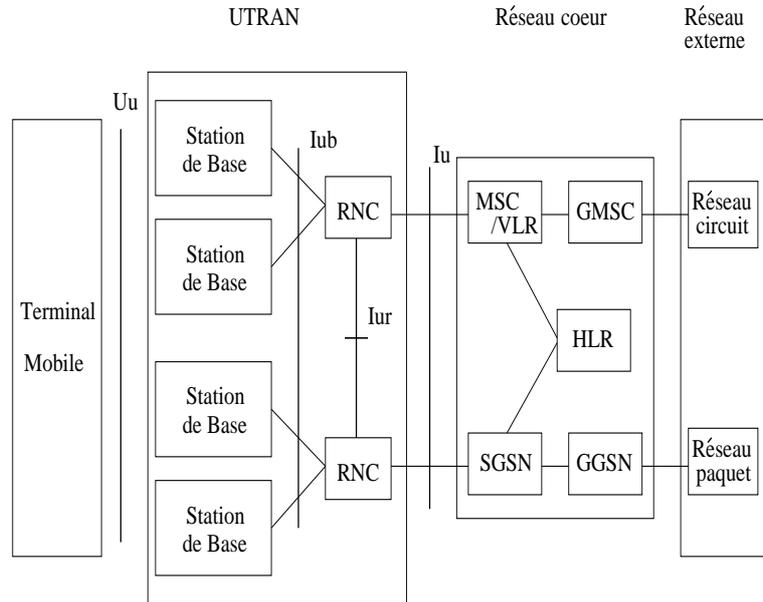


Figure 2.1: Architecture d'un réseau UMTS

L'UTRAN comporte les deux éléments suivants :

- le RNC (Radio Network Controller) est le point d'entrée (et de sortie) de l'UTRAN. Un RNC peut être relié à une ou plusieurs Station de Base (BS), dont il gère les ressources radio;
- la Station de Base, parfois appelée Node B, convertit le flux de données entre le RNC et l'interface air WCDMA et participe à la gestion des ressources radios.

Le réseau cœur (Core Network) comporte notamment les éléments suivants :

- le SGSN (Serving GPRS Support Node) est un nœud GPRS qui est en charge de la gestion des services à commutation de paquets des abonnés attachés au réseau. Un SGSN peut être relié à un ou plusieurs RNC.
- le GGSN (Gateway GPRS Support Node) est un nœud passerelle GPRS. Il route les paquets venant du réseau filaire vers le SGSN du destinataire.

-le HLR (Home Location Register) est la base de données qui gère l'ensemble des abonnés et leurs profils. Le HLR stocke également des données de localisation de l'abonné.

- le MSC/VLR (Mobile Switching Centre/Visitor Location Register) fournit des services circuit aux terminaux des utilisateurs. Le MSC permet la commutation des connexions circuit et le VLR contient une copie du profil de l'abonné et des informations sur sa localisation.

- le GMSC (Gateway MSC) est un commutateur. C'est le point d'entrée et de sortie en mode circuit du réseau UMTS.

Les interfaces entre les différents éléments du réseau sont également spécifiées :

- l'interface Uu. Il s'agit de l'interface air WCDMA. Elle sera détaillée dans le chapitre 5 de cette thèse. Elle relie Le terminal mobile à l'UTRAN;

- l'interface IU. Elle relie l'UTRAN au Core Network;

- l'interface Iub. Elle relie les Stations de Bases au RNC.

- l'interface Iur permet le changement de cellule (softhandover).

La partie filaire peut être sous IP ou sous ATM. Dans cette thèse, nous considérerons que le réseau filaire est sous IPv6.

## 2.4 Diffserv

Différentes architectures existent pour gérer la QoS dans les réseaux filaires. Diffserv (Differentiated Services) est une des plus couramment utilisée sous IPv6.

Le modèle DiffServ consiste à classer le trafic grâce à un code présent dans le paquet IP. On applique ensuite des traitements différenciés aux différentes classes de trafic [8] [9].

DiffServ se base sur le champ Traffic Class d'IPv6 (TOS, Type Of Service dans IPv4) que l'IETF a redéfini en champ DS (DiffServ) pour effectuer la classification [10] [11].

Les opérations de classification, de contrôle et de marquage sont effectuées par les routeurs périphériques (Edge Router) tandis que les routeurs centraux (Core Router) traitent les paquets selon un comportement

spécifique en fonction de la classe codée dans l'en-tête d'IP (champ DS). Ce traitement s'appelle PHB (Per Hop Behavior).

Il existe deux types de nœud dans Diffserv : les nœuds frontières qui effectuent les opérations complexes, et les nœuds intérieurs. Chaque équipement frontière met en oeuvre les mécanismes suivants [12] :

- classification des trafics (Classifier);
- conditionnement des trafics (Conditionner);
- ordonnancement (Scheduling);
- acheminement (Forwarding);

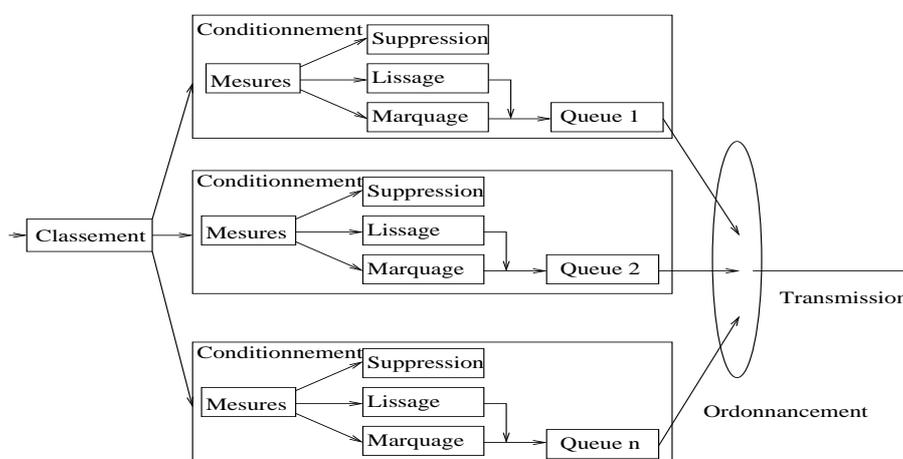


Figure 2.2: Classification et conditionnement du trafic dans Diffserv

Le Classifier sélectionne les paquets en se basant sur une partie de l'en-tête. On distingue deux types de Classifier:

- BA (Behaviour Aggregate): classification établie uniquement en fonction de la valeur du champ DS;
  - MF (Multi-Field): classification établie selon la valeur d'un ou de plusieurs champs de l'en-tête du paquet (champ DS, @ source, @ destination, port...)
- Les règles utilisées par le Classifier n'ont pas été fixées par l'IETF. Cette classification pourra être faite manuellement, l'administrateur du réseau renseigne alors des tables déterminant le mode de marquage des paquets en fonction de différents champs (adresses d'origines, numéro de port de destination ) [13].

Le Conditionner met en oeuvre 4 composants :

- meter : mesure du trafic pour vérifier s'il est compatible avec le contrat et

fourniture des infos aux autres composants;

- marker: affectation d'un DS qui peut être différent de celui qui est reçu;
- shaper: lissage du trafic en retardant certains paquets de telle sorte qu'il respecte le débit contractuel;
- dropper: suppression de paquets dépassant le trafic contractuel;

### 2.4.1 Les classes de trafic Diffserv

Diffserv propose trois grandes classes de trafic : Expedited Forwarding, Assured Forwarding et Best Effort.

#### Expedited Forwarding

L'Expedited Forwarding [14], également appelé service premium, fournit un service garanti en terme de débits, de taux de pertes et de délais. Ce service requiert au niveau du réseau plusieurs conditions :

- à chaque nœud, le débit sortant doit être supérieur au débit entrant;
- les routeurs doivent être capable d'ordonnancer les paquets;
- le trafic sortant doit être lissé;
- la proportion de l'Expedited Forwarding doit être faible par rapport au reste du trafic.

#### Assured Forwarding

L'Assured Forwarding [15] garantit l'acheminement des paquets. Il peut être divisé en quatre classes de traitements et trois niveaux de priorité. Un routeur doit allouer un minimum de ressources pour chaque classe AF (taille de buffer, bande passante). Une classe AF peut recevoir davantage de ressources quand des ressources sont disponibles. En cas de congestion, le niveau de priorité détermine l'importance du paquet, le dropper éliminant en priorité les paquets de priorités faibles.

#### Best Effort

Le PHP Best Effort ne correspond pas au Best Effort d'IP. En effet dans DiffServ, le Best Effort est un service sans garantie. Les paquets Best Effort prennent la place qui reste dans le réseau. Dans IP, Best Effort est un service équitable où tous les utilisateurs sont traités identiquement.

## 2.5 Architecture de la QoS dans le Core Network et l'UTRAN

L'UMTS utilise le concept de Bearer Service (porteuse de service). Un Bearer Service contient tous les éléments permettant la fourniture de la QoS UMTS requise. La figure 2.3 décrit l'architecture de la QoS. Sur chaque couche, la QoS est exprimé par une série d'attributs qui lui est propre. Ces attributs servent à demander un service approprié aux couches inférieures.

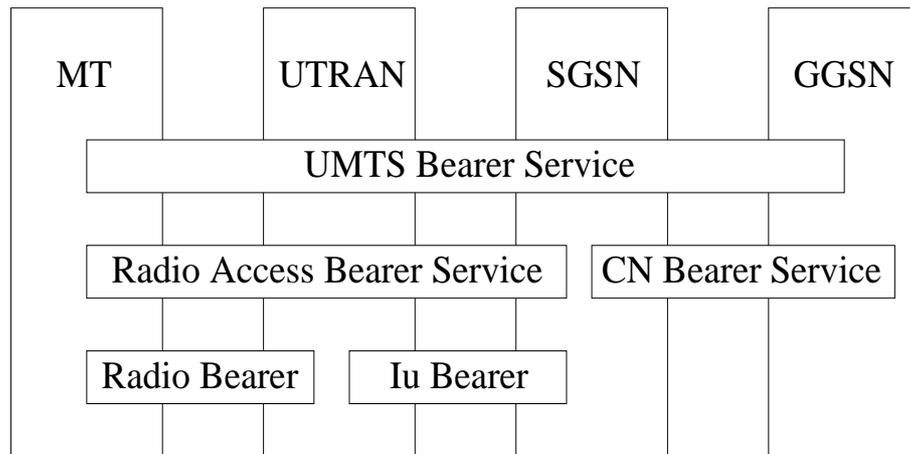


Figure 2.3: Architecture de la QoS dans l'UMTS

L'UMTS Bearer Service a deux parties : le Radio Access Bearer Service (RAB) et le Core Network Bearer Service. Le RAB fournit le transport des données et la signalisation du SGSN du Mobile au SGSN du Core Network. La QoS du RAB est négocié avec l'UMTS Bearer Service. Le RAB est basé sur les caractéristiques de l'interface radio. Il est réalisé par deux sous parties : Le Radio Bearer service et le Iu Bearer Service. Le Radio Bearer service couvre tous les aspects du transport radio. Le Iu Bearer Service fournit le transport entre l'UTRAN et le Core Network. Le Core Network Bearer Service connecte le SGSN avec le GGSN. Le rôle de ce service est de contrôler et d'utiliser le Core Network dans le but de fournir l'UMTS Bearer Service négocié.

Une connexion virtuelle doit être établie entre le mobile et le GGSN. Cette connexion virtuelle s'appelle le PDP context. Une demande d'activation de PDP context comporte notamment les caractéristiques du service en terme de débits et de QoS.

- 1) L'utilisateur envoie au SGSN une demande d'un Activate PDP contexte. Ce message contient notamment les critères de QoS qu'il requiert.
- 2) Le SGSN valide l'Activate PDP contexte à partir de la demande de l'utilisateur et des informations sur les services souscrits par cet utilisateur. Il peut rejeter le PDP contexte, ou modifier les critères de QoS en fonction de ses capacités.
- 3) Le SGSN envoie une demande de création de PDP Context au GGSN.
- 4) Le GGSN crée une nouvelle entité PDP contexte. Il autorise le routage des paquets entre le SGSN et le réseau filaire. Le GGSN signale la création du PDP contexte au SGSN, en indiquant notamment la QoS négociée.
- 5) Le SGSN signale au Mobile l'Activation du PDP contexte. Le SGSN peut router les PDU entre le GGSN et le mobile.

**Vue générale du mécanisme de la QoS**

La figure 2.4 décrit les principales étapes de la gestion de la QoS dans le Plan Contrôle. Les étapes sont les suivantes :

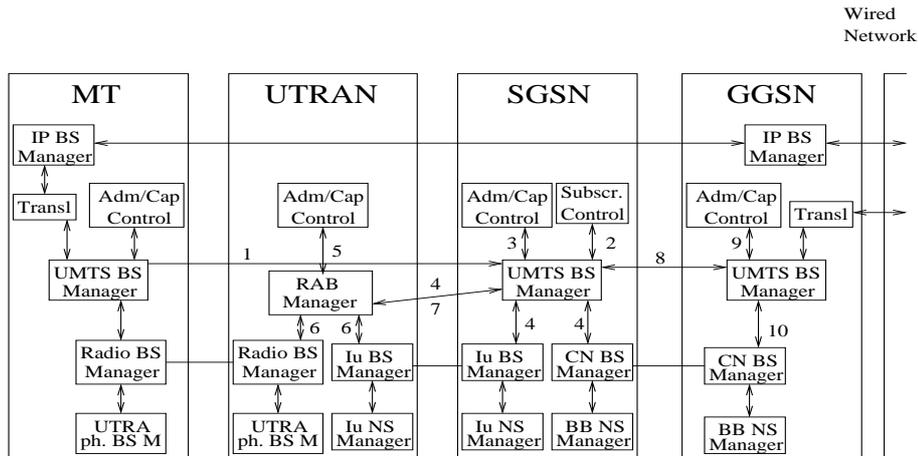


Figure 2.4: Mécanisme de gestion de la QoS

- 1) L'UMTS Bearer Service manager du mobile indique le profil de QoS de son utilisateur à l'UMTS BS manager du SGSN. Pour cela, il lance un PDP context request.
- 2) L'UMTS BS manager du SGSN vérifie les droits de l'utilisateur (recensement des services souscrits par l'utilisateur)

3) Le gestionnaire de l'UMTS BS du SGSN réalise un contrôle d'admission (La BS est-elle en mesure de fournir au service les ressources dont il a besoin ?).

4) L'UMTS BS manager du SGSN traduit les attributs de l'UMTS bearer service en attributs RAB service, en attributs Iu bearear service , et en attributs Core Network bearer service. Il demande ensuite à son Iu BS manager, à son CN BS manager et à son RAB manager dans l'UTRAN de fournir le service demandé.

5) Le RAB manager vérifie avec son Contrôle d'admission et son contrôle de capacité si l'UTRAN a assez de ressources et si il est capable de fournir le service.

6) Le RAB manager traduit les attributs RAB service en Radio Bearer service et en attributs Iu Bearer service . De plus, il demande au Radio BS manager et à l'Iu BS manager de fournir un service Bearer avec les attributs requis.

7) Quand le Radio Access Bearer est établi, le RAB manager informe l'UMTS BS manager du SGSN.

8) L'UMTS BS manager du SGSN communique avec l'UMTS BS manager du GGSN pour demander les critères de QoS de l'utilisateur. Il crée un PDP context.

9) L'UMTS BS manager du GGSN effectue un contrôle d'admission et vérifie si il y a assez de ressources pour transmettre la nouvelle application.

10) L'UMTS BS manager du GGSN traduit les attributs UMTS bearer service en attributs Core Network bearer service. Il demande à son Core Network BS manager de fournir le service.

Nous allons maintenant décrire le QoS Information Element (QoS IE). Cette structure donnera au Radio Resource Control les informations pour configurer les services dans l'UTRAN.

### **2.5.1 Quality of Service Information Element**

Dans l'UTRAN, la QoS est signalé dans le Quality of Service Information Element (QoS IE) [16] [6]. Le QoS IE contient 13 octets. Les champs en sont indiqués dans la figure 2.5.

Octet 1 et 2 : Les deux premiers octets servent à reconnaître le QoS IE.

Octet 3 : Les 3 premiers bits (de droite à gauche) du troisième octet indiquent la reliability class. Cela définit notamment le type de retransmission que devra utiliser l'UTRAN. Les bits 4 à 6 indiquent la delay class. Quatre classes de délais sont possibles.

Quality of Service IEI		
Length of Quality of Service IEI		
spare	delay class	reliability class
peak throughput		spare
precedence class		spare
spare		mean throughput
traffic class	delivery order	delivery of erroneous SDU
Maximum SDU size		
Maximum bit rate for uplink		
maximum bit rate for downlink		
Residual BER		SDU Error Ratio
Transfer delay		Traffic Handling priority
Guaranteed bit rate for uplink		
Guaranteed bit rate for downlink		

Figure 2.5: Quality of Service Information Element

Octet 4 Bits 1 à 3. Precedence class. Indique le niveau de la priorité.  
 Bits 5 à 8. Peak throughput. Indique le débit crête.

Octet 5 Bits 1 à 5. Mean throughput. Indique le débit moyen.

Octet 6 Bits 1 à 3. Delivery of erroneous SDU. Indique si les SDU où des erreurs ont été détectées doivent être délivrés à la couche supérieure ou être détruits.

Bits 4 et 5. Delivery order. Indique si les SDU doivent être délivrés dans l'ordre.

Bits 6 à 8. Traffic class. Indique si le flux est de type conversational, background, streaming ou interactive.

Octet 7. Maximum SDU size. Indique la taille maximum du SDU.

Octet 8 et 9 Maximum bit rate for uplink, Maximum bit rate for downlink. Indique respectivement le débit maximum en uplink et downlink.

Octet 10 Bits 1 à 4. SDU Error Ratio. Indique le taux de paquets qui peut être perdu.

Bits 5 à 8. Residual Bit Error Rate. Indique le taux d'erreur maximum par paquet. Ce champ sert à déterminer la taille de la CRC.

Octet 11

Bits 1 et 2. Traffic handling priority. Indique la priorité du trafic  
Bits 3 à 8. Transfer delay. 95% des SDU d'une application doivent être délivrés dans un délai inférieur à la valeur donnée dans le QoS IE. Ces délais ne concernent que le transfert sur le Radio Access Network.

Octet 12 et 13. Guaranteed bit rate for uplink, Guaranteed bit rate for downlink Indique respectivement le débit garanti en uplink et downlink.

## 2.6 Classification des applications multimédias

Le champ Traffic Class de l'octet 6 du QoS IE permet de regrouper les applications qui ont des besoins proches et doivent donc avoir des traitements proches. Les autres champs serviront à préciser plus finement ces besoins et permettront un traitement plus spécifique des différentes applications. Un traitement adapté des applications a deux objectifs : fournir le service bien sur, mais sans lui allouer trop de ressources. En surdimensionnant le service d'une application, la QoS est certes atteintes de façon sûre, mais au détriment du partage des ressources pour la fourniture d'autres applications.

3 GPP [6] a proposé de regrouper les applications en 4 classes de service, en fonction des principaux critères de qualité requis par ces applications.

Ces 4 classes sont : conversational, streaming, interactive et background.

### 2.6.1 Classe conversational

La classe conversational regroupe des services bidirectionnels (les deux parties échangent des quantités de données similaire), qui ne supportent que des temps de transfert très faibles et constants et qui demandent des débits garantis. Elle préserve l'ordre des paquets d'un même flot.

### 2.6.2 Classe Streaming

La classe conversational regroupe des services monodirectionnels. Ces services demandent des temps de transfert faibles, constants ainsi que des débits garantis.

### 2.6.3 Classe Interactive

La classe Interactive correspond à du trafic bidirectionnel mais potentiellement asymétrique et à débit très variables. Ce trafic a des contraintes sur les délais. Ces contraintes sont toutefois beaucoup moins importantes que dans le case des classes Conversational et Streaming.

### 2.6.4 Classe Background

La classe Background correspond à du trafic sans contrainte de délais, asymétrique, à débit variable et nécessitant des taux d'erreurs très faibles.

La table 2.1 donne des exemples d'application en fonction de la classe de service UMTS.

Classe de trafic	Contraintes	Exemples d'application
Conversational	temps de transfert très court et constant	voix, vidéotéléphonie, jeux en réseaux
Streaming	temps de transfert court	vidéo à la demande
Interactive	peut de contraintes de temps, intégrité des données	internet
Background	pas de contraintes de temps, intégrité des données	courriel, ftp

Table 2.1: Les classes de trafic 3 GPP

Comme pour le champ Traffic Class et les politiques qui y correspondent, la façon de définir le QoS IE et de paramétrer l'UTRAN sont des choix de l'opérateur. Nous allons dans cette thèse définir une méthode de transcription. Cette méthode a été réalisée dans le cadre de la plate-forme expérimentale UMTS Platon. Avant de détailler cette méthode, nous décrirons l'architecture de cette plate-forme et montrer en quoi elle se distingue des spécifications 3 GPP.

## 2.7 Architecture de la QoS dans la plate-forme PLATON

La plate-forme UMTS PLATON a été mise au point dans l'optique d'un monde tout IP. Elle présuppose donc que le réseau filaire est obligatoirement sous IPv6. Cette hypothèse permet de simplifier la transition entre le réseau filaire (sous IPv6) et l'UTRAN.

Les fonctionnalités d'IPv6 pour gérer la mobilité, la QoS et le Paging remplacent donc celles utilisées dans le SGSN et le GGSN. Une couche

d'adaptation est introduite : la Generic Radio Access Adaptation Layer (GRAAL). Cette couche modifie les fonctions du Radio Resource Control (RRC) et du Radio Resource Management (RRM) (figure 2.6). La Station de Base est également modifiée, car elle devient un Radio Gateway avec des fonctions de routages IP.

La figure 2.7 montre l'évolution entre les deux protocoles. En utilisant IP sur la Radio Gateway, permet de supprimer le RNC, le GGSN et le SGSN, et de supprimer plusieurs bearer services.

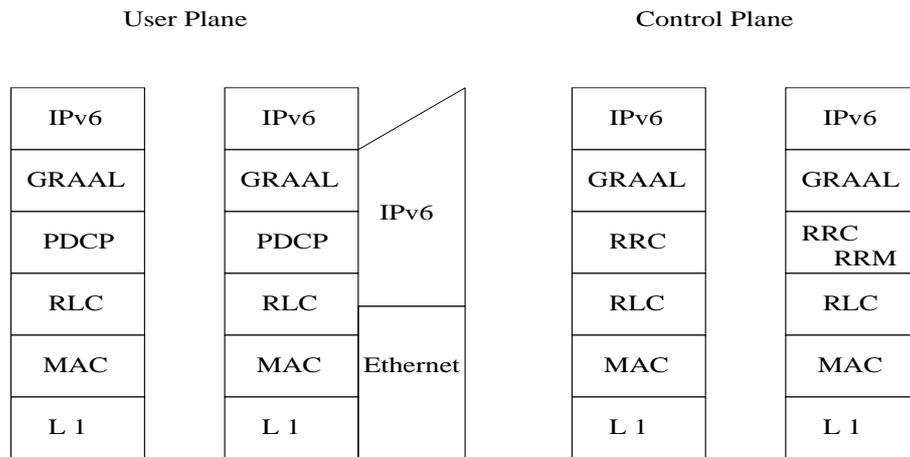


Figure 2.6: Architecture Platón : Plan Utilisateur et Plan Contrôle

Lorsqu'un paquet IPv6 arrive au GRAAL, celui-ci en contrôle l'entête. Les paquets passent à travers un classifieur qui sépare la signalisation des données. La signalisation est traitée et transportée par une interface radio spécifique (Plan contrôle). Si un RAB a déjà été créé pour cet utilisateur et ce type de service alors les données sont transmises à l'entité RLC (Radio Link Control) qui lui correspond (cette étape sera décrite au prochain chapitre). Sinon, le plan contrôle du GRAAL communique avec le RRC par un Service Access Point. Le Service Access Point fournit des services pour l'établissement, et le relâchement d'une connexion. En fonction de la QoS décrite dans le champ Differentiated Service (DS), le GRAAL crée un QoS Information Element (QoS IE) pour ce service. Quand le GRAAL reçoit la Connection Indication, il vérifie si assez de ressources sont disponibles (information donnée par le RRM) et accepte ou refuse la connexion.

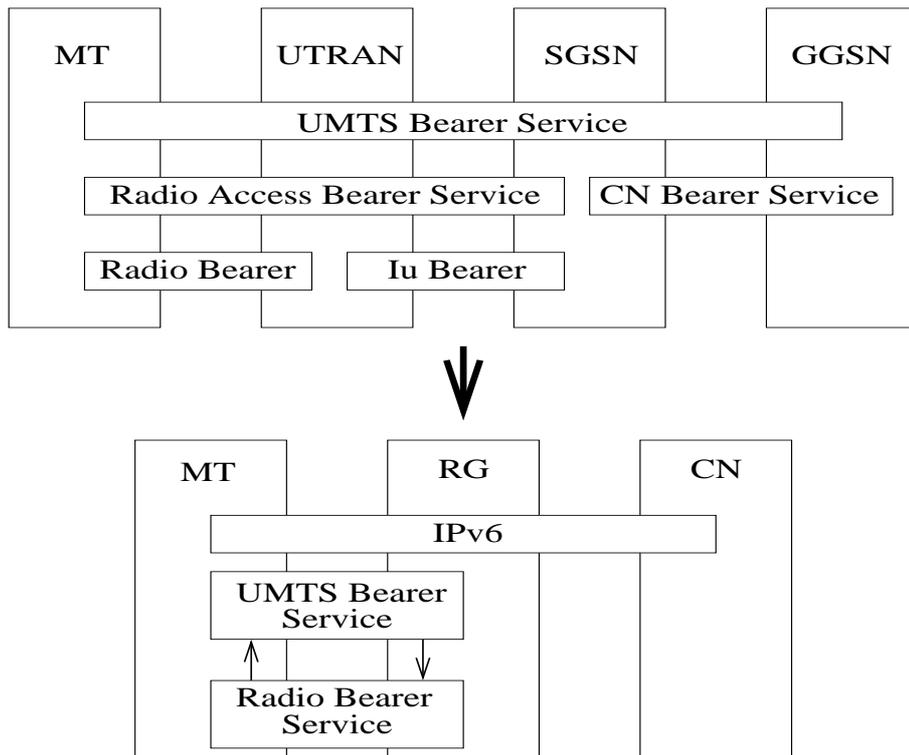


Figure 2.7: Différence entre l'Architecture Platon et l'Architecture 3GPP

Pour réaliser la traduction de la QoS, le GRAAL doit disposer de table de correspondances. Le paragraphe suivant va montrer comment nous avons réalisé ces tables.

## 2.8 Traduction de la QoS DiffServ en QoS UMTS

Comment remplir le QoS IE pour ces applications ? Pouvons nous définir des règles générales pour tous les champs ? Devrons nous en spécifier certains de façons particulières ?

Considérons dans un premier temps les champs Traffic Class. Il y a trois classes de trafic dans Diffserv et quatre dans le QoS IE. A quelle classe UMTS peut on faire correspondre les classes Diffserv ?

La classe EF demande des débits garantis, des délais très courts et un taux de perte des paquets très faible. La classe conversational est sensé fournir un service avec des temps de transfert très courts et constant, un débits garantie, mais est tolérante aux pertes. Les autres services UMTS ne sont pas sensés fournir des délais aussi courts. Nous nous rendons compte que la correspondance entre ces deux classes n'est pas parfaite, mais qu'elle recouvre l'un des besoins fondamentaux qu'est celui des délais. Pour vérifier si une telle correspondance est valide, il est nécessaire de regarder au plus près les applications transportées. Les applications auxquelles on associe généralement EF sont des applications de types conversations téléphoniques. Ces applications sont tolérantes aux erreurs, auquel cas la classe conversational de l'UMTS est adaptée.

La classe AF garantit un débit minimum, et l'acheminement de tous les paquets. Il peut y avoir cependant des délais importants si pendant un laps de temps assez long le débit réel est très supérieur au débit garanti et que le réseau n'est apte qu'à fournir le débit garanti. La classe Streaming demande des délais courts, et peut garantir un débit. Ces deux classes correspondent assez bien. Comme AF propose plusieurs niveaux de services, il sera possible de différencier les différents services Streaming par les autres champs du QoS IE.

La classe Best Effort est un service sans garantie dans Diffserv. La classe Background a les mêmes caractéristiques. Là encore, la correspondance semble adaptée.

Nos choix sont similaires ceux de [18], [19], [20]. [21] a choisit une méthode similaire pour maintenir la QoS entre les deux parties du réseau, mais en faisant des correspondances différentes entre les classes.

La table 2.2 indique la transcription que nous avons proposé entre les

Classes de Trafic Diffserv et les Classes de Services UMTS. Notons que nous n'avons pas utilisé la classe Interactive de l'UMTS. Certaines applications Internet y correspondent pourtant. Elles devront utiliser des classes Diffserv, qui ne seront peut être pas les plus adaptées pour elles (du AF ou du Best Effort par exemple) et devront être transcrites en Interactive Class. La transcription proposé table 2.2 est donc donnée a titre indicative, et montre ainsi la limite d'une transcription automatique.

Classe de trafic Diffserv	Classe de Trafic UMTS
Expedited Forwarding	Classe Conversational
Assured Forwarding	Classe Streaming
Best Effort	Background

Table 2.2: Proposition de transcription de classe de trafic Diffserv en classe de trafic UMTS

Nos choix sont similaires à ceux de [18], [19], [20]. [21] et [22] ont choisi une méthode similaire pour maintenir la QoS entre les deux parties du réseau, mais en faisant des correspondances différentes entre les classes.

Comment remplir les autres champs du QoS IE ? Les Guaranteed bit rate doivent correspondre aux débits garantis par Diffserv. Nous faisons donc ici une transcription automatique des débits de l'unité de l'un à l'unité de l'autre. Les Maximum bit rate du QoS IE doivent se définir à partir des ressources disponibles. La valeur de ce champ se négocie donc avec le RRM à l'initialisation de la connexion. La precedence class et le Traffic Handling priority peuvent se remplir à partir de la classe de trafic Diffserv. On peut donner à l'EF la plus grande priorité, puis aux différentes applications AF en fonction des sous priorités définies au sein d'AF, et enfin les applications Best Effort reçoivent la priorité la plus faible.

Que dire des autres champs ? Les délais, le BER et le SDU ER dépendent grandement de l'application transportée. Le BER est une notion peu importante dans Diffserv, car le BER est particulièrement faible dans les réseaux filaires. Il est automatiquement respecté, même pour les applications les plus contraintes. La transcription ne peut donc être automatique. Au moment d'établir les tables dans les nuds frontières de Diffserv, il faut aussi établir les tables de transcription en QoS IE. Il est donc nécessaire d'avoir une connaissance précise des besoins des applications au moment de renseigner ces tables. Ces tables devront être mise à jour lorsque apparaîtront de nouvelles applications.

Nous allons montrer dans le prochain paragraphe des exemples de tran-

scription pour des applications typiques.

## 2.9 Exemples de traduction

Des services différenciés ont été définis pour la partie filaire de la plate-forme. Leurs caractéristiques sont indiquées table 2.3 [3] [23]. Ces applications sont celles qui ont été utilisées dans le cadre du projet Moby Dick.

Service Name	Service Class	Relative Priority	Service Parameters	Typical Usage description	Radio QoS Class
SIG	AF41	2a	Unspecified	Network Usage	R1
S1	EF	1	Peak BW: 32 kbit/s	Real Time Services	R2
S2	AF21	2b	CIR: 256 kbit/s	Priority (urgent) data transfer	R3
S3	AF11	2c	64 kbps	Olympic Service	R4
S3	AF12	2c	128 kbps	Olympic Service	R5
S3	AF13	2c	256 kbps	Olympic Service	R6
S4	BE	3	Peak bit rate 32 kbit/s	Best Effort	R7
S5	BE	3	Peak bit rate 64 kbit/s	Best Effort	R8
S6	BE	3	Peak bit rate 128 kbit/s	Best Effort	R9

Table 2.3: Description des services IP

Pour que la traduction puisse être faite, il faut qu'à un service Diffserv correspondent un service UMTS. Un numéro de classe radio est donc donné à chaque classe Diffserv. Nous devons maintenant renseigner le QoS IE pour chacune de ces applications.

Le service SIG est un service de signalisation sur le Core Network. Cette signalisation spécifique ne concerne pas l'UTRAN. Lorsque des paquets de ce type de service arrivent sur le GRAAL, ils sont donc traités immédiatement sur le Plan Contrôle, mais ne seront pas transportés sur le plan utilisateur.

Le Service S1 correspond à une application de type Real Time. Il est du type Expected Forwarding dans Diffserv. Sa priorité est de 1, très grande. Son débit crête est de 32 kbit/s. Les applications de type conversational décrivent des services qui correspondent typiquement à cette application.

En lui attribuant un débit garanti égal à son débit crête, cette application sera toujours servie immédiatement, ce qui respectera les critères de priorité. Les applications de type Real Time ont des délais bornés de bout en bout. Ces délais ont été fixés à 100 ms sur l'UTRAN. Cela a été fait conformément aux recommandations 3GPP [25]. Les délais étant très courts, nous avons estimé qu'il n'était pas utile de faire des retransmissions. C'est donc un mode sans retransmission de paquets sur la partie radio (RLC UM) que nous avons choisi. Le taux de perte des paquets IP (SDU ER) est fixé à  $10^{-2}$ . Ce type d'application est tolérant aux erreurs, donc les paquets erronés ne sont pas détruits, mais transmis vers la destination. Nous avons fixé le BER (le taux d'erreur résiduel, donc le taux d'erreur non détecté) à  $4 * 10^{-3}$ . C'est le taux d'erreur non détecté maximum que permet 3 GPP. Il se traduira par l'emploi d'un CRC de 8 bits (le plus petit CRC proposé dans la norme). Ce taux d'erreur permettra tout de même une utilisation acceptable de ce service. Le *Guaranteed Bit Rate* a été fixé comme pour le *Maximum Bit Rate* à 32 kbit/s. En le fixant égal au peak rate demandé, nous garantissons un transport rapide de ce service.

	Radio Class	Traffic Class	Reliability class	Peak throughput	Delivery order	Delivery of erroneous SDU	Max SDU size	SDU ER
SIG AF41	R1	NA	NA	NA	NA	NA	NA	NA
S1 EF	R2	Conversational class 1	UM 4	32 kbit/s 3	Yes 1	Yes 2	1500	$10^{-2}$ 1
S2 AF21	R3	Streaming class 3	AM 3	256 kbit/s 6	Yes 1	Yes 2	1500	$10^{-6}$ 6
S3 AF11	R4	Streaming class 3	AM 3	64 kbit/s 4	Yes 1	Yes 2	1500	$10^{-5}$ 5
S3 AF12	R5	Streaming class 3	AM 3	128 kbit/s 5	Yes 1	Yes 2	1500	$10^{-5}$ 5
S3 AF13	R6	Streaming class 3	AM 3	256 kbit/s 6	Yes 1	Yes 2	1500	$10^{-5}$ 5
S4 BE	R7	Background class 4	AM 3	32 kbit/s 3	Yes 1	No 2	1500	$10^{-6}$ 6
S5 BE	R8	Background class 4	AM 3	64 kbit/s 3	Yes 1	No 2	1500	$10^{-6}$ 6
S6 BE	R9	Background class 4	AM 3	256 kbit/s 3	Yes 1	No 2	1500	$10^{-6}$ 6

Table 2.4: QoS IE des services IP 1

Le Service S2 correspond à des messages d'urgence (services de santé, de police). Il est du type Assured Forwarding dans Diffserv. Sa priorité est assez haute. Son débit peut aller jusqu'à 256 kbit/s. Les applications du type AF demandent une garantie sur l'arrivée des paquets. Nous utiliserons un protocole de retransmission pour réduire la perte des paquets, le RLC sera donc en mode acquitté (AM). Les paquets erronés seront transmis aux destinataires. Mais le taux de perte des paquets IP (SDU ER) devra être de  $10^{-6}$ . Le BER nécessaire à ce type d'application est de  $10^{-5}$ , ce qui correspond à un CRC de 16 bits. Les délais de transfert sur l'UTRAN pour ce type d'application sont de 250 ms. Le MBR sera de 256 kbit/s, égal au peak rate demandé, mais le GBR sera à négocier en fonction des autres utilisateurs présents.

Les Services S3 sont de type Assured Forwarding dans Diffserv. Leur priorité est moyenne. Leurs débits crêtes vont de 64 kbit/s à 256 kbit/s. Les applications du type AF demandent une garantie sur l'arrivée des paquets. Nous utiliserons un protocole de retransmission pour réduire la perte des paquets, le RLC sera donc en mode acquitté (AM). Les paquets erronés seront transmis aux destinataires. Le taux de perte des paquets IP (SDU ER) devra être de  $10^{-5}$ . Le BER nécessaire à ce type d'application est de  $10^{-5}$ , ce qui correspond à un CRC de 16 bits. Les délais de transfert sur l'UTRAN pour ce type d'application sont de 250 ms. Le MBR sera égal au peak rate demandé, mais le GBR sera à négocier en fonction des autres utilisateurs présents.

Les Services S4 à S6 correspondent à du transport de données pour des applications Non Real Time. Il n'y a donc pas de contraintes sur les délais, mais les SDU ER doivent être très faibles. C'est donc le mode RLC AM qui sera sélectionné. Les BER doivent être minimaux, c'est la plus grande taille de CRC qui sera sélectionnée. Ce CRC, de longueur 24 bits, permet un BER de  $6 \times 10^{-8}$ . Les SDU erronés seront détruits. Le MBR sera égal au MBR demandé, mais il n'y aura pas de débit garanti.

Les tables 2.4 et 2.5 contiennent les principaux champs du QoS IE pour les applications étudiées.

### 2.10 Conclusion

Le réseau UMTS transporte des applications multimédias sur des liens filaires et sur des liens radios. Ces applications ont des besoins en termes de

	Radio Class	Traffic Class	Residual BER	Transfer delay	Traffic Handling priority	MBR DL	GBR DL
SIG AF41	R1	NA	NA	NA	NA	NA	NA
S1	R2	Conversational	$4 * 10^{-3}$	100 ms		32 kbit/s	32 kbit/s
EF		class 1	4	10	2	32	32
S2	R3	Streaming	$1 * 10^{-5}$	250 ms		256 kbit/s	TBN
AF21		class 3	7	16	1	88	
S3	R4	Streaming	$1 * 10^{-5}$	250 ms		64 kbit/s	TBN
AF11		class 3	7	16	2	64	
S3	R5	Streaming	$1 * 10^{-5}$	250 ms		128 kbit/s	TBN
AF12		class 3	7	16	2	72	
S3	R6	Streaming	$1 * 10^{-5}$	250 ms		256 kbit/s	TBN
AF13		class 3	7	16	2	88	
S4	R7	Background	$6 * 10^{-8}$			32 kbit/s	
BE		class 4	9	16	3	32	
S5	R8	Background	$6 * 10^{-8}$			64 kbit/s	
BE		class 4	9	16	3	64	
S5	R5	Background	$6 * 10^{-8}$			256 kbit/s	
BE		class 4	9	16	3	88	

Table 2.5: QoS IE des services IP 2

QoS qui s'expriment différemment sur ces deux parties. Nous avons considérés dans ce chapitre un réseau filaire sous IPv6, où la QoS est gérée par Diffserv. Sur la partie radio, une architecture spécifique, la plate-forme UMTS PLATON, a été présentée. Comme pour Diffserv, la gestion de la QoS demande la création et le maintien de tables pour identifier les besoins des applications. Nous avons démontré que la mise en correspondance des tables Diffserv et des tables UTRAN n'était pas automatique. Cette mise en correspondance doit être effectuée à la création de ces tables, en fonction des besoins spécifiques des applications. Une fois les besoins des applications connus, il faut définir une politique pour permettre au réseau de respecter ces critères. Dans le prochain chapitre, nous montrerons comment nous paramétrons l'UTRAN, pour que le réseau UMTS fournisse la QoS aux applications qu'il transporte.



## Chapitre 3

---

# Paramétrage de l'UTRAN

---

### 3.1 Introduction

Nous avons vu au chapitre précédent comment les critères de QoS d'une application étaient indiqués à l'UTRAN. Celui-ci doit maintenant faire respecter ces critères pour les applications qu'ils transportent.

L'UTRAN est constitué de plusieurs couches de protocoles, chacune de ces couches étant caractérisées par de nombreux paramètres. Or le comportement de ces couches varie selon les valeurs fixées pour ces paramètres. Nous devons donc étudier les délais de transports et les taux de pertes dans l'UTRAN en fonctions de la valeur de ces paramètres. Comme les critères de qualité varient d'une application à l'autre, le paramétrage de l'UTRAN et la configuration du service varieront d'une application à l'autre. Lorsque plusieurs paramétrages seront possibles pour une application, le critère de sélection se fera sur l'efficacité des différents paramétrages proposés.

Dans ce chapitre, nous commencerons donc par présenter les différentes couches de l'UTRAN et pour chaque couche identifier les paramètres qui vont influencer sur la fourniture de la QoS. Nous présenterons également les structures utilisées pour indiquer la configuration aux différents éléments du réseau. Puis, en utilisant le code de la plate-forme Platon couplé à un simulateur de canal radio, nous ferons varier la valeur de différents paramètres.

Enfin, nous montrerons pour des applications typiques les configurations que nous avons sélectionnées.

## 3.2 Architecture de l'UTRAN

L'interface radio est divisée en trois couches de protocoles (figure 3.1) [26]: la couche physique (L1), la couche liaison de données (L2), la couche réseau (L3). La couche 2 est divisée selon les sous couches suivantes: Medium Access Control (MAC) et Radio Link Control (RLC).

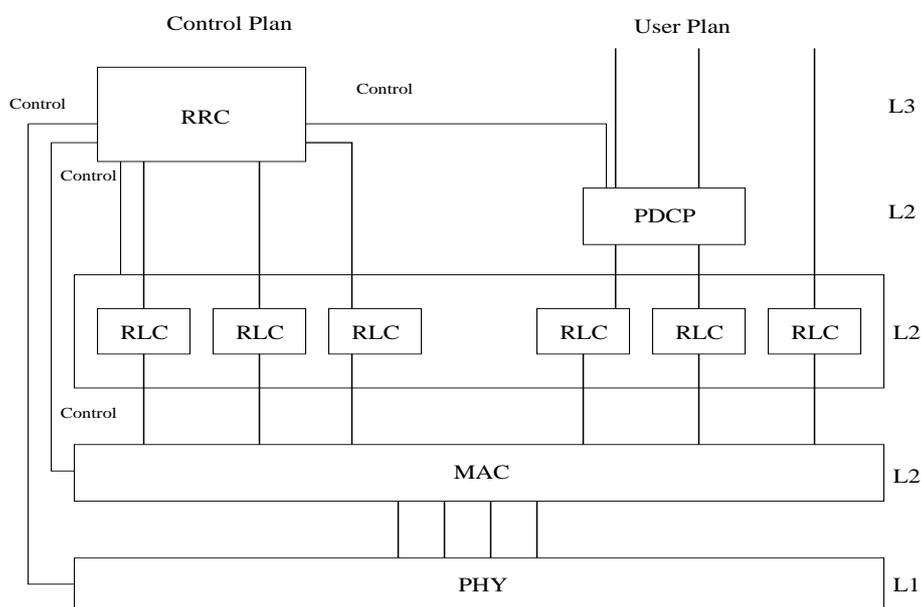


Figure 3.1: Architecture de l'UTRAN

### 3.2.1 Radio Resource Control

Le *Radio Resource Control* (RRC) [28] établit les connexions entre le mobile et l'UTRAN. Il configure les différentes couches, et gère la signalisation. Le RRC gère l'allocation de ressource en fonction des indications données par l'entité *Radio Resource Management*.

### 3.2.2 Packet Data Convergence Protocol

Lorsque l'entité *Packet Data Convergence Protocol* (PDCP) [27] reçoit des SDU de la couche réseau, il en compresse les entêtes et les transmet à une entité RLC. Lorsque le PDCP reçoit des SDU du RLC, il en décompresse les entêtes et les transmet à la couche réseau.

### 3.2.3 Radio Link Control

Une entité (*Radio Link Control* (RLC) [29] est créée par le RRC au début de chaque connexion. Il peut y avoir simultanément plusieurs entités RLC au sein d'un même émetteur. Le RLC est en charge de la segmentation des SDU (Service Data Unit, paquet IP dans l'UMTS) en PDU (Protocol Data Unit). Les PDU sont les unités de données de la couche RLC. Leur taille est fixée par le RRM pour la durée du service. Le temps est divisé en TTI (Transmission Time Interval), un intervalle de temps de durée fixe. En début de TTI, la couche MAC demande au RLC un certain nombre de PDU. Le RLC lui transmet les PDU disponibles. Il y a trois modes RLC: le Mode Transparent (TM), le Mode Non-acquitté (Unacknowledged Mode, UM) et le Mode Acquitté (Acknowledged Mode, AM).

#### RLC Transparent Mode

En Mode Transparent (TM), la couche RLC reçoit les SDU de la couche 3. Les SDU doivent avoir une taille qui est un multiple de la taille d'un PDU. Il n'est pas ajouté d'entête sur les PDU. Pendant un TTI, tous les PDU d'un SDU sont transmis, et seulement les PDU d'un seul SDU. (Exception, si un PDU contient un SDU, alors, on pourra envoyer plusieurs SDU en même temps). Le RLC du récepteur assemble les PDU reçus pendant un TTI et reconstitue le SDU. Le SDU est ensuite envoyé sur la couche réseau. Il n'y a pas de demande de retransmission en Mode Transparent.

#### RLC Unacknowledged Mode

En mode non acquitté, la couche RLC reçoit des SDU de la couche supérieur. Un PDU peut contenir des SDU segmentés ou concaténés. Un PDU peut également contenir des bits de bourrage.

Ces différents cas peuvent se présenter :

- Un PDU peut contenir exactement un SDU.
- Un nombre entier de PDU peuvent contenir un SDU.
- Un SDU peut commencer au début d'un PDU et finir avant la fin de ce

PDU. Il pourra alors être placé le début d'un autre SDU. Ce nouveau SDU pourra se terminer sur ce PDU, ou sur un autre.

- Un PDU peut contenir plusieurs SDU.
- Si il n'y a plus de SDU en attente, on pourra ajouter des bits de bourrage pour compléter un PDU.

Un entête est présent sur chaque PDU. L'un des champs de cet entête est le Length Indicator. Il est utilisé pour connaître les limites des SDU dans un PDU. L'entête contient également un numéro de séquence qui permet au RLC du récepteur de reconstituer les SDU. Le RLC peut temporairement stocker des PDU, mais le RLC UM ne fait pas de demande de retransmission en cas de PDU erroné ou perdu.

### **RLC Acknowledged Mode**

Le RLC AM utilise un ARQ inspiré du Selective Repeat. Le RLC Récepteur acquitte les PDU reçus et demande la retransmission des PDU manquants. En Mode Acquitté, il y a deux types de PDU, ceux de données et ceux de contrôle. Lorsque le RLC reçoit des SDU de la couche supérieure, il les segmente ou les concatène en PDU de taille fixe. La taille du PDU est donnée au RLC par le RRC. Un PDU peut contenir des bits de bourrage. Puis un en-tête est ajouté. Puis, le RLC transmet les PDU à la couche MAC. Ils sont transmis ou retransmis en fonction des acquittements ou des demandes de retransmission du récepteur. Le nombre de retransmissions d'un PDU est limité. Si cette limite est atteinte, son SDU et tous les PDU qui lui sont associés sont détruits. Un SDU a un temps limité pour être acquitté. Le temporisateur MRW (Move Receiving Window) est enclenché à son arrivé au RLC. Quand ce temporisateur expire, le SDU et ses PDU sont détruits. Un message est également envoyé au récepteur pour qu'il détruise de son côté ceux qu'il avait reçus. Le message de destruction doit être acquitté dans un délai fixé. S'il n'est pas acquitté à la fin de ce temps, il doit être retransmis. Le nombre de PDU que le RLC peut transmettre est limité par la demande du MAC et la taille de la fenêtre. Celle-ci correspond au nombre maximum de PDU non acquittés que le RLC peut traiter simultanément. Le RLC envoie en priorité les PDU à retransmettre. Les autres PDU ne sont envoyés que si la fenêtre le permet. Avant la transmission à la couche MAC, les PDU peuvent être marqués et cryptés. Si un PDU est marqué, alors le récepteur enverra automatiquement un message de contrôle pour acquitter les PDU reçus et demander la retransmission des manquants. Le nombre maximum de PDU émis entre deux PDU marqués est donné par *poll\_PDU*.

C'est un paramètre fixé par le RRC. *Poll\_window* est un autre paramètre donné par le RRC qui indique le seuil de la fenêtre à partir duquel tous les PDU seront marqués. Le Récepteur envoie également un acquittement dès qu'il détecte qu'un PDU est manquant (détection par trous dans une séquence) ou dès qu'il reçoit un PDU de contrôle. Il y a un délai maximum entre deux acquittements. A la fin de ce délai, un acquittement est envoyé automatiquement. Ce délai est géré par le temporisateur Status Periodic. Pour chaque demande de retransmission, le temporisateur EPC (Estimated PDU Counter) est déclenché. Si à l'expiration de ce temporisateur le PDU manquant n'a pas été reçu, un message de contrôle est envoyé pour redemander sa retransmission et un nouveau temporisateur EPC est déclenché. Si un même PDU est reçu deux fois, le double est détruit automatiquement. Quand un SDU est complet (c'est à dire que tous les PDU qui le composent sont arrivés), les en-têtes sont enlevés de ses PDU et le SDU est ré-assemblé. Puis, le SDU est délivré à la couche supérieure. Si les SDU doivent être délivrés en séquence, alors un SDU complet est transmis à la couche supérieure uniquement lorsque tous les SDU avant lui ont été soit délivrés soit détruits. L'UMTS ne s'occupe pas de la retransmission des SDU.

### Paramètres du RLC AM

Les principaux paramètres du RLC AM sont [29] [30] :

TTI (Transmission Time Interval) : L'intervalle de temps sur la couche RCL.

PDU size : La taille du champs de donné d'un PDU.

Max pdu : le nombre maximum de PDU que peut transmettre une entité RLC par TTI.

Poll pdu : intervalle entre deux PDU marqués. Si un PDU est marqué alors le récepteur enverra un acquittement.

Window : la taille de la fenêtre indique le nombre de PDU que peut traiter une entité RLC simultanément.

Poll window : pourcentage de la fenêtre à partir de laquelle tous les PDU sont marqués.

Timer status periodic : le délais maximum entre deux messages d'acquittement.

Timer epc. Temps minimum entre deux demandes de retransmission pour un même PDU.

Maxdat : nombre maximum de retransmission pour un PDU.

### Exemples de transmissions

Nous allons détailler des exemples de transmissions entre 2 entités RLC. Dans ces exemples, un SDU se décompose en 3 PDU. Le RLC peut transmettre 4 PDU par TTI. Un PDU ne peut être retransmis qu'une seule fois. Le timer mrw est fixée à 35 ms. Poll pdu est fixé à 1, tous les PDU demande un acquittement. La figure 3.2 montre différents évènements qui peuvent survenir au cours de la transmission d'un SDU. Nous allons décrire ces évènements TTI par TTI.

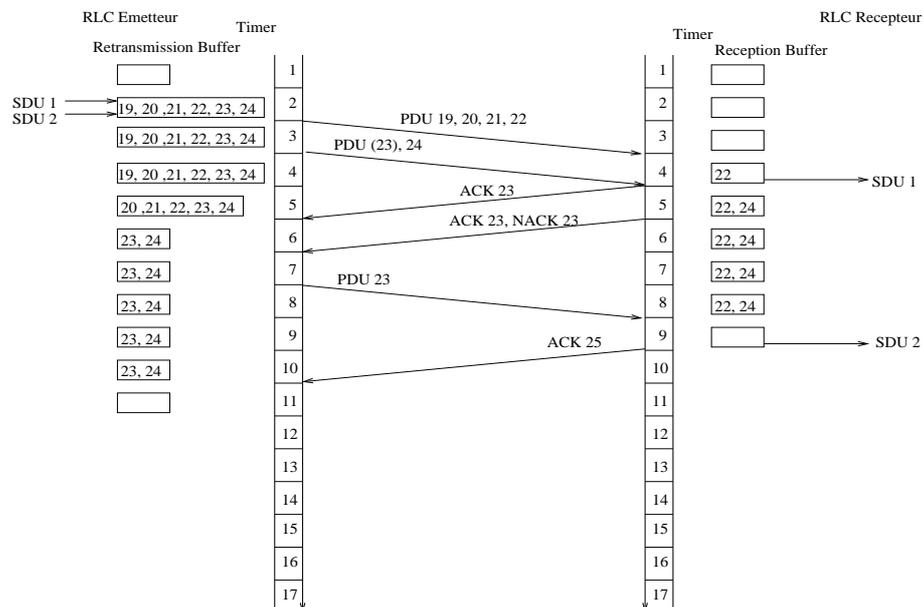


Figure 3.2: Communication entre deux entités RLC

TTI 1. Les Buffers du RLC émetteur et du RLC récepteur sont vides.

TTI 2. La couche RLC reçoit 2 SDU. Ils sont segmentés. Le premier est réparti sur les PDU 19, 20 et 21. Le second est réparti sur les PDU 22, 23 et 24. Les 6 PDU sont stockés dans le buffer de retransmission.

TTI 3. Le RLC émetteur envoie aux couches inférieures les PDU 19 à 22.

TTI 4. Le RLC émetteur envoie aux couches inférieures les PDU 23 et 24. Le PDU 23 est erroné et détruit. Le RLC récepteur reçoit les PDU 19 à 22. Le premier SDU est reconstitué et envoyé à la couche supérieure. Le PDU 22 est stockés dans le Buffer de réception.

TTI 5. Le RLC récepteur envoie un acquittement (ACK) pour le PDU 23. Il confirme ainsi la réception des PDU jusqu'au 22, et déclare être en attente du 23. Il reçoit le PDU 24, et le place dans le Buffer de réception. Il détecte l'absence du PDU 23. Il enclenche le timer EPC pour le PDU 23.

TTI 6. Le RLC récepteur envoie à nouveau un acquittement pour le PDU 23 et envoie un acquittement négatif (NACK) pour ce même PDU, puisqu'il en a détecté l'absence. Le RLC émetteur reçoit l'acquiescement du PDU 23. Il supprime de son buffer de retransmission les PDU 19 à 22.

TTI 7. Le RLC émetteur reçoit le NACK du PDU 23.

TTI 8. Le RLC émetteur retransmet le PDU 23.

TTI 9. Le RLC récepteur reçoit le PDU 23. Il assemble le second SDU et le transmet à la couche supérieure. Les PDU 22 et 24 sont supprimés du buffer de réception. Le timer EPC 23 expire. Mais le PDU 23 vient d'être reçu, on ne demandera donc pas sa retransmission.

TTI 10. Le RLC récepteur envoie un acquittement pour le PDU 25.

TTI 11. Le RLC émetteur reçoit l'acquiescement du PDU 25. Il supprime les PDU 23 et 24 du buffer de retransmission.

La figure 3.3 montre un exemple de perte de SDU.

TTI 1. Le RLC émetteur reçoit deux SDU. Ils sont segmentés. Le premier est réparti sur les PDU 19, 20 et 21. Le second est réparti sur les PDU 22,

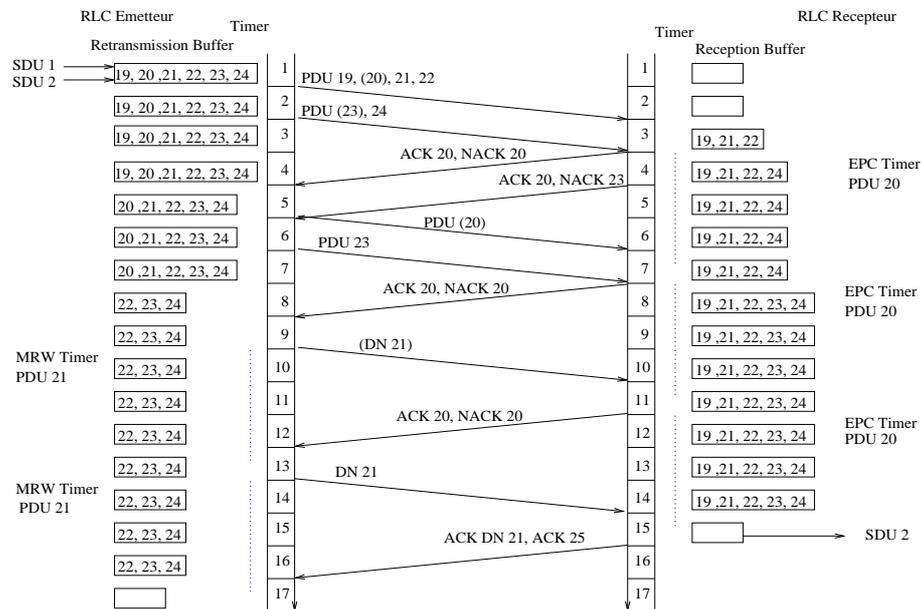


Figure 3.3: Communication entre deux entités RLC, perte d'un SDU

23 et 24. Les 6 PDU sont stockés dans le buffer de retransmission.

TTI 2. Le RLC émetteur envoie aux couches inférieures les PDU 19 à 22. Des erreurs sont détectées sur le PDU 20 par la couche physique du récepteur, qui le détruit.

TTI 3. Le RLC émetteur envoie aux couches inférieures les PDU 23 et 24. Des erreurs sont détectées sur le PDU 23 par la couche physique du récepteur, qui le détruit. Le RLC récepteur reçoit les PDU 19, 21 et 22 et la place dans le buffer de réception. Il constate que le PDU 20 est manquant. Il enclenche le Timer EPC pour le PDU 20.

TTI 4. Le RLC récepteur envoie un ACK et un NACK pour le PDU 20. Il reçoit le PDU 24 et le place dans le buffer de réception. Il constate que le PDU 23 est manquant.

TTI 5. Le RLC émetteur reçoit l'ACK et le NACK du PDU 20. Il supprime le PDU 19 du buffer de retransmission. Le RLC récepteur envoie un NACK pour le PDU 23.

TTI 6. Le RLC émetteur reçoit un ACK du PDU 20, et un NACK du PDU 23. Il retransmet le PDU 20. Des erreurs sont détectées sur le PDU 20 par la couche physique du récepteur, qui le détruit.

TTI 7. Le RLC récepteur retransmet le PDU 23. Le timer EPC du PDU 20 expire (sur le RLC récepteur). Le PDU 20 n'a toujours pas été reçu, un NACK pour le PDU 20 sera envoyé au prochain TTI, et le Timer EPC est à nouveau enclenché.

TTI 8. Le RLC récepteur envoie un ACK et un NACK pour le PDU 20. Il reçoit le PDU 23, et le place dans le buffer de réception. Bien que le second SDU soit complet, il n'est pas délivré, car l'option SDU en séquence a été sélectionnée.

TTI 9. Le RLC émetteur reçoit un NACK pour le PDU 20. Le PDU 20 a déjà été retransmis, donc le premier SDU est perdu. Les PDU 20 et 21 sont supprimés du buffer de retransmission (car issu du SDU à détruire).

TTI 10. Le RLC émetteur envoie un Discard Notification pour le PDU 21. Il signale ainsi au récepteur que tous les PDU antérieur au PDU 21 doivent être détruits. Un temporisateur MRW est enclenché pour le PDU 21. Le message Discard Notification doit être acquitté. Ce message sera retransmis à l'expiration de ce temporisateur si il n'a pas été acquitté entre temps. Or le message est perdu.

TTI 11. Le timer EPC du PDU 20 expire. Le PDU 20 n'a toujours pas été reçu, un NACK pour le PDU 20 sera envoyé au prochain TTI, et le Timer EPC sera à nouveau enclenché.

TTI 12. Pas d'évènement.

TTI 13. Le Timer MRW expire pour le PDU 21 (RLC émetteur).

TTI 14. Le RLC émetteur retransmet un Discard Notification pour le PDU 21. Le Timer MRW est à nouveau enclenché.

TTI 15. Le RLC récepteur reçoit le Discard Notification du PDU 21. Il supprime les PDU 19 et 21 du buffer de réception. Le second SDU peut enfin être transmis à la couche supérieure, car il n'y a plus de SDU antérieur

en attente. Les PDU 22 à 24 sont aussi supprimé du buffer de réception.

TTI 16. Le RLC récepteur envoi un ACK pour le PDU 25, et ACK pour le message Discard Notification.

TTI 17. Le RLC émetteur reçoit l'ACK du PDU 25. Il supprime les PDU 22 à 24 du buffer de retransmission. L'acquittement du Discard Notification est également reçu. Le timer MRW n'aura pas besoin d'être réenclenché à son expiration.

Les deux exemples d'échanges entre l'entité RLC du récepteur et l'entité RLC de l'émetteur montrent l'importance des retransmissions dans les délais de transmissions. Dans le premier exemple, le premier SDU arrive pendant le TTI 2 et peut être délivré dès le TTI 4. Dans le second exemple, le second SDU arrive au TTI 1 et n'est délivré qu'au TTI 15. Il a fallu 12 TTI de plus pour le délivrer.

### 3.2.4 Medium Access Control

La couche MAC [32] demande des PDU à la couche RLC et les transmet à la couche physique. Une entité MAC peut travailler avec plusieurs entités RLC. Au début de chaque TTI, le MAC demande un certain nombre de PDU au RLC. La quantité de PDU qui sera demandée à une entité RLC dépend de la priorité de l'utilisateur, de la priorité de son application, et des ressources disponibles. Le RRC fixe une limite maximum au nombre de PDU que la couche MAC peut demander à une entité RLC spécifiée. Le MAC transmet tous les PDU reçus à la couche Physique sous la forme de Bloc de Transport. Au récepteur, la couche MAC transmet les PDU reçus à la couche RLC.

### 3.2.5 Couche Physique

La couche physique reçoit les blocs de transport de la couche MAC. Elle ajoute des bits de détection d'erreur sur chacun d'eux (Cyclic Redundancy CheckSum) et les code en trame radio. Elle peut multiplexer plusieurs blocs sur une même trame. Puis la couche physique transmet les trames radio au récepteur. Le récepteur les décode en Bloc de Transport et détecte les erreurs. S'il y a eu multiplexage, alors les trames radios sont démultiplexées. Si un bloc de transport est erroné, il peut être envoyé à la couche supérieure accompagné d'un signal d'erreur. Il peut être conservé, ou il peut être

détruit. Si aucune erreur n'est détectée, les Blocs de Transport sont transmis à la couche MAC. Les codes proposés par 3GPP sont les codes convolutifs de taux  $\frac{1}{2}$  et  $\frac{1}{3}$  et les turbo codes. Il est également possible de ne pas coder les PDU [33].

### Codes convolutifs

Les codes convolutifs génèrent de la redondance en étalant et mélangeant les symboles sur plusieurs bits. Un code est caractérisé par 3 paramètres: le nombre de bits en entrée  $k$ , le nombre de bits en sortie  $n$ , et la taille de la mémoire  $m$ . Les bits d'entrée sont insérés un à un dans le registre. Les  $n$  bits de sortie sont calculés en fonction des  $n$  polynômes (un par sortie) et des  $m$  bits en mémoire. L'algorithme de Viterbi est utilisé pour le décodage. La figure 3.4 montre la structure et les polynômes des codes convolutifs utilisés dans l'UMTS.

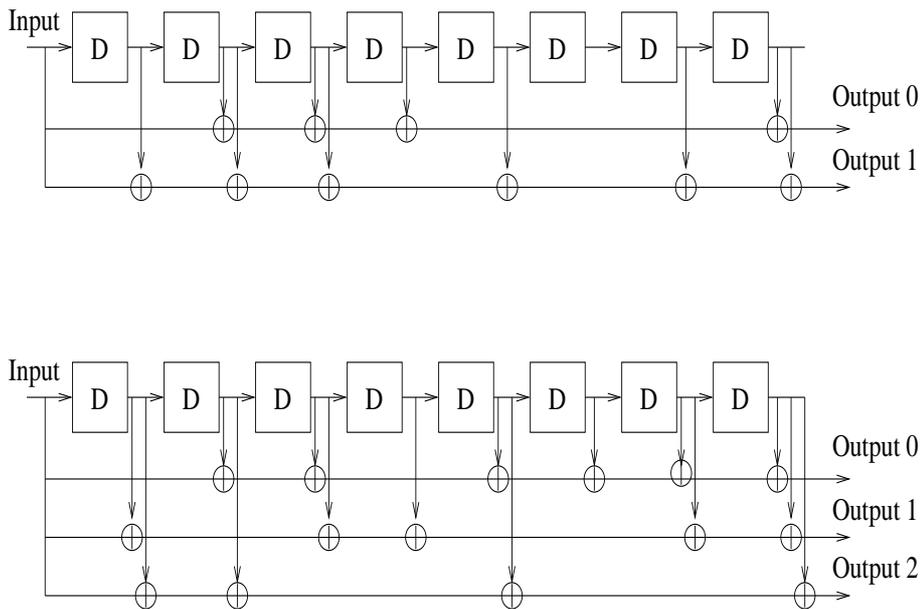


Figure 3.4: Codes Convolutifs utilisés dans l'UMTS

### Turbo code

Les turbo codes utilisent deux techniques supplémentaires, le codage en parallèle (le codage en série existe également) concaténé et le décodage itératif. Un Bloc de Transport  $m_0$  est codé en un mot de code  $c_1$  avec un code convolutif. En parallèle,  $m_0$  est entrelacé en un bloc de transport  $m_1$ , puis  $m_1$  est codé avec un code convolutif et donne la trame radio  $c_2$ . Puis,  $m_0$ ,  $c_1$  et  $c_2$  sont multiplexés et transmis sur le canal. La figure 3.5 montre le schéma des turbos codes utilisé dans l'UMTS.

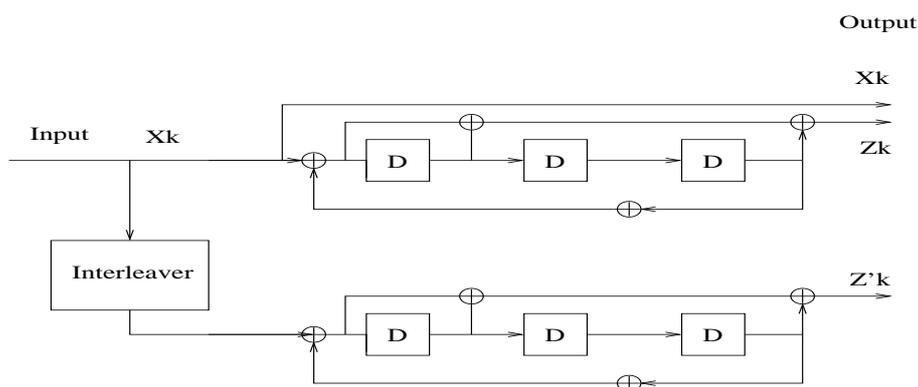


Figure 3.5: Turbo Code utilisé dans l'UMTS

### 3.2.6 Hybrid ARQ

Les couches RLC et physique jouent un grand rôle pour un transport fiable (en anglais reliable) des données. La couche physique s'occupe de la correction des erreurs (par les codes correcteurs) et de leurs détections (par le CRC). La couche RLC s'occupe des retransmissions des PDU perdus ou erronés (en mode AM et UM).

La combinaison des techniques de codes correcteurs et de retransmission de données s'appelle Hybrides ARQ (HARQ). Il existe trois types d'HARQ selon le type de données retransmis [38] [34].

#### HARQ type 1

Les données sont codées puis transmises. A la réception, la trame radio est décodée et la présence d'erreurs détectée. S'il y a des erreurs dans la trame,

elle est détruite. Le Récepteur demande sa retransmission qui sera effectuée de manière identique.

### **HARQ type 2**

Sur la couche physique, les trames erronées ne sont pas détruites mais conservées en mémoire. Le RLC récepteur demande la retransmission des PDU manquants. La trame retransmise est en général différente de la première. Elle transporte de nouvelles informations de redondance. Elle est combinée avec la première trame reçue ce qui doit permettre une meilleure correction. Les trames retransmises sont en général différentes pour chaque retransmission et une nouvelle retransmission ne peut être utilisée qu'en la combinant avec les transmissions précédentes.

### **HARQ type 3**

En HARQ 3, comme en HARQ 2, les trames erronées ne sont pas détruites mais conservées en mémoire. Le RLC récepteur demande la retransmission des PDU manquants. Les trames retransmises sont combinées avec celle reçues précédemment, mais chaque trame peut également être décodée individuellement [39].

La façon d'associer les trames retransmises aux trames erronées conservées en mémoire n'est pas encore spécifiée dans les standards UMTS. Mais [47] propose une méthode intéressante de détection aveugle des paquets retransmis.

## **3.3 Choix des paramètres**

Le temps nécessaire pour délivrer un SDU dépend de la probabilité de perte des PDU, du nombre de PDU d'un SDU, du nombre maximum de fois qu'un PDU pourra être retransmis, et du nombre de PDU qui peut être transmis en 1 TTI. Le taux de perte des PDU dépend de la taille du PDU, du code correcteur sélectionné et de l'état du canal. Dans la suite de ce chapitre, nous étudierons les taux de perte et les délais de transfert des SDU dans l'UTRAN en fonction des valeurs fixées, de l'état du canal, de la taille du PDU, du nombre de retransmission autorisée et du code correcteur utilisé.

La durée des temporisateurs (timers) influe également, mais dans une moindre mesure, sur la durée totale de transmission dans l'UTRAN : en diminuant ou en allongeant la durée du temporisateur MRW, par exemple,

il est possible de retarder ou d'accélérer le temps entre deux retransmissions. Si fixer la durée de ces paramètres est intéressantes pour améliorer le comportement global de l'UTRAN, il ne nous a pas paru pertinent de fixer ces valeurs en fonctions du service transporté. Nous réduirons donc notre étude aux paramètres cités au paragraphe précédent. Des valeurs pour la durée des temporisateurs peuvent être trouvée dans [30] et [31].

La table 3.1 donne la liste et les valeurs des paramètres que nous ferons varier dans nos expériences. Les abréviations utilisées sur les légendes des figures sont indiquées dans la dernière colonne.

Paramètre	valeurs	abréviation
code correcteur	code convolutif de taux 1/2	cc1/2
	code convolutif de taux 1/3	cc1/3
	turbo code	tc
	sans codage	noc
maxdat	0, 1, 2, $\infty$	md
Longueur du PDU (en bits)	80, 160, 320	

Table 3.1: Paramètres variant dans nos expériences

### 3.3.1 Conditions expérimentales

Trois tailles de PDU ont été testées, leurs champs de données sont respectivement de : 80 [34], 160 [35] et 320 bits [36] [40]. Pour ces tailles de PDU, les SDU sont respectivement segmentés en 16, 8 et 4 PDU. Les 3 codes correcteurs proposés par ont été utilisés (le turbo code n'a été utilisé que pour les PDU de 320 bits). Le mode sans code correcteur a également été utilisé. Pool pdu a été fixé à un : à chaque fois qu'un PDU sera détecté manquant, une retransmission sera demandé [37]. Enfin, maxdat, le nombre de retransmissions autorisées, a été fixé selon les expériences à 0, 1, 2 ou à l'infini. Le canal utilisé est de type gaussien, et il varie au cours des expériences entre 0 et 10 dB.

### 3.3.2 Mesures sur les pertes

Nous mesurons le taux de perte des SDU du RLC de l'émetteur au RLC du Récepteur. Le SDU ER (SDU Error rate) est calculé par:

$$SDUER = 1 - \frac{r}{g} \quad (3.1)$$

où  $r$  est la quantité de SDU reçus au RLC Récepteur et  $g$  la quantité de SDU reçus par le RLC de l'émetteur. Nous rappelons qu'un SDU n'est reçu que si tous les PDU qui le composent ont été reçus. La demande de retransmission d'un SDU perdu ne se fait pas sur la couche RLC.

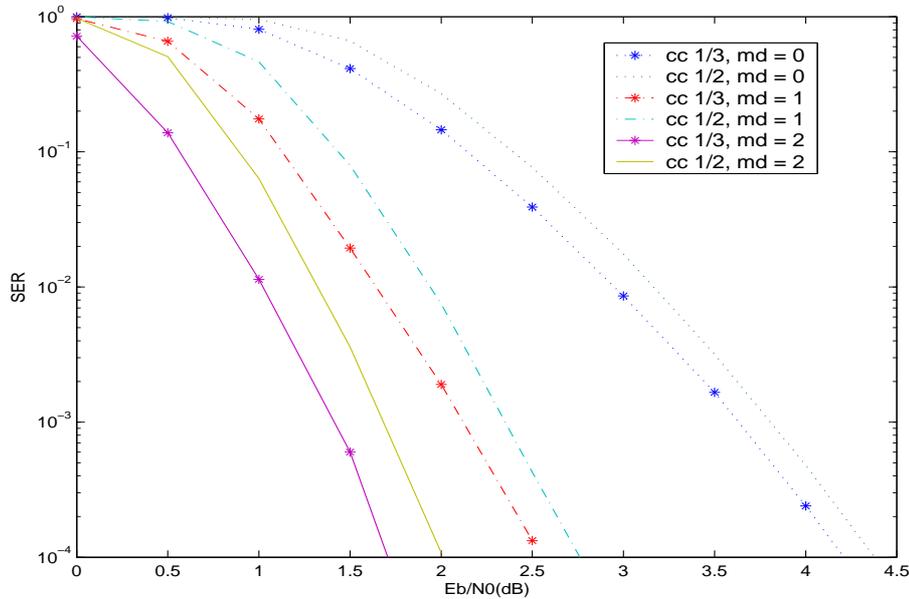


Figure 3.6: SDU Error Rate pour des PDU de 80 bits avec codes correcteurs.

Les figures 3.6 à 3.11 nous montrent que plus le taux du code est faible, plus faible est la proportion de SDU perdus. Plus il y a de retransmissions et moins il y a de SDU perdus. Le codage permet un gain notable de 6 dB par rapport à un ARQ sans codage. Par contre, à nombre de retransmissions égal, le gain entre le codage de taux 1/2 et de taux 1/3 n'est que de 0,5 dB. La retransmission accroît l'effet du codage. Quand nous ne retransmettons pas, utiliser le codage 1/3 plutôt que le codage 1/2 divise le taux de perte par 2, alors qu'avec 1 retransmission le taux de perte est divisé par 5, et avec 2 retransmissions le taux de perte est divisé par 10. Pour avoir un taux de perte de 0, de 0 à 5 dB, il est nécessaire de coder l'information et d'autoriser un nombre infini de retransmissions. Dans cet intervalle, il n'est pas possible de transmettre l'information sans codage, car le débit que nécessiteraient les retransmissions serait supérieur au débit qu'autorise l'UMTS. A partir de 4 dB, un nombre infini de retransmissions permet d'avoir un SER de 0, même sans codage. On note également que la taille du PDU à une faible influence

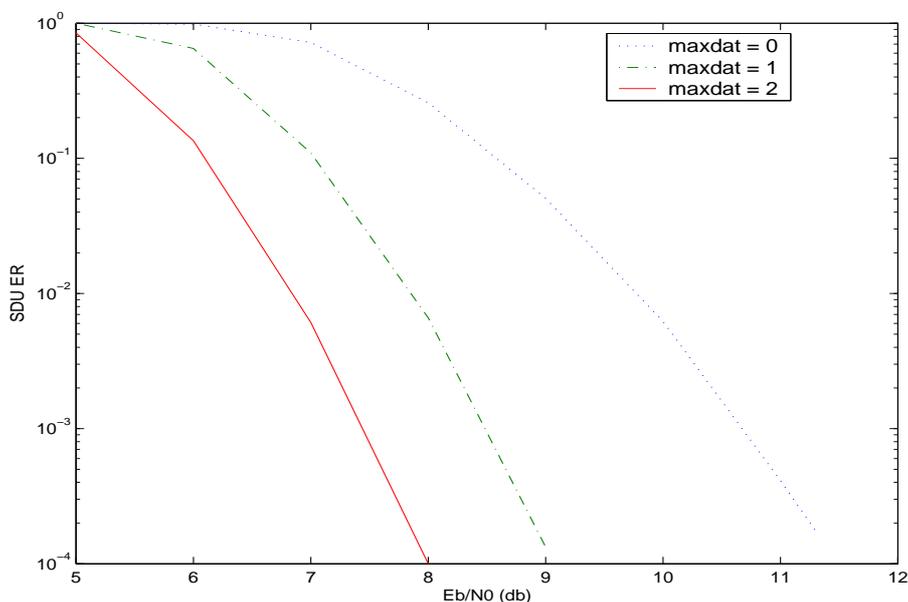


Figure 3.7: SDU Error Rate pour des PDU de 80 bits sans codes correcteurs.

sur le SDU ER. Les PDU de plus petites tailles ont une probabilité d'erreur plus faible, mais comme ils sont plus nombreux à être transmis, le taux de perte SDU est peu modifié.

### 3.3.3 Mesures sur les délais

Nous effectuons maintenant des mesures sur les délais de transmission des SDU dans l'UTRAN. Les mesures commencent à l'arrivée d'un SDU sur le RLC de l'émetteur et s'arrêtent lorsque le SDU est délivré par le RLC du récepteur à la couche supérieure. Le temps de transfert d'un SDU perdu n'est pas pris en compte.

Les mesures montrent que les délais augmentent avec le nombre de re-transmissions et avec la détérioration du canal. Les délais croissent de façon exponentielle avec la détérioration du canal mais lorsque le canal est bon, les délais atteignent une valeur asymptotique légèrement supérieur à 10 ms. Cela correspond au temps de transfert sur un TTI et à du temps de traitement. Les délais augmentent avec le SDU ER : en effet, la principale cause des délais est la perte et la retransmission de PDU. La perte d'un PDU entraîne un retard d'au moins 4 TTI (détection de la perte et envoi d'un

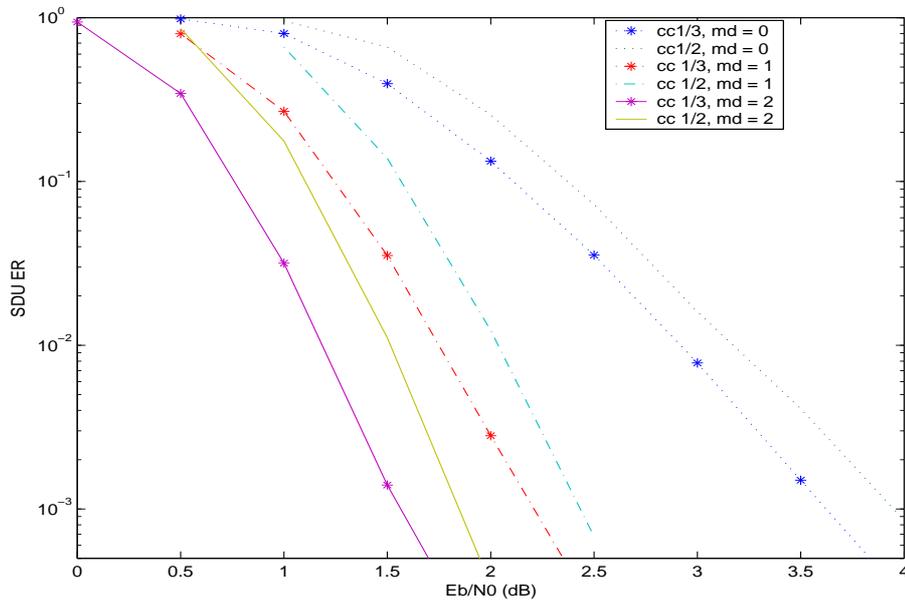


Figure 3.8: SDU Error Rate pour des PDU de 160 bits avec codes correcteurs.

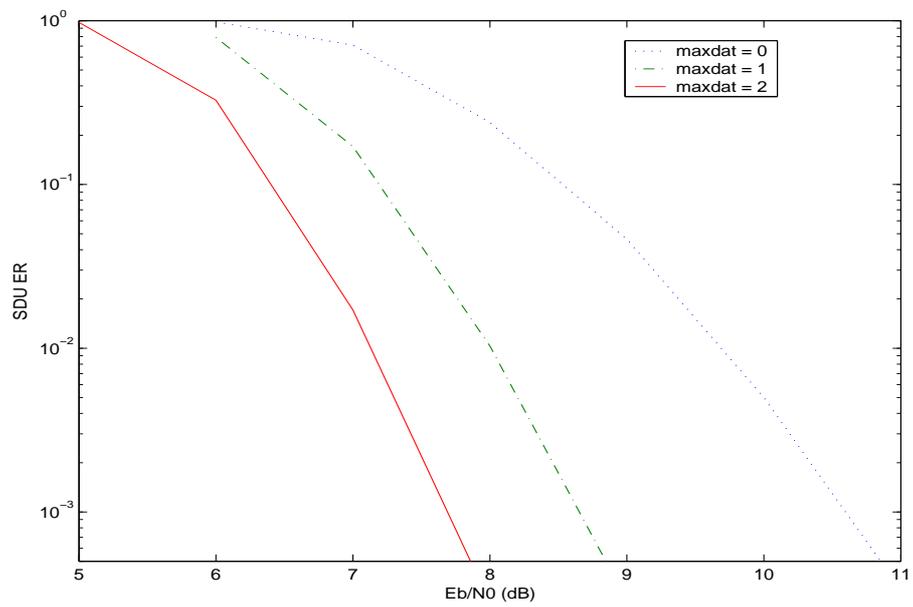


Figure 3.9: SDU Error Rate pour des PDU de 160 bits sans codes correcteurs.

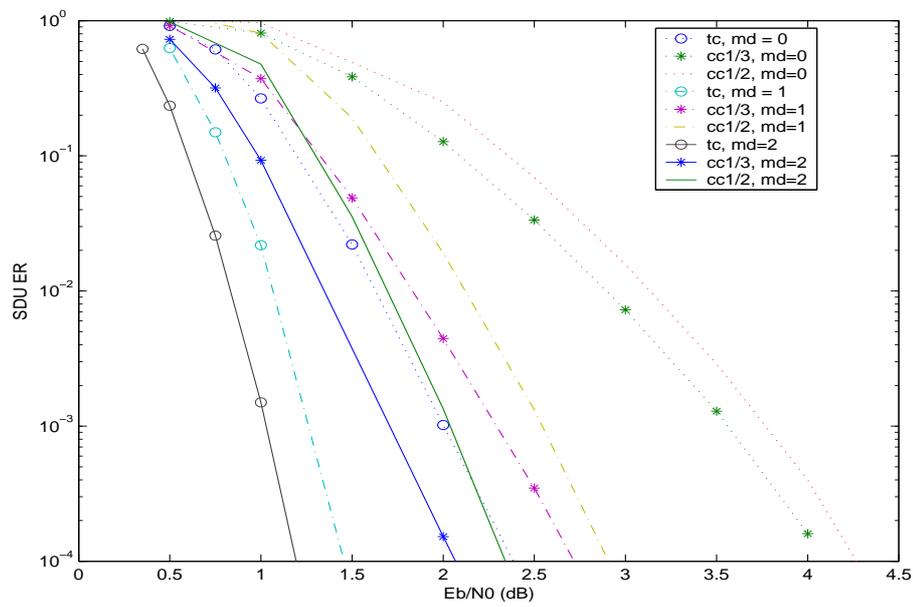


Figure 3.10: SDU Error Rate pour des PDU de 320 bits avec codes correcteurs.

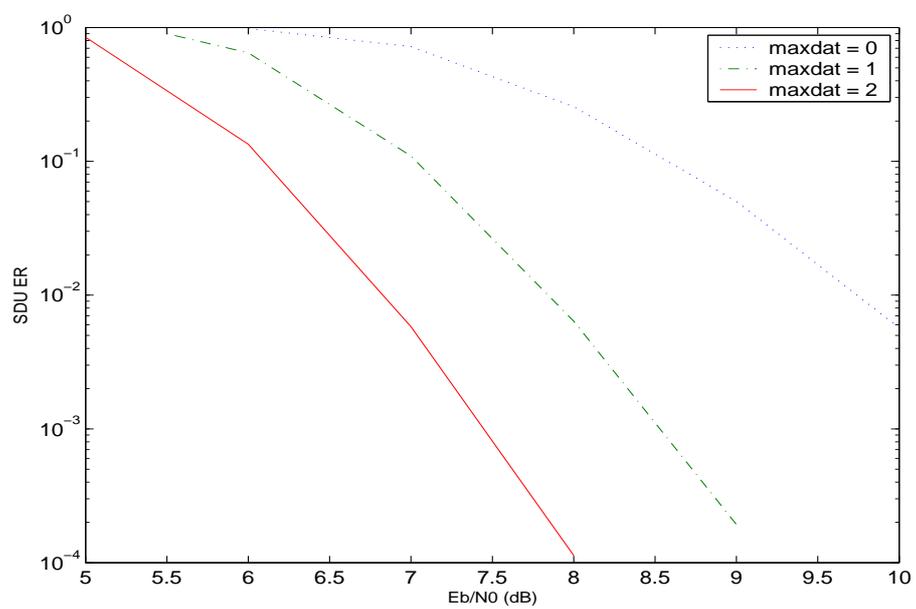


Figure 3.11: SDU Error Rate pour des PDU de 320 bits sans codes correcteurs.

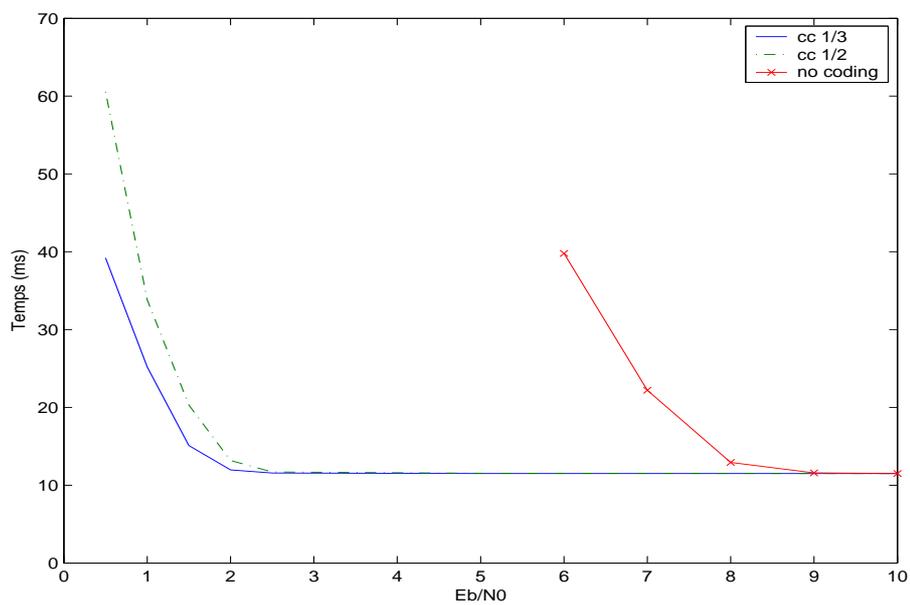


Figure 3.12: Délais moyens de transfert, PDU = 80 bits, maxdat = 0.

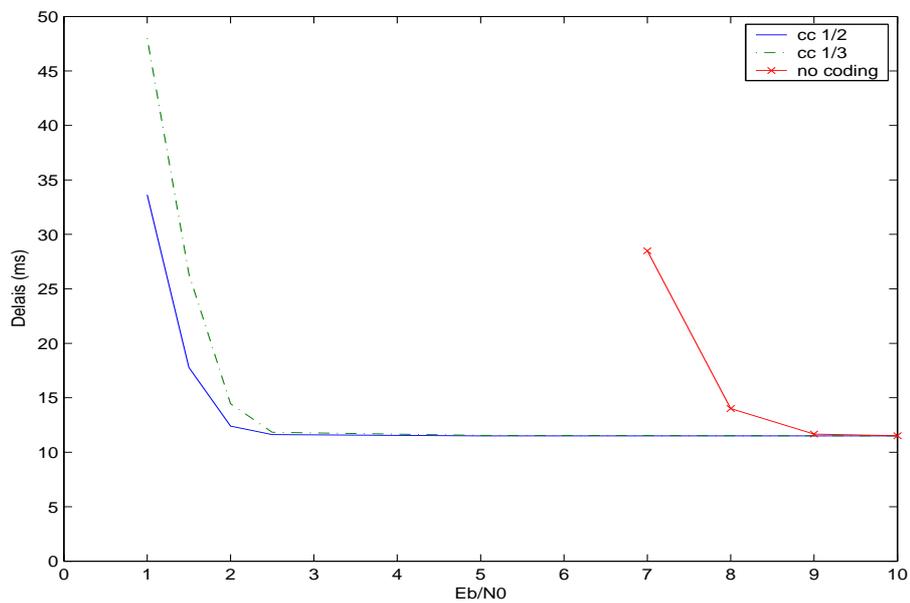


Figure 3.13: Délais moyens de transfert, PDU = 160 bits, maxdat = 0.

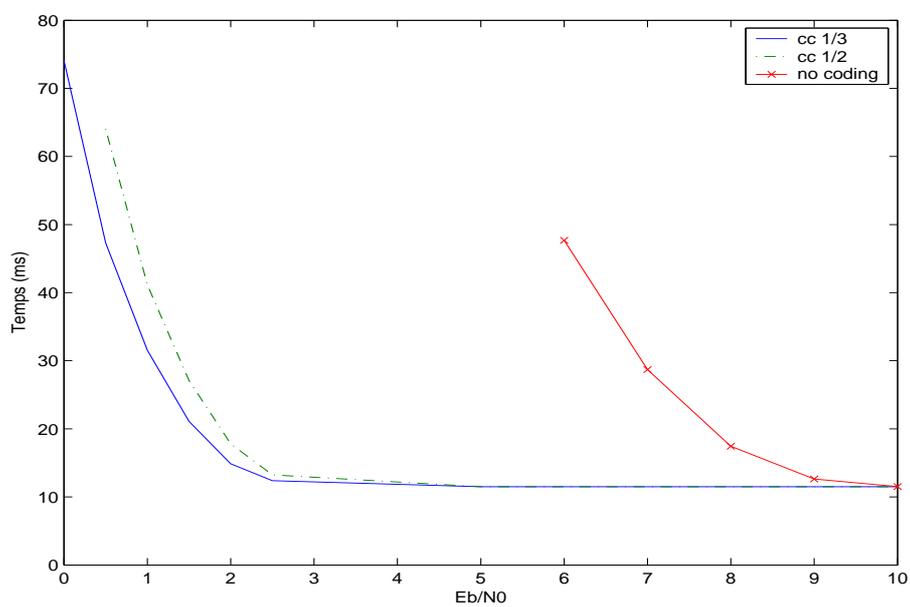


Figure 3.14: Délais moyens de transfert, PDU = 80 bits, maxdat = 1.

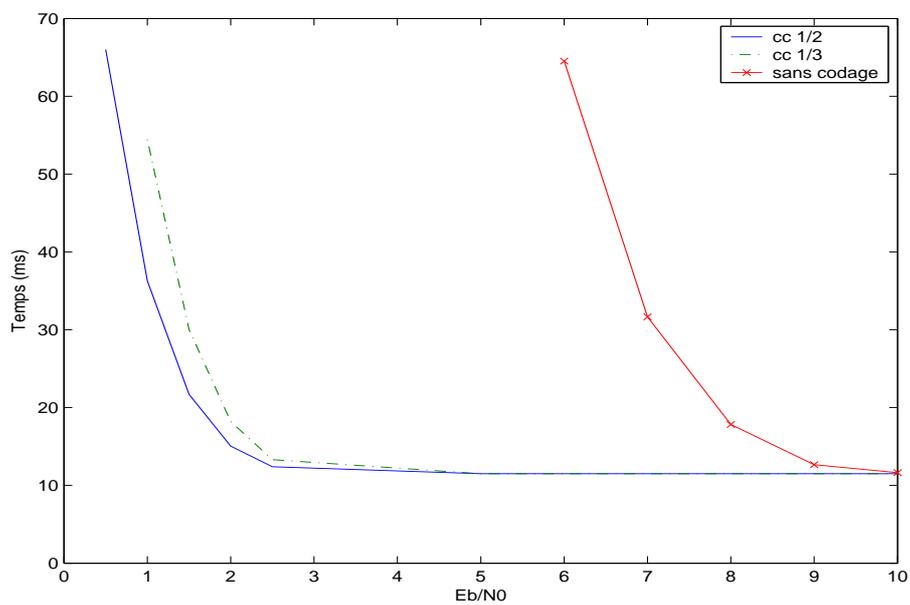


Figure 3.15: Délais moyens de transfert, PDU = 160 bits, maxdat = 1.

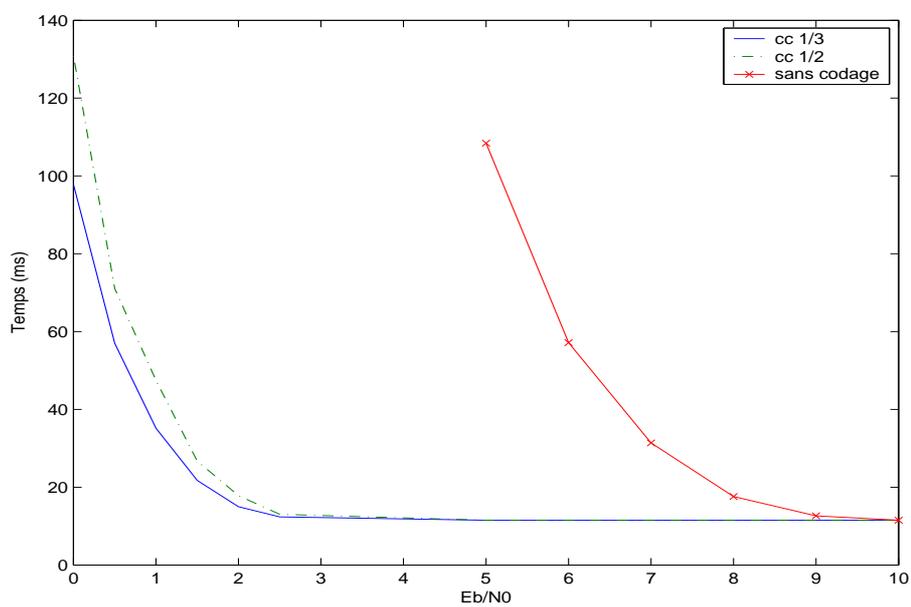


Figure 3.16: Délais moyens de transfert, PDU = 80 bits, maxdat = 2.

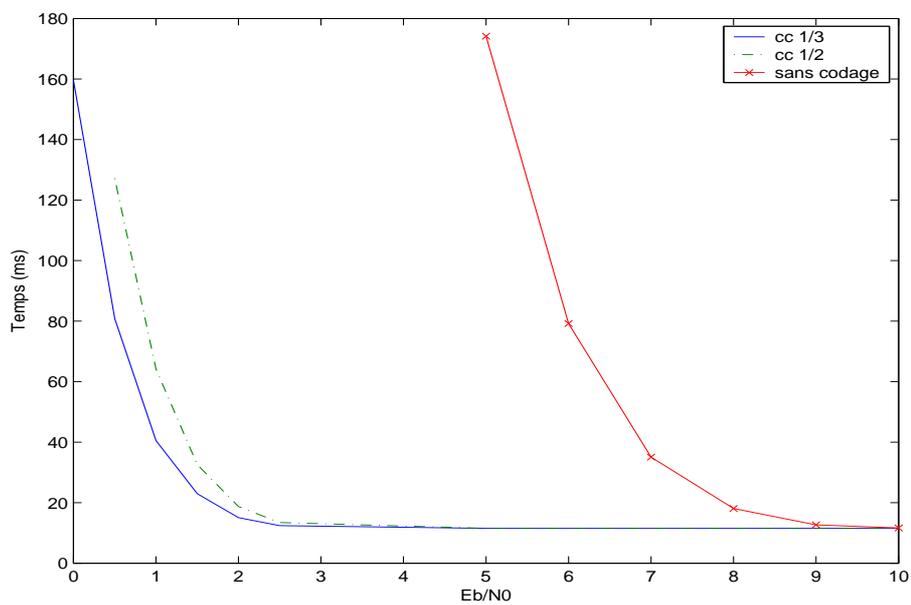


Figure 3.17: Délais moyens de transfert, PDU = 160 bits, maxdat = 2.

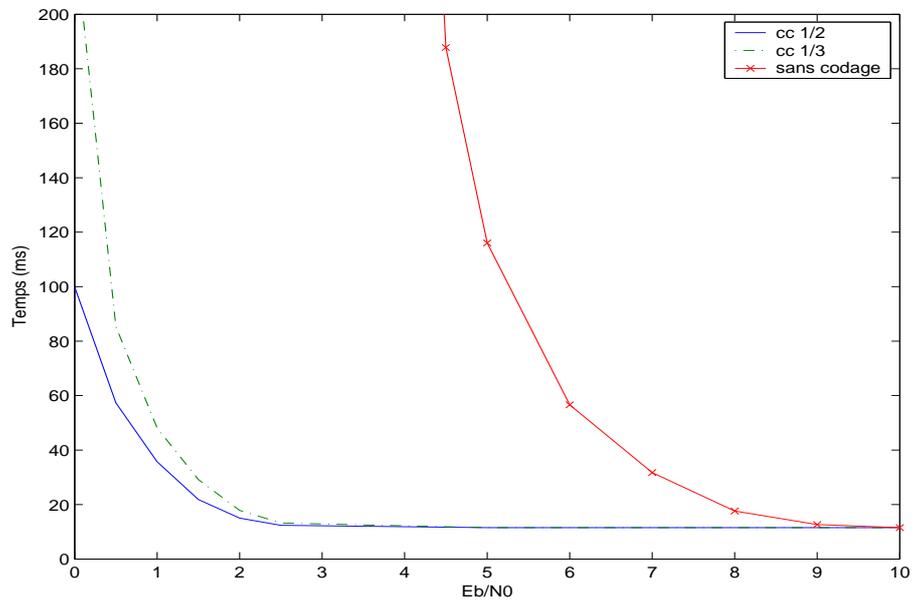


Figure 3.18: Délais moyens de transfert, PDU = 80 bits, maxdat = infini.

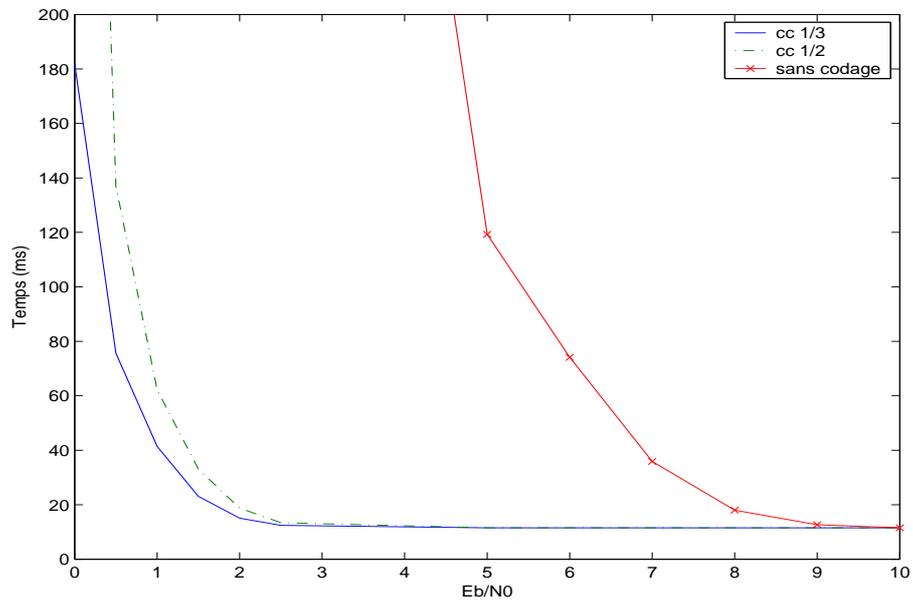


Figure 3.19: Délais moyens de transfert, PDU = 160 bits, maxdat = infini.

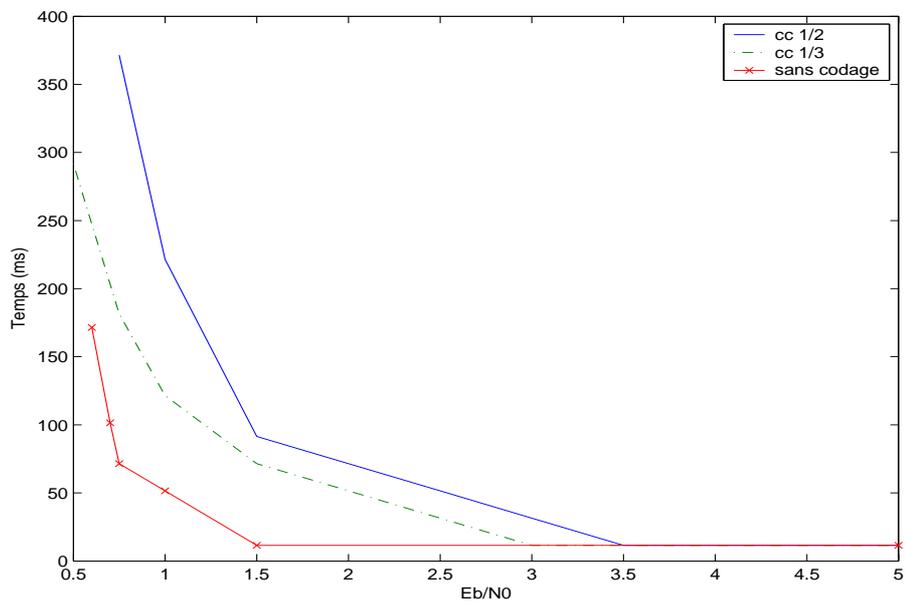


Figure 3.20: Délais pour recevoir 99 % des SDU.

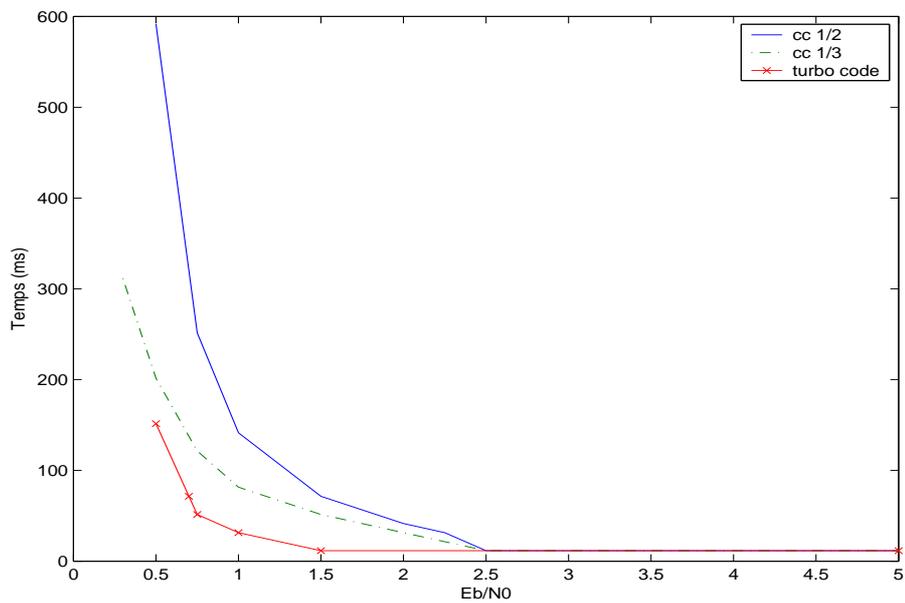


Figure 3.21: Délais pour recevoir 90 % des SDU.

statut demandant la retransmission par le récepteur, réception du statut et retransmission de ce PDU par l'émetteur). Comme un SDU n'est délivré à la couche supérieure que si tous ses PDU sont correctement arrivés au récepteur, les délais de bout en bout d'un SDU dépendent du délai de son PDU le plus retransmis.

Nous observons également sur les figures 3.16 à 3.19 que les délais moyens n'augmentent pas de façon significative au-delà de la deuxième retransmission, les mécanismes de codages combinés avec les mécanismes de retransmission permettant d'avoir un canal très fiable à partir de la deuxième retransmission.

Certaines courbes ne sont définies qu'à partir de certaines valeurs de  $E_b/N_0$ . Ce sont des courbes pour lesquelles le nombre de retransmission autorisé pour un même PDU est limité (maxdat différent de + infini) et pour des états du canal très mauvais. Le mauvais état du canal entraîne des taux de pertes de PDU très élevés, et donc un SDU ER très élevés (une majorité de paquets perdus). Les mesures pour les rares paquets arrivés non alors plus de sens, et ne sont donc pas indiqués.

Un point particulier concerne le cas où la retransmission n'est pas autorisée (figure 3.12 et 3.13). Si les SDU sont délivrés en séquence, un SDU complet ne sera délivré que si tous les SDU qui le précèdent ont soit été délivrés, soit été détruits. (Un SDU est détruit au récepteur lorsqu'il reçoit un statut du receveur demandant sa destruction. Ce statut est envoyé quand l'émetteur a reçu un nombre de demandes de retransmission pour un même PDU supérieur au nombre autorisé). Aussi, lorsque les retransmissions ne sont pas autorisées, les délais de bout en bout ne sont pas constants mais peuvent s'accroître avec la détérioration du canal.

### 3.3.4 Calcul de l'efficacité des paramétrages

L'efficacité  $E$  est donnée par :

$$E = \frac{(1 - t) * p}{r * f} \quad (3.2)$$

où  $t$  est le taux de perte des SDU,  $p$  est le nombre de bits dans un PDU,  $r$  est le nombre moyen de transmissions d'un PDU et  $f$  est le nombre de bits d'une trame radio.

La figure 3.22 montre l'efficacité des différents codes correcteurs pour un PDU de taille 80, en fonction de maxdat et de l'état du canal.

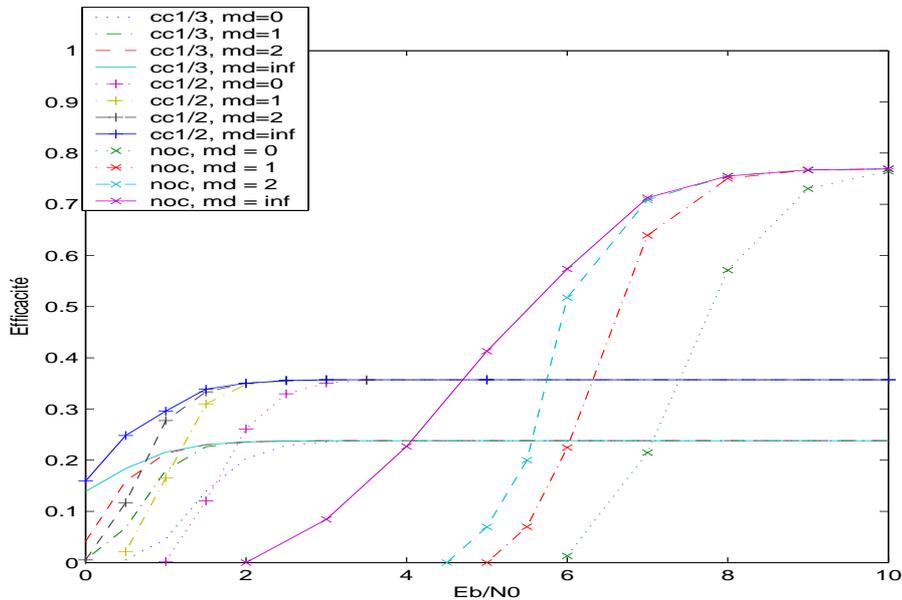


Figure 3.22: efficacité du paramétrage, PDU de 80 bits.

cc 1/3 indique qu'il s'agit du code convolutif de taux 1/3, cc1/2 qu'il s'agit du code convolutif de taux 1/2, noc, qu'il s'agit de l'option sans codage et tc qu'il s'agit du turbo code. Md indique la valeur de maxdat, le nombre maximum de retransmissions autorisées.

La figure 3.23 montre l'efficacité des différents codes correcteurs pour un PDU de taille 160, en fonction de maxdat et de l'état du canal.

La figure 3.24 montre l'efficacité des différents codes correcteurs pour un PDU de taille 320, en fonction de maxdat et de l'état du canal pour des codes convolutifs et pour l'option sans codage.

La figure 3.25 montre l'efficacité des turbo codes pour un PDU de taille 320, en fonction de maxdat et de l'état du canal.

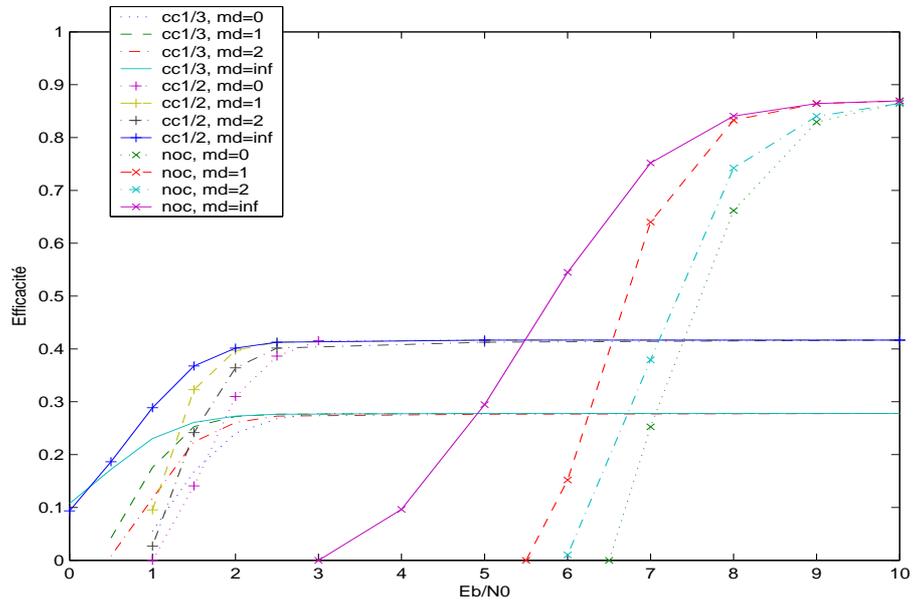


Figure 3.23: efficacité du paramétrage, PDU de 160 bits.

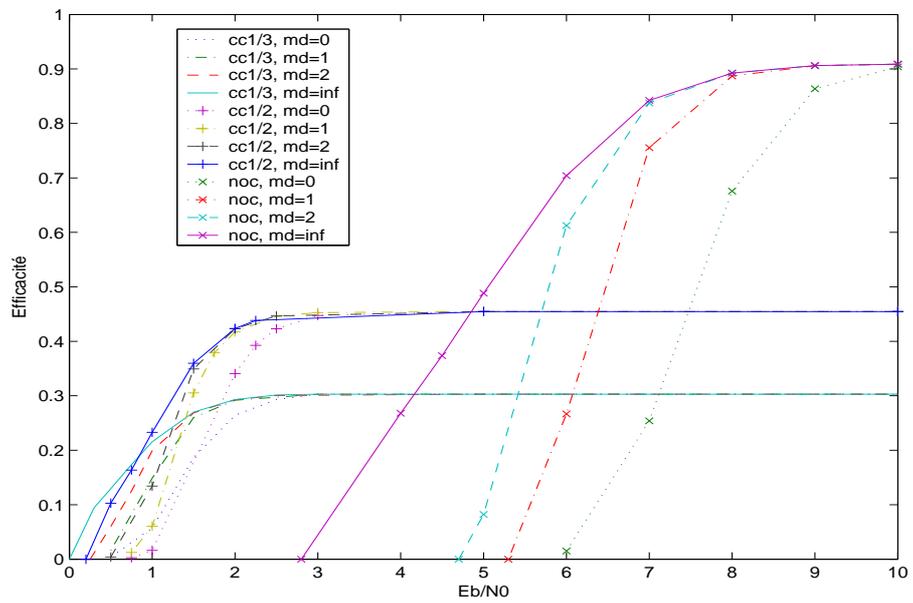


Figure 3.24: efficacité du paramétrage, PDU de 320 bits.

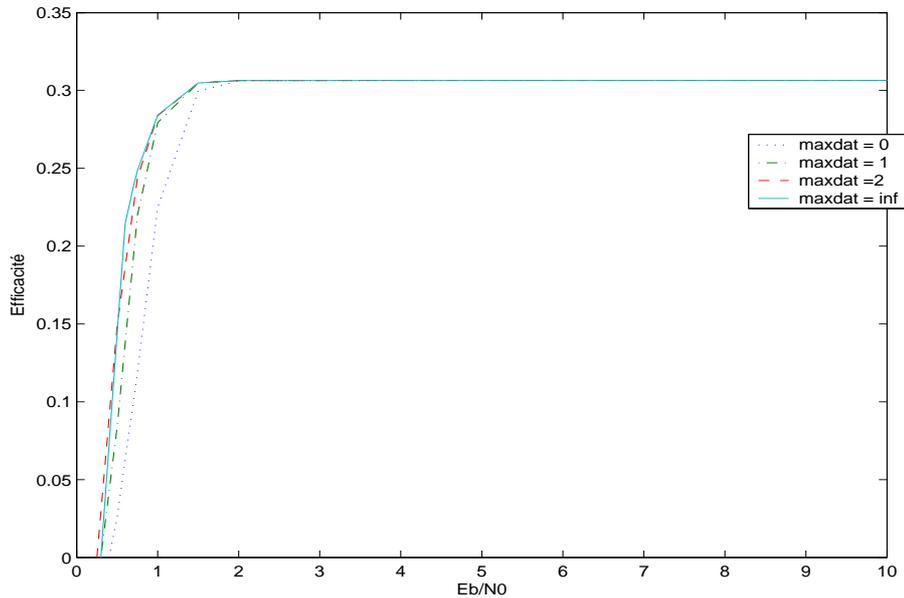


Figure 3.25: efficacité du paramétrage, PDU de 320 bits.

L'efficacité est très basse quand le canal est de mauvaise qualité, elle progresse quand le canal s'améliore et finit par se stabiliser. On remarque que la valeur pour laquelle l'efficacité se stabilise dépend du taux de codage et de la taille du PDU, et non du nombre de retransmissions (en fait, l'efficacité se stabilise quand la qualité du canal rend les retransmissions inutiles).

Pour un même type de codage, l'efficacité augmente avec le nombre de retransmissions. En effet, comme un SDU n'est utilisable que si tous ses PDU sont arrivés, il peut suffire de la retransmission d'un seul PDU pour qu'un SDU soit complet.

La figure 3.26 montre l'efficacité des différents codes correcteurs en fonction de la taille du PDU pour un nombre de retransmission non limité.

Au dessus de 1.5 dB, les PDU de petites tailles ont une plus grande efficacité, car leurs taux de pertes sont plus faibles que pour les PDU de grandes tailles. Mais, lorsque le canal s'améliore, l'efficacité augmente avec la taille du PDU. En effet, comme il y a un entête de 32 bits dans chaque PDU, la proportion de l'entête diminue avec l'accroissement de la taille du

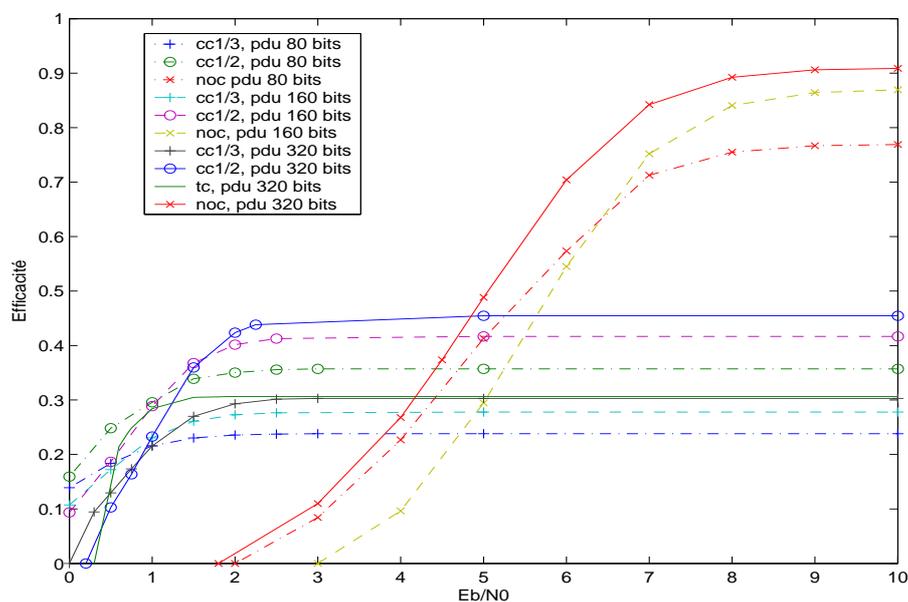


Figure 3.26: efficacité des codes correcteurs en fonction de la taille du PDU.

champs de données.

L'efficacité d'un code correcteur dépend fortement de l'état du canal. Si l'efficacité d'un code correcteur est toujours plus faible quand le canal est de mauvaise qualité, le code de taux 1/2 et l'option sans codage se dégradent beaucoup plus vite que le code convolutif de taux 1/3 et le turbo code.

La meilleur efficacité est obtenue:

- en dessous de 1.5 dB par le turbo code et le code convolutif de taux 1/3
- de 1.5 dB à 5 dB par le code convolutif de taux 1/2
- au dessus de 5 dB par l'option sans code correcteur.

### 3.4 Configuration d'un service

Un service est configuré à l'établissement de sa connexion. Sa configuration comprend un ensemble de Format de Transport (Transport Format, TF) qui décrit comment le service est transporté [56]. La configuration d'un service comprend également un TRCH et un CCTRCH qui seront décrits au chapitre 5.

### 3.4.1 Format de Transport

Les paramètres suivants constituent les champs d'un Format de Transport. Ces paramètres définissent la forme et les caractéristiques d'émission des blocs de transport.

Block Size (BS) : Taille du Bloc en bits. Dans [57] les tailles de blocs sont comprises entre 32 et 640 bits.

Block Set Size (BSS) : Le nombre de blocs par TTI multiplié par la taille d'un bloc. Cela permet de définir le nombre de bloc à transmettre pour ce canal à ce TTI et de régler le débit. Avec le TTI, le BSS permet de calculer le débit instantané.

Cyclic Redundancy CheckSum (CRC) : Le CRC sert sur la couche physique à éliminer les trames erronées. Un CRC peut avoir une taille de 8, 12, 16 ou 24 bits. La taille du CRC dépend uniquement du BER demandé dans le QoS IE. La table 3.2 donne la taille du CRC en fonction du BER requis par l'application.

Code QoS IE	Residual BER	Taille du CRC
0001	$5 \cdot 10^{-2}$	8 bits
0010	$1 \cdot 10^{-2}$	8 bits
0011	$5 \cdot 10^{-3}$	8 bits
0100	$4 \cdot 10^{-3}$	8 bits
0101	$1 \cdot 10^{-3}$	12 bits
0110	$1 \cdot 10^{-4}$	12 bits
0111	$1 \cdot 10^{-5}$	16 bits
1000	$1 \cdot 10^{-6}$	16 bits
1001	$6 \cdot 10^{-8}$	24 bits
1111	Reserved	-

Table 3.2: Taille du CRC en fonction du BER requis

TTI : Un TTI est l'unité de temps aux couches MAC et RLC. Ils peuvent durer de 10 à 80 ms. Les TTI courts conviennent à des applications qui ont-elles mêmes des délais courts. Au contraire, un TTI long est préférable pour les applications n'ayant pas de contraintes de temps. Sur la couche physique, la durée d'un TTI est de 10 ms. Le principal usage d'un TTI long sur les couches RLC et MAC est de permettre l'entrelacement entre les TTI.

Codage : quatre types de code correcteur sont proposés : le code convolutif de taux  $\frac{1}{2}$ , le code convolutif de taux  $\frac{1}{3}$ , le turbo code et il est également possible de transmettre sans utiliser de code correcteur.

### 3.4.2 Transport Format Set

Le Transport Format indique le débit instantané par le BSS. Lors de la configuration, il est donné une liste de Formats de Transport. Chacun des TF varie uniquement par le BSS, donc par le nombre de block transmis. L'ensemble des TF d'un service est indiqué à un utilisateur dans le Transport Format Set (TFS, Liste de Format de Transport).

### 3.4.3 Transport Format Combination

L'UMTS permet le multiplexage entre les services d'un même utilisateur (plusieurs services simultanément sur le même ensemble de ressources). Aussi, il est créé un Transport Format Combination Set (TFCS) qui est une liste de Transport Format Combination (TFC). Un Transport Format Combination est une combinaison de TF, un par service multiplexé. Un TFC peut être indiqué dans le TFCS si la combinaison est valide, c'est-à-dire, si le paramétrage et l'allocation de ressource permettent la transmission des paquets demandés. En pratique, seule une fraction des TFC possibles est indiquée dans le TFCS. La Figure montre un exemple du TFCS de deux services multiplexés. Le premier élément de chaque couple indique le BSS du premier service, le second élément du couple indique le BSS du second service. Le TFCS est transmis à un utilisateur lors de la configuration des services de son CCTRCH. La liste est ordonnée. Le premier couple est le premier TFC, et le nième couple est le TFC numéro n. A chaque TTI, on indique dans le TFCI, le numéro du TFC sélectionné (le TFCI est détaillé dans le chapitre 4).

Si un seul TRCH est transmis dans un CCTRCH, alors le TFCS est simplement une copie du TFS.

## 3.5 Paramétrage des Formats de Transport et des Combinaisons de Formats de Transport

Nous voulons maintenant paramétrer les exemples d'applications vus au chapitre précédent.

BS : les simulations de ce chapitre, ont montré que sur une très grande plage de  $E_b/N_0$ , les PDU de grandes tailles étaient à préférer. Nous allons donc sélectionner pour chacune de ces applications une taille de PDU dont le champ de donnée sera de 320 bits [42] [43]. Pour des applications à bas débits (en dessous de 32 kbit/s), des PDU de plus petites tailles pourront être utilisés (de l'ordre de débit en bit par seconde divisé par 100). [44]

### 3.5 Paramétrage des Formats de Transport et des Combinaisons de Formats de Transport 59

donne des exemples de tailles de PDU en fonction du débit pour des applications de type conversation audio.

Code correcteur : le code convolutif de taux 1/2 permet un transport fiable des informations sur un grand intervalle et est assez efficace, alors que les turbo codes et le code de taux 1/3 n'augmentent que de peu la plage de fiabilité, et sont en moyenne beaucoup moins efficace. L'option sans codage est très efficace lorsque l'état du canal est très bon, mais sa plage de fiabilité est trop faible. C'est donc le code convolutif de taux 1/2 que nous sélectionnerons pour toutes ces applications.

Service	BS	BSS	CRC	TTI	Codage
R2	320	0, 1	8 bits	10 ms	cc 1/2
R3	320	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	16 bits	10 ms	cc 1/2
R4	320	0, 1, 2, 3	16 bits	10 ms	cc 1/2
R5	320	0, 1, 2, 3, 4, 5	16 bits	10 ms	cc 1/2
R6	320	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	16 bits	10 ms	cc 1/2
R7	320	0, 1	24 bits	10 ms	cc 1/2
R8	320	0, 1, 2	24 bits	10 ms	cc 1/2
R9	320	0, 1, 2, 3, 4, 5, 6, 7, 8	24 bits	10 ms	cc 1/2

Table 3.3: Transport Format Set

Taille du CRC : la taille du CRC dépend directement du champ Residual BER du QoS IE.

Durée du TTI : le TTI est fixé à 10 ms pour des raisons techniques. Il ne nous a pas été possible d'utiliser d'autres valeurs de TTI pour nos expériences. Un TTI plus grand aurait permis un entrelacement inter TTI et aurait pu réduire les pertes, mais aurait accru les délais.

BSS : les différentes valeurs de BSS sont fixées en fonction du débit. Dans la table 3.3, le BSS est indiqué en nombre de blocs transporté par TTI. La couche MAC peut demander ces différents nombre de PDU en fonction notamment de ce qui est réellement présent au RLC. Le BSS zéro est toujours configuré. Puis nous augmentons de 1 jusqu'à atteindre le débit requis. Notons que pour les applications R3 à R6, nous avons dépassé de 1 la valeur requise. Cette surcapacité est créée dans le but de permettre des retransmissions sans faire diminuer le débit garanti. Nous ne le faisons pas pour R2, qui n'en effectue pas, ni pour R7 à R9 où il n'y a pas de débit

garanti. En augmentant de 1 le nombre de blocs transmis à chaque TF d'un TFS, nous obtenons ainsi la granularité maximum pour le débit. Il aurait été possible d'en supprimer quelques uns pour alléger la configuration et donc la signalisation au prix d'une baisse de la granularité.

La table 3.5 résume les ensembles de Format de Transport que nous préconisons pour des applications de références [3] citeillia1.

### 3.6 Conclusion

Nous avons vu dans ce chapitre, comment à partir du QoS IE, nous pouvions définir les Formats de Transports. Ainsi, simplement à partir de la signalisation Diffserv, nous avons été capables de proposer une méthode automatique qui assure la configuration de l'ensemble des applications multimédias. Nous avons également noté, que les paramétrages optimums dépendaient de l'état du canal. Les courbes d'efficacité se croisant, il est possible d'imaginer un mécanisme qui au lieu de proposer la configuration la plus efficace en moyenne, proposera une configuration évoluant dans le temps en fonction de l'état du canal. Ce mécanisme permettra d'accroître l'efficacité générale du système. C'est ce que nous verrons au prochain chapitre.

# Chapitre 4

---

## Lien adaptatif

---

### 4.1 Introduction

Nous avons noté que les courbes d'efficacité du chapitre précédent se coupaient les unes les autres. En fonction des intervalles de  $E_b/N_0$ , certains paramétrages sont donc plus efficaces. Comme une des caractéristiques du canal radio est de varier au cours du temps, des paramétrages qui étaient peu avantageux à un moment donné peuvent devenir les plus performants peu de temps après. L'objectif de ce chapitre va être de proposer un mécanisme adaptatif : un paramétrage sera proposé en fonction de l'état du canal. Dans un premier temps, nous allons examiner les différents mécanismes adaptatifs qui ont été proposés dans les réseaux sans fils. Nous proposerons ensuite un mécanisme de paramétrage adaptatif dans l'UMTS et nous calculerons le gain que peut apporter ce mécanisme par rapport à un paramétrage statique.

### 4.2 Aperçu des techniques de lien adaptatif dans les réseaux sans fils

#### 4.2.1 GPRS

Un système de lien adaptatif a été proposé en [48] pour GPRS (General Packet Radio Service). GPRS utilise le codage adaptatif. GPRS emploie des codes de taux variables. Les 4 codes sont les codes convolutifs de taux  $\frac{1}{2}$ ,

$\frac{2}{3}$  et  $\frac{3}{4}$ , et GPRS peut également transmettre sans utiliser de code correcteur (no coding).

Code correcteur	Payload	Max. Throughput
Convolutif 1/2	181	9.05
Convolutif 2/3	268	13.4
Convolutif 3/4	312	15.6
Sans codage (1)	428	21.4

Table 4.1: Codes correcteurs utilisés dans GPRS

La qualité du canal est estimée à partir du Block Error Rate (BLER).

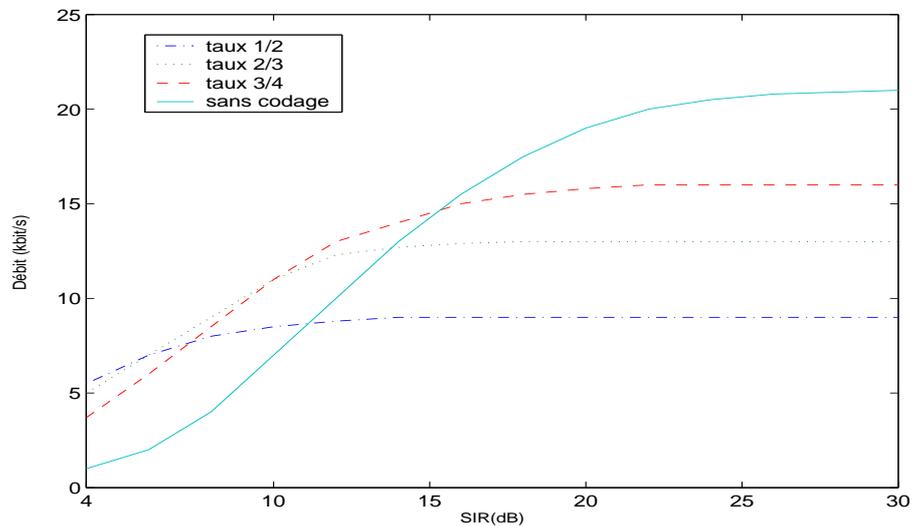


Figure 4.1: débit en fonction du code correcteur et de l'état du canal

Le débit  $D$  peut être calculé par :

$$D = (1 - BLER) * D_{max} \quad (4.1)$$

$$BLER = \left(1 - \frac{D}{D_{max}}\right) \quad (4.2)$$

La figure 4.2 montre le graphe des transitions pour les codes correcteurs, et les seuils de BLER pour ces transitions.

## 4.2 Aperçu des techniques de lien adaptatif dans les réseaux sans fils63

Code correcteur	Bad channel	Good channel
Convolutif 1/2	7.0 Kbps	8.7 Kbps
Convolutif 2/3	8.3 Kbps	12.3 Kbps
Convolutif 3/4	8.5 Kbps	13.6 Kbps
Sans codage	5.4 Kbps	13.7 Kbps
Adaptive coding	8 Kbps	14 Kbps

Table 4.2: Débit moyen dans GPRS

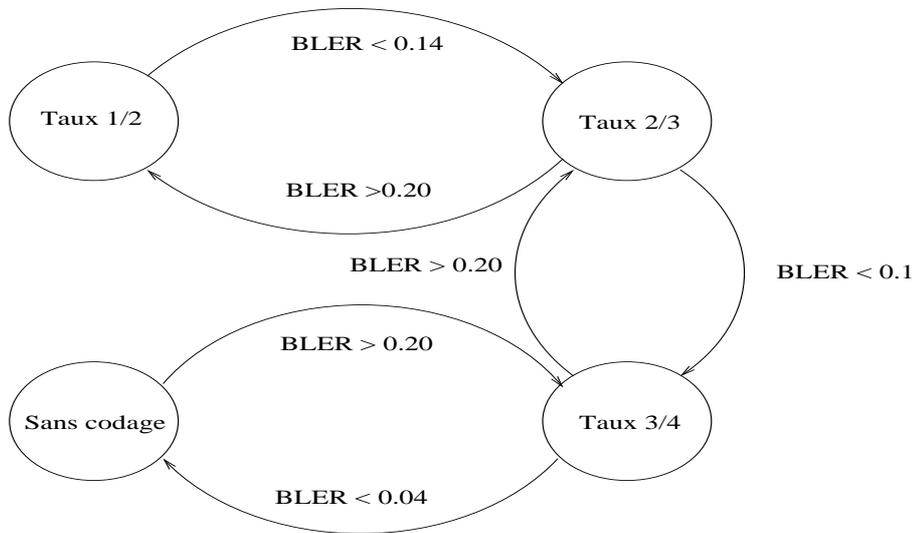


Figure 4.2: Graphe des Transitions pour le lien adaptatif dans GPRS

### 4.2.2 ATM sur satellite

Les canaux des réseaux satellites varient dans le temps en raisons d'évanouissement (fading) et d'interférences. Dans ATM pour satellite, [50] utilise un système hybride de FEC et ARQ. Il utilise les RCPC (Rate Compatible Punctured convolutional code). La figure 4.3 donne l'efficacité des RCPC code en fonction de leurs taux. Le code utilisé évolue en fonction du BER mesuré : lorsque le BER augmente au delà de certains seuils, le taux du RCPC code est diminué, lorsqu'il descend au-dessous de certains seuils, le taux du RCPC code est augmenté.

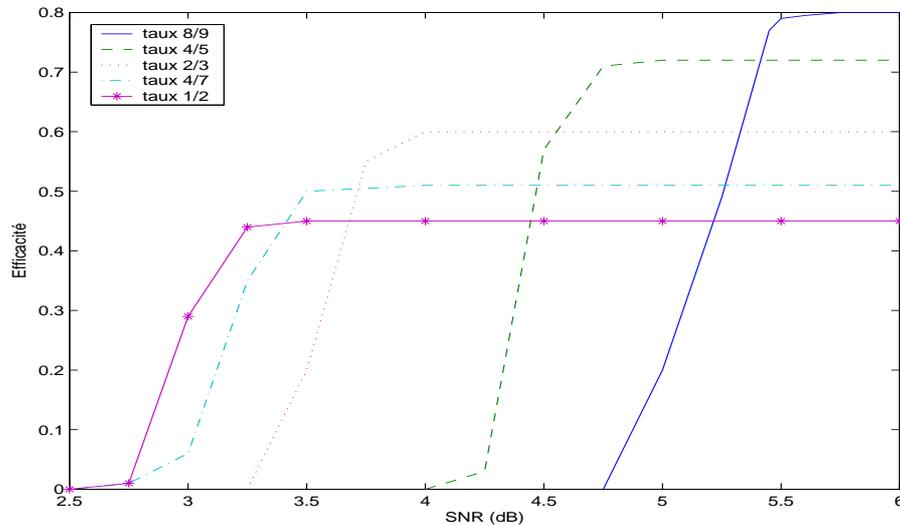


Figure 4.3: Efficacité des taux des RS code dans ATM sur Satellite

### 4.2.3 Hiperlan2

Orthogonal Frequency Division Multiplex (OFDM) est utilisé sur la couche radio d'hiperlan 2. [52] propose un système qui adapte la modulation et le taux de codage en fonction des conditions radios. La table 4.3 donne les différents types de modulation et les différents taux de codage possibles. L'adaptation est effectuée toutes les 20 ms avec pour objectif de maximiser le débit.

Mode	Modulation	Taux du code	débit (couche physique)
1	BPSK	1/2	6 Mbit/s
2	BPSK	3/4	9 Mbit/s
3	QPSK	1/2	12 Mbit/s
4	QPSK	3/4	18 Mbit/s
5	16QAM	9/16	27 Mbit/s
6	16QAM	3/4	36 Mbit/s
7	16QAM	3/4	54 Mbit/s

Table 4.3: Mode de la couche physique d'Hiperlan 2

La table 4.4 rappelle pour les systèmes que nous venons d'examiner, les méthodes utilisées pour estimer et prévoir le prochain état du canal, les

paramètres adaptés et la façon dont le changement de ces paramètre est signalé.

Système ou standard	Estimation de l'état du canal	Mécanismes	Signalisation
GPRS [45]	ACK et NACK message	codage adaptatif	use of separately coded stealing bits
ATM over satellite [50]	Measurements of the error rate carried in ARQ messages	Adaptive coding	
HIPERLAN 2 [52]	Interference Measurement	Adaptive modulation and coding	Physicals packet header

Table 4.4: Exemple de lien adaptatifs dans les réseaux mobiles

D'autres mécanismes de liens adaptatifs sont décrits en [46] [49] [51] [54] et [55].

### 4.3 Proposition de lien adaptatif dans l'UMTS

Le principal paramètre qui influe sur la fourniture du service et sur l'efficacité est le code correcteur. Le paramètre qui évoluera pour adapter le lien radio sera donc le code correcteur. Ce choix est conforme à ceux de [45], [50] et [53]. Le critère de sélection sera, parmi les paramétrages qui fournissent le service, celui qui a la plus grande efficacité.

Nous allons considérer deux types d'applications :

- les applications de type Non Real Time, qui demande une transmission sans perte, mais sans contraintes sur les délais ou la gigue.
- les applications de types Real Time, qui ont des contraintes strictes sur les délais, mais ont moins de contraintes sur les pertes.

Nous diviserons ces applications en deux sous groupes, un sous groupe avec des contraintes extrêmes sur les délais et un sous groupe permettant des contraintes moins importantes sur les délais. La table 4.5 décrit les contraintes que nous prendrons pour ces applications.

Applications	Non Real Time	RT Interactive	RT Non Interactive
SDU Error Ratio	minimum	$10^{-1}$	$10^{-2}$
Délais (ms)	N/A	100	250

Table 4.5: Exemples de critères de QoS

La figure 4.4 montre l'algorithme que nous utilisons pour le lien adaptatif.

1 Mesure du canal radio. Comme nous sommes en TDD, les mesures du canal uplink peuvent être utilisé pour le downlink et réciproquement.

2 Consultation de la QoS requise pour le service considéré.

3, 4 Consultation des tables des SDU ER et des délais. Sélection de la liste des paramétrages qui respectent la QoS requise.

5, 6 Consultation des tables d'efficacité. Sélection du paramétrage qui fournit la QoS le plus efficace.

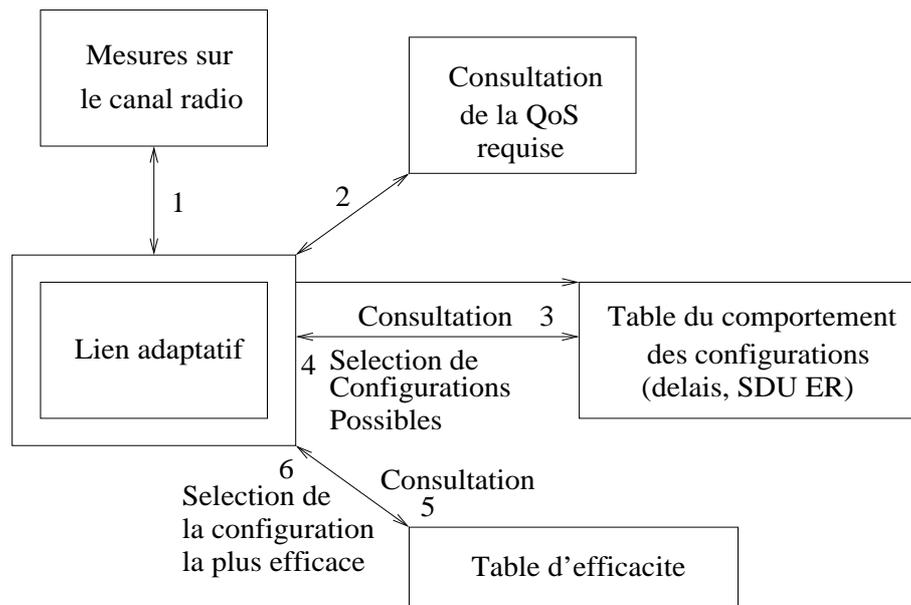


Figure 4.4: Mécanisme adaptatif

Les tables des SDU ER, délais et efficacité s'obtiennent à partir des résultats du chapitre précédent.

### 4.3.1 Evaluation de l'algorithme de lien adaptatif

Nous allons définir 4 scénarios pour comparer le mécanisme de lien adaptatif avec le schéma non adaptatif du chapitre précédent. Ces scénarios se différencient par le comportement du canal radio. La table 4.6 décrit ces scénarios. Dans le premier scénario,  $E_b/N_0$  varie de 0 à 3 dB. La qualité du canal est faible, et le canal est peut stable. Dans le second,  $E_b/N_0$  varie de 3 à 6 dB. La qualité du canal est moyenne et le canal est peut stable. Dans le troisième,  $E_b/N_0$  varie de 6 à 9 dB. La qualité du canal est bonne et le canal est peut stable. Dans le quatrième,  $E_b/N_0$  varie de 6 à 9 dB. La qualité du canal peut être faible ou bonne et le canal est très instable. La distribution de  $E_b/N_0$  est uniforme pour tous les scénarios.

Scenario	Intervalle de $E_b/N_0$	Stabilité	Qualité du canal radio
1	0 - 3 dB	assez instable	faible
2	3 - 6 dB	assez instable	moyen
3	6 - 9 dB	assez instable	bonne
4	0 - 9 dB	très instable	faible à bonne

Table 4.6: Caractéristiques des scénarios.

$$gain = 100 \left( \frac{\overline{eff_{ad}}}{\overline{eff_{st}}} - 1 \right) \quad (4.3)$$

Où  $\overline{eff_{ad}}$  et  $\overline{eff_{st}}$  sont les efficacités moyenne pour le codage adaptatif et le codage statique. Ces efficacités moyennes sont calculés par :

$$\overline{eff_{ad}} = \frac{\int_a^b eff(x) dx}{b - a} \quad (4.4)$$

Où  $eff()$  est la fonction efficacité d'un code et a et b sont les limites des intervalles considérés de  $E_b/N_0$ .

### 4.3.2 Application Non Real Time

Pour une application Non Real Time, nous devons fournir un SER minimum sans contrainte sur les délais. Un SER proche de zéro s'obtient en ne limitant pas le nombre de retransmission des PDU. Nous considérerons donc ici essentiellement les courbes d'efficacité. La courbe d'efficacité du lien adaptatif pour cette application Non Real Time s'obtient en prenant l'enveloppe

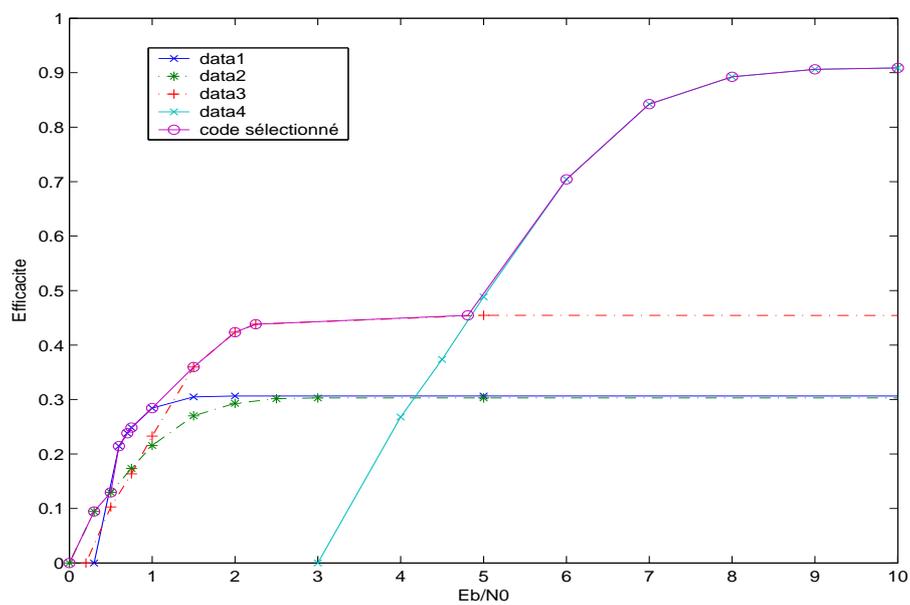


Figure 4.5: Code à sélectionner en fonction de l'application et de l'état du canal

supérieure des différentes courbes d'efficacité. La figure 4.5 montre cette enveloppe supérieur avec les différents codages.

De 0 à 0.5 dB, le code convolutif de taux 1/3 est sélectionné. De 0.5 à 1.3 dB, le turbo code est sélectionné. De 1.3 à 5 dB, le code convolutif de taux 1/2 est sélectionné. Au dessus de 5 dB, l'option sans codage est sélectionnée.

La figure 4.6 montre quel est le code à sélectionner en fonction de  $E_b/N_0$ . La table 4.7 donne pour chacun des scénarios proposés le gain fournit par le mécanisme adaptatif par rapport au paramétrage statique.

Dans le premier scénario, 3 codes correcteurs sont sélectionnables par le mécanisme adaptatif : les codes convolutifs de taux 1/2 et 1/3 et le turbo code. L'efficacité moyenne du codage adaptatif est de 0,32. Le gain apporté par rapport au code convolutif de taux 1/2, le plus efficace en moyenne sur cet intervalle, est de 10 %. Par rapport aux turbo code, le gain est même de 33.3 % et de 39.1 % pour le code convolutif de taux 1/3.

Dans le second scénario, 2 codes sont sélectionnables : le code convolutif de taux 1/2 et l'option sans codage. Les gains vont de 11.5 % à 32.1 % par rapport à un lien statique avec respectivement le code convolutif de taux 1/2 et l'option sans codage.

Dans le troisième scénario, l'option sans codage est toujours la meilleure solution. Le codage adaptatif n'apporte donc aucun gain dans ce scénario. On note que si des codes correcteurs étaient sélectionnés pour ce scénario, le gain serait alors de 86.7 %, 171 % et 180 % par rapport au respectivement code convolutif de taux 1/2, code convolutif de taux 1/3 et au turbo code.

Dans le quatrième scénario, les 4 codes sont sélectionnables. Le codage adaptatif permet un gain de 35.9 % par rapport au meilleur code en moyenne (option sans codage). Les gains vont jusqu'à 39.4 %, 94.1% et 99.9% par rapport au respectivement code convolutif de taux 1/2, code convolutif de taux 1/3 et au turbo code.

	Scénario	1	2	3	4
Addaptatif	eff	0.32	0.50	0.84	0.55
cc 1/3	gain	39.1%	67.3%	180%	99.9%
turbo	gain	33.3%	61.9%	171%	94.1%
cc 1/2	gain	10.3%	11.5%	86.7%	39.4%
sans codage	gain	$\infty$	32.1%	0.00%	35.9%

Table 4.7: Gain du codage adaptatif pour une application Non Real Time

### 4.3.3 Application Real Time Interactive

Nous considérons ici une application Real Time Interactive qui demande des délais tel que 90 % de ses paquets traversent l'UTRAN en moins de 100 ms. D'après les courbes d'efficacité et de délais, il faut un  $E_b/N_0$  supérieur à 0.6 dB pour que le turbo code respecte cette contrainte. Avec le code convolutif de taux 1/3, il faut un  $E_b/N_0$  supérieur à 0.9 dB, avec le code convolutif de taux 1/2, il faut un  $E_b/N_0$  supérieur à 1.4 dB. Sans codage,  $E_b/N_0$  doit être supérieur à 5.9 dB. La figure 4.6 indique le type de codage à sélectionner en fonction de  $E_b/N_0$ .

De 0 à 0.6 dB, aucune configuration ne permet de fournir le service.

De 0.6 à 0.9 dB, seul le turbo code fournit le service. Comme, il est le seul sélectionnable, il est automatiquement sélectionné.

De 0.9 à 1.4 dB, le turbo code et le code convolutif de taux 1/3 sont capable de fournir la QoS requise. Le turbo code étant cependant toujours le plus efficace sur cet intervalle, il sera sélectionné.

De 1.4 à 5.9 dB, les 3 codes correcteurs peuvent fournir la QoS requise. C'est le code convolutif étant le plus efficace sur cet intervalle, il sera sélectionné.

A partir de 5.9 dB, toutes les options de codage peuvent être utilisées. C'est l'option sans codage qui donne la plus grande efficacité, elle sera sélectionnée.

Nous notons que la plage d'utilisation du turbo code est très courte, et que le code convolutif de taux 1/3 n'est pas utilisé.

	Scénario	1	2	3	4
Addaptatif	eff	0.30	0.45	0.84	0.53
turbo	gain	27.0%	46.6%	171%	87.7%
cc 1/2	gain	17.0%	0.00%	84.8%	40.8%
sans codage	gain	$\infty$	$\infty$	0.00%	89.0%

Table 4.8: Gain du codage adaptatif pour une application Real Time Interactive

#### 4.3.4 Application Real Time Non Interactive

Nous considérons ici une application Real Time Non Interactive qui demande des délais tel que 99 % de ses paquets traversent l'UTRAN en moins de 250 ms. D'après les courbes d'efficacité et de délais il faut sélectionner les codes suivants :

De 0.6 à 1 dB, seul le turbo code et le code convolutif de taux 1.3 peuvent fournir le service. Le turbo code est toujours le plus efficace sur cet intervalle, c'est donc celui qui sera sélectionné.

De 1 à 1.3 dB, le turbo code et le code convolutif de taux 1/2 et 1/3 sont capable de fournir la QoS requise. Mais, le turbo code est toujours le plus efficace sur cet intervalle. C'est donc le code qui sera sélectionné.

De 1.3 à 5 dB, les 3 mêmes codes peuvent fournir la QoS. Mais c'est le code convolutif de taux 1/2 qui est le plus efficace sur cet intervalle.

A partir de 5.0 dB, toutes les options de codage peuvent être utilisées, et c'est l'option sans codage qui donne la plus grande efficacité.

Nous notons que la plage d'utilisation du turbo code est très courte, et que le code convolutif de taux 1/3 n'est pas utilisé.

	Scénario	1	2	3	4
Addaptatif	eff	0.30	0.50	0.84	0.55
turbo	gain	27.0%	61.6%	171%	94.3%
cc 1/2	gain	13.8%	11.3%	86.7%	41.6%
sans codage	gain	$\infty$	152%	0.00%	57.6%

Table 4.9: Gain du codage adaptatif pour une application Real Time Non Interactive

La figure 4.6 indique les codes à sélectionner en fonction de  $E_b/N_0$  et de l'application considérée. L'application 1 est l'application NRT, l'application 2 est l'application RT ineractive, et l'application 3 est l'application RT non interactive.

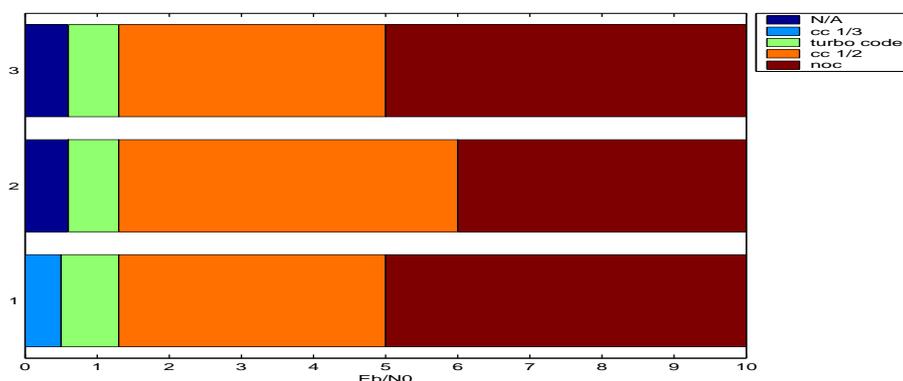


Figure 4.6: Code à sélectionner en fonction de l'application et de l'état du canal

#### 4.4 Conclusion

Dans ce chapitre, nous avons montré le gain que peut apporter un lien adaptatif. Ce gain est particulièrement dépendant de l'amplitude de la variation de l'état du canal. Notons que l'état du canal dépend également du nombre d'utilisateurs. Or, jusqu'à présent nous n'avons traité que des utilisateurs pris séparément, sans tenir compte de la puissance.

Dans les chapitres suivants, nous allons considérer un environnement à utilisateurs multiples. Nous prendrons en compte la puissance d'émission des utilisateurs. Nous distinguerons les services Real Time des services Non Real Time. En particulier, dans le prochain chapitre nous considérerons le contrôle d'accès et l'allocation de ressources pour les applications Real Time.

## Chapitre 5

---

# Contrôle d'accès et Allocation de Ressources

---

### 5.1 Introduction

Nous prendrons maintenant en compte un environnement à plusieurs utilisateurs. Ces différents utilisateurs vont se partager la ressource radio mais également interférer les uns avec les autres. Dans certaines circonstances, l'ajout d'un nouvel utilisateur peut dégrader de façons importantes les applications en cours. Il peut donc être plus avantageux de ne pas autoriser cet utilisateur à transmettre ces données. Deux cas peuvent alors être considérés :

- l'utilisateur ne demande pas un débit garanti et ses paquets peuvent être mis en attente. Nous parlons dans ce cas d'application Non Temps Réels (NRT, Non Real Time).

- l'utilisateur demande un débit garanti, et ses données ne supportent pas un long délai d'attente. Ce sont les applications de type Temps Réels (RT, Real Time) [58]. Pendant l'établissement de la configuration d'une application RT, l'entité Radio Resource Management (RRM) vérifiera si il est possible de lui fournir son service. On appelle cette étape le contrôle d'accès (CAC, Call Access Control). Si on ne peut leur fournir le service

qu'elle demande, une application RT est rejetée.

Dans ce chapitre, nous nous intéresserons au contrôle d'accès et à l'allocation de ressources pour les applications de types temps réels. Nous commencerons par définir la notion de ressources dans l'UMTS-TDD. Nous présenterons les différents Canaux de Transport (TRCH, Transport Channel) et indiquerons celui qui est le plus adapté pour le transport des applications de type Real Time. Nous présenterons ensuite différents algorithmes d'allocation de ressource que nous avons joints au contrôle d'accès. Nous indiquerons ainsi en une seule étape si un utilisateur peut avoir accès au service et quelles ressources lui sont attribuées. Ces algorithmes tiennent compte de la puissance d'émission des utilisateurs et refusent l'accès à des applications demandant un service si celui-ci devait dégrader ceux des utilisateurs en cours. Ces algorithmes seront comparés en termes de répartition de la puissance entre les slots et taux de rejet des appels.

## 5.2 Notion de Ressources Radio

Pour aborder la notion de ressources radio, nous commencerons par développer la notion d'accès multiples. L'UMTS-TDD utilisent simultanément deux méthodes de multiplexages : un multiplexage par code, le WCDMA et un multiplexage temporel, le TDD.

### 5.2.1 Principe du WCDMA

L'UMTS utilise la technique d'accès WCDMA. Le WCDMA est un système d'accès multiple par répartition de code [59]. Les bits correspondants aux données utilisateurs sont étalés sur une large bande passante en multipliant ces données par une séquence pseudo aléatoire de bits (appelés chips) provenant des codes d'étalement CDMA.

En UMTS-TDD, le débit chip (débit binaire de la séquence de codage) est de 3.84 Mc/s. En une seconde il y a 100 trames de 10 ms. Chaque Trame est elle-même divisé en 15 slot de 667  $\mu$ s. Il y a donc 2560 chips par slot. La figure 5.1 montre la structure d'une trame radio.

La figure 5.2 décrit le fonctionnement de base de l'étalement et du désétalement d'un système CDMA.

Dans cet exemple, le signal initial à une modulation initial BPSK de fréquence R. Ce signal est donc composé d'une séquence de bits valant +1 et -1. La méthode d'étalement consiste à multiplier chaque bit du signal initial par une séquence de huit bits, ces huit bits constituent un code orthogonal (Orthogonal Variable Spreading Factor, OVSF). Le résultat obtenu est un

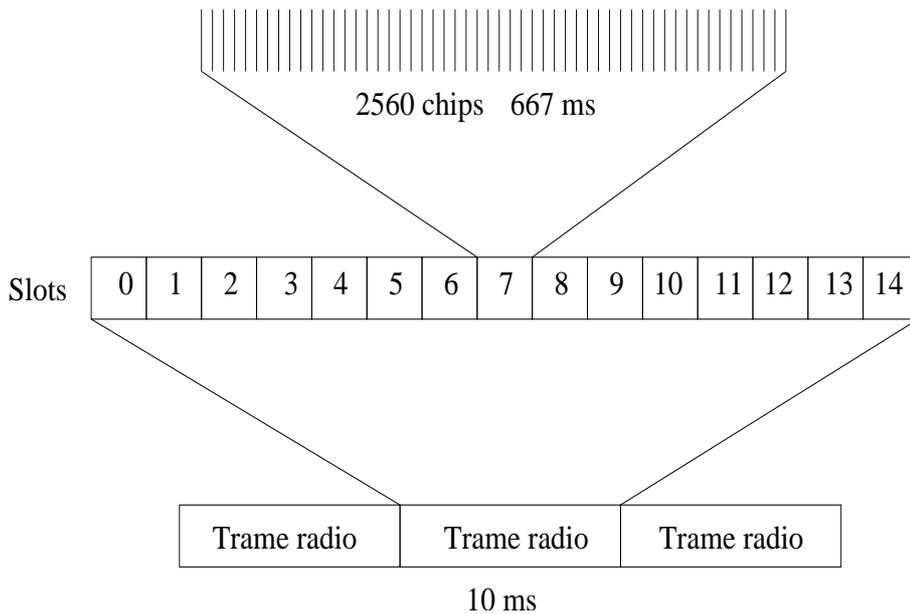


Figure 5.1: Structure d'une Trame Radio

signal de fréquence  $8 \cdot R$ . Le facteur d'étalement (Spreading Factor, SF) est donc de 8. Ce signal sera transmis sur l'interface air.

Pour désétalement, le signal reçu est multiplié par la même séquence de huit bits qui avait servi à étaler les données. Le signal après désétalement est le même qu'avant l'étalement.

Puis, le récepteur somme le produit des bits du signal reçu. La partie inférieure du schéma montre l'effet du désétalement quand il est appliqué à un signal pour lequel l'étalement a été effectué avec une autre séquence d'étalement. Les sommes obtenues sont proches de zéro.

L'étalement et le désétalement permettent en moyenne d'augmenter l'amplitude du signal d'un facteur 8 par rapport aux autres signaux qui interfèrent. Cet effet est appelé gain de traitement (processing gain).

La figure 5.3 montre l'effet du désétalement quand il est appliqué à un signal étalé avec un code orthogonal différent de celui utilisé pour l'étalement. La sommation donne des valeurs proches de zéro.

L'orthogonalité des codes permet à plusieurs utilisateurs d'émettre ou de recevoir simultanément des données sur la même porteuse. Les données de chaque utilisateur seront codées avec des codes orthogonaux différents. Le nombre d'utilisateurs pouvant émettre simultanément dépend des différentes

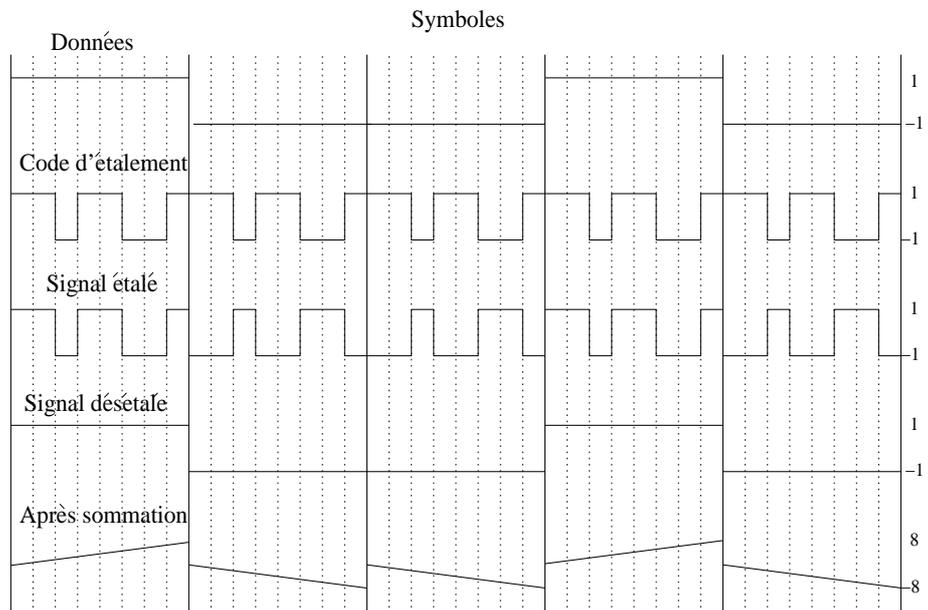


Figure 5.2: Etalement et désétalement en CDMA

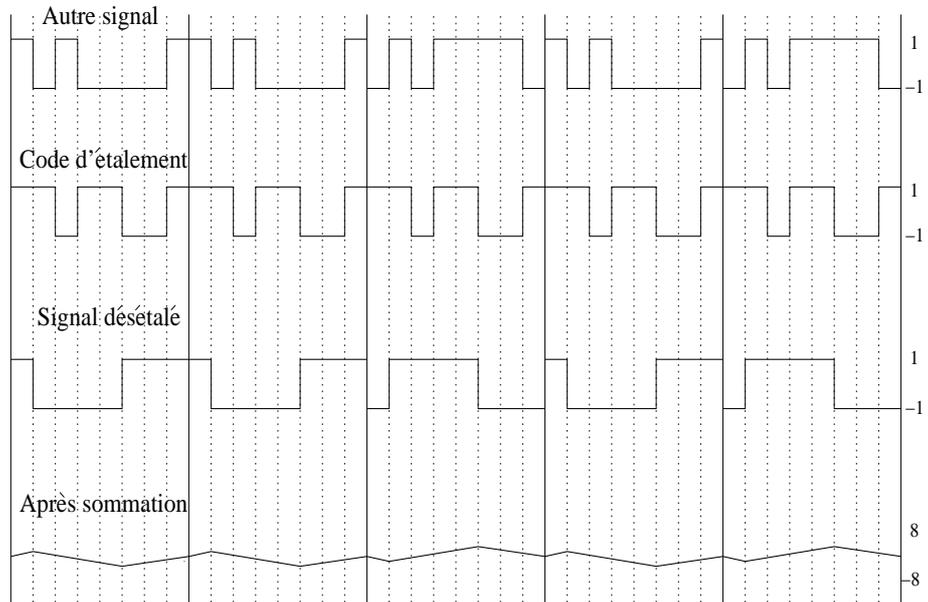


Figure 5.3: Désétalement avec un OVSF qui ne coresspond pas.

valeurs du Facteur d'étalement. En downlink, le Facteur d'étalement est de 1 ou 16, en uplink, il peut être de 1, 2, 4, 8 ou 16.

La figure 5.4 représente l'arbre des OVSF.

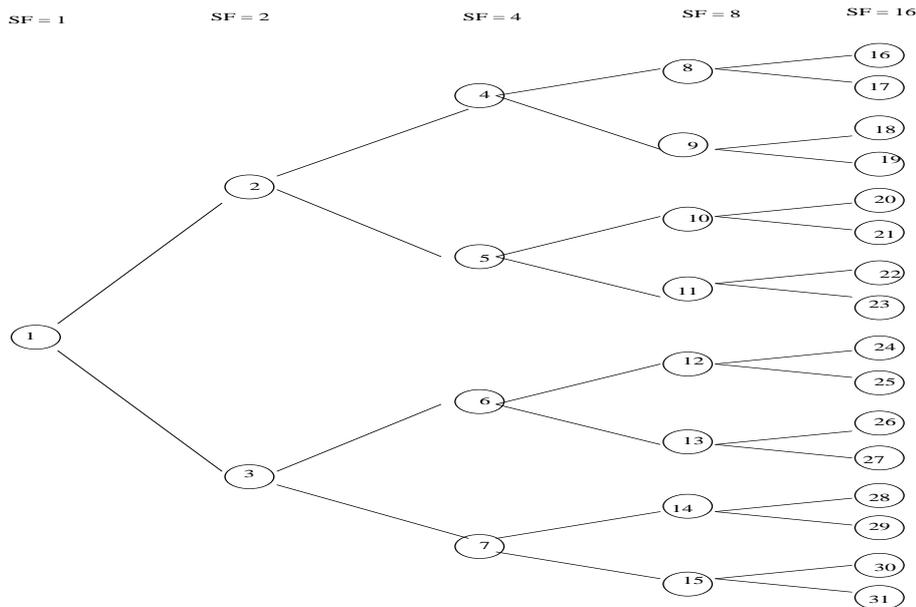


Figure 5.4: Arbre des OVSF codes.

Chaque nœud ou feuille représente un OVSF différent. La valeur du facteur d'étalement d'un code est donnée par sa profondeur dans l'arbre. Le Facteur d'étalement de la racine est de 1, celui des feuilles est de 16. Quand un code est utilisé, tous les autres codes parents ou descendants dans l'arbre ne peuvent plus être utilisés. Ainsi, si le code 4 est utilisé, les codes 1, 2, 8, 9 et 16 à 23 ne peuvent être utilisés.

Si tous les utilisateurs utilisent un SF de 16, alors 16 utilisateurs peuvent émettre (ou recevoir) simultanément. Par contre, en utilisant un SF de 16, alors il faut 16 chips pour coder un bit, alors qu'il n'en faut que 1 seul si le SF est de 1. Un même utilisateur peut utiliser simultanément plusieurs codes orthogonaux, et donc recevoir simultanément plusieurs signaux.

### 5.2.2 Burst Format

La façon dont sont agencées les données à l'intérieur d'un slot est décrite dans les Bursts Format [60]. Il y a 3 types de bursts qui diffèrent par la taille de leurs champs de données, de leurs séquences d'apprentissage et de leur période de garde. La table 5.1 décrit la structure des bursts en UMTS TDD.

Nom	Champ de données I	Séquence d'apprentissage	Champ de données II	Période de garde
Burst Type I	976 chips	512 chips	976 chips	96 chips
Burst Type II	1104 chips	256 chips	1104 chips	96 chips
BBurst Type III	976 chips	512 chips	880 chips	192 chips

Table 5.1: Structure des bursts

Les bursts de type I et II s'utilisent en downlink et en uplink, alors que le burst de type III ne s'utilise que en uplink.

La séquence d'apprentissage, ou midamble, est une séquence de bit connue par le récepteur. Cela permet d'estimer le canal.

La période de garde est nécessaire en TDD en raison de la transmission discontinue (passage de downlink à l'uplink). Le changement de direction de transmission nécessite un espacement temporel et doit être scrupuleusement contrôlé. Les transmissions dans les deux sens doivent être synchronisées afin d'éviter les collisions dues à d'éventuels retards dans la propagation. En downlink, le Spreading Factor est de 1 ou 16, le nombre de symbole par burst est donc égale au nombre de chips divisé par le SF. La modulation est QPSK [61] (c'est-à-dire qu'un symbole peut prendre 4 valeurs complexes :  $1+i$ ,  $1-i$ ,  $-1-i$ ,  $-1+i$ ). Un symbole peut donc transporter 2 bits (la modulation peut également être 8PSK dans certains cas).

Un champ de données peut contenir un champ TFCI. Le TFCI permet d'indiquer à un mobile quels sont les slots qu'il doit écouter et avec quels OVSF. La taille des TFCI peut être de 0, 4, 8, 16 ou 32 bits.

Il y a donc 20 types de burst possibles en downlink, selon les combinaisons de SF (2 possibilités), de type de burst (2 possibilités) et de taille de TFCI (5 possibilités).

En uplink, le Spreading Factor peut prendre 5 valeurs, il y a 3 types

de bursts, 5 tailles de TFCI et il y a également la possibilité d'inclure un champ TPC (Transmission Power Control). Le TPC sert à indiquer s'il faut accroître ou diminuer la puissance de transmission. La taille du TPC est de 0 ou 2 bits. 3GPP ne définit cependant que 90 types de bursts parmi les 150 possibles [60].

### 5.2.3 Time Division Duplex

Deux méthodes de transmission duplex sont utilisées dans l'UMTS : le FDD (Frequency Division Duplex) et le TDD (Time Division Duplex). La méthode FDD utilise des bandes de fréquences séparées pour le sens montant et le sens descendant. La méthode TDD utilise la même bande de fréquence dans les deux sens de transmission, mais pas au même moment. Certains slots sont dit downlink. Pendant ces slots, les données ne peuvent aller que de la Station de Base vers le Mobile. D'autres slots sont dit uplink. Pendant ces slots, les données ne peuvent aller que du Mobile vers la Station de Base.

En TDD, en cas de trafic asymétrique, il est possible d'allouer un nombre supérieur de slots à un sens qu'à un autre [66].

### 5.2.4 Notion de ressource

Un OVSF code utilisé à un slot donné peut être appelé une ressource [62]. Pour un SF fixé à 16, (valeur qui permet la plus grande granularité dans l'allocation des ressources), la figure 5.5 représente l'ensemble des ressources disponibles pendant un TTI. Cette définition correspond également à celle d'un canal physique.

Une ressource représente la quantité minimale de symboles transmissibles à un utilisateur. Ses données seront toujours transportées sur un nombre entier de ressources. Pour le transport de donnée, une ressource ne pourra pas être utilisée simultanément par plusieurs utilisateurs.

## 5.3 Canaux de Transport

Un Canal de Transport (TRCH, Transport Channel) est un service offert par la couche physique à la couche MAC pour le transfert d'information. Il est défini par la manière et les caractéristiques selon lesquelles les données sont transférées sur l'interface radio. Les canaux de transport peuvent être classés en deux catégories : les canaux communs et les canaux dédiés.

OVSF 15															
OVSF 14															
OVSF 13															
OVSF 12															
OVSF 11															
OVSF 10															
OVSF 9															
OVSF 8															
OVSF 7															
OVSF 6															
OVSF 5															
OVSF 4															
OVSF 3															
OVSF 2															
OVSF 1															
OVSF 0															
Slot	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Figure 5.5: Ressource en UMTS TDD, SF = 16

### 5.3.1 Dedicated Channel

Le Dedicated Channel (DCH), ou canal dédié, transporte des données ou des informations de contrôle d'un utilisateur donné. Il y a des DCH dans le sens montant et des DCH dans le sens descendant.

Les canaux communs : six types de canaux de transport communs ont été définis dans l'UTRA TDD. Les canaux communs sont des ressources partagées par différents utilisateurs d'une même cellule.

### 5.3.2 Broadcast Channel

Le Broadcast Channel (BCH) est un canal de transport descendant utilisé pour transmettre des informations spécifiques au réseau d'accès ou à une cellule donnée. Le terminal ne peut s'inscrire dans une cellule que si il peut décoder le BCH. Le BCH émet en général à une puissance élevée pour qu'il puisse être capté par l'ensemble des mobiles de la cellule. Son débit est également faible pour que même les mobiles de faible capacité puissent le décoder.

### 5.3.3 Forward Access Channel

Le Forward Access Channel (FACH) est un canal de transport descendant qui transmet des informations de contrôle aux terminaux localisés dans une cellule donnée. Il est également possible de transmettre de petite quantité de données. Il peut y avoir plusieurs FACH dans une même cellule, mais l'un d'eux doit cependant avoir un faible débit pour pouvoir être écouté par tous les mobiles de la cellule. Les messages transmis doivent comprendre des informations d'identification pour être correctement interprétés par leurs destinataires.

### 5.3.4 Paging Channel

Le Paging Channel (PCH) est un canal de transport descendant servant à initier une communication avec un terminal.

### 5.3.5 Random Access Channel

Le Random Access Channel (RACH) est un canal de transport montant qui est utilisé pour transporter des informations de contrôle provenant d'un mobile telles que les demandes d'établissement de connexion. Il peut également être utilisé pour transmettre de petites quantités de données.

### 5.3.6 Downlink Shared Channel

Les Downlink Shared Channel (DSCH) sont des canaux de transport de sens descendant. Ils peuvent transporter des données ou des informations de contrôles. Ils peuvent être attribués d'une trame à l'autre à différents utilisateurs. Lors de la configuration d'un DSCH, le RRM lui attribue un certain nombre de canaux physiques. Le premier canal physique du premier slot d'un DSCH contient un TFCI. C'est en lisant le TFCI qu'un utilisateur sait si le DSCH lui est attribué pour ce TTI. Le MAC sélectionne parmi les différents flux celui qui sera transmis sur un DSCH en fonction de différents critères (priorités, occupation des buffers (RLC)) indiqué par le RRM. Ces mécanismes seront détaillés dans le chapitre 6.

### 5.3.7 Uplink Shared Channel

Les Uplink Shared Channel (USCH) sont semblables aux DSCH, mais dans le sens montant.

## 5.4 Coded Composite Transport Channel

Le CCTRCH (Coded Composite Transport Channel) est une structure qui indique et décrit les canaux physiques attribués à un utilisateur. Le CCTRCH lui est donné à l'établissement de sa connexion. Le CCTRCH contient une liste de canaux physiques (slot, OVSF), et pour chaque canal physique le numéro de burst qui sera utilisé (pour connaître la taille et la position des champs de données). Le premier canal physique d'un CCTRCH contient un TFCI. Il n'y a qu'un seul TFCI par CCTRCH. Si le TRCH est un DSCH, alors le TFCI indique l'identifiant du mobile à qui il est destiné. Si le TRCH est ou sont des DCH (les DCH d'un même utilisateur peuvent être multiplexés sur un même CCTRCH), alors le TFCI indique le numéro du Transport Format Combination utilisé.

## 5.5 Contrôles d'accès

Si la charge de l'interface aire pour une cellule donnée augmente excessivement, la qualité de service des connexions en cours ne pourra être garantie. Avant d'accepter une nouvelle connexion, le contrôle d'accès (ou contrôle d'admission) doit vérifier si celle-ci ne va pas dégrader la Qualité de Service des connexions en cours. L'algorithme de contrôle d'accès est exécuté lorsque la configuration d'un service RT est créé ou lorsque la configuration d'un service RT est modifié. Le contrôle d'accès est réalisé de façon indépendante dans le sens montant et le sens descendant. Les limites d'admission d'une cellule sont fixées durant la phase de planification du réseau. Le contrôle d'admission se fonde souvent en downlink sur la puissance d'émission [72] [73], en uplink sur la puissance reçue par la Station de Base, ou sur les niveaux d'interférences des utilisateurs [74] [65].

Les DCH sont créés au moment de la configuration d'un service. Ils leur sont attribués un certain nombre de ressources en fonction des besoins de l'application transportée. Ces ressources sont réservées à un utilisateur pour la durée de son service. Au contraire, les DSCH sont créés à l'initialisation de la Station de Base (il est cependant possible de les reconfigurer périodiquement). Les ressources d'un DSCH peuvent ne pas correspondre au besoin d'un service. Il peut ne pas avoir assez de ressources, et il faudra alors plusieurs DSCH pour transporter les données d'un utilisateur, ou bien il peut avoir trop de ressources, auquel cas, il y aura un certain gaspillage. De plus les ressources d'un DSCH ne sont pas réservées à un utilisateur. Un autre pourra utiliser ces ressources le TTI d'après.

Les applications Temps Réels utiliseront donc les DCH : eux seuls peuvent garantir le débit, et ils évitent un gaspillage de ressource en ayant un dimensionnement adapté [70] [71].

L'algorithme de contrôle d'accès downlink développé au cours de cette thèse est couplé avec l'algorithme d'allocation de ressources des DCH.

Il comprend deux étapes de base : 1) déterminer le nombre  $N$  de ressources nécessaires au service entrant

2) trouver une combinaison de  $N$  ressources libres dont l'utilisation ne fait pas dépasser le seuil de puissance des slots concernés. La façon de calculer le nombre de ressources d'un service est donnée en annexe. Les paragraphes suivants vont comparer différents algorithmes chargés de définir ces combinaisons.

## 5.6 Algorithmes d'allocation

Quatre algorithmes d'allocation ont été comparés. Ces algorithmes allouent des ressources dédiées, et donc, configurent des DCH.

### 5.6.1 Max in slot

Cet algorithme alloue les ressources à partir du premier slot. Il remplit le slot de bas en haut, jusqu'à ce que toutes les ressources soient utilisées ou jusqu'à ce que le seuil de puissance soit atteint. Le premier slot rempli, il passe au second et ainsi de suite.

### 5.6.2 Min in slot 1

Cet algorithme alloue les ressources à partir du premier code orthogonal (OVSF), et continue d'allouer ce code sur des slots différents, jusqu'à ce que ce code soit utilisé sur tous les slots. Il utilise ensuite l'OVSF suivant. Si l'utilisation d'un code fait dépasser le seuil de puissance d'un slot, le code n'est pas utilisé sur ce slot et l'on passe au slot suivant.

### 5.6.3 Min in slot 2

Cet algorithme compte le nombre de ressources attribuées par slot. Il alloue le premier OVSF libre du slot qui a le moins de ressources utilisées. (si plusieurs slots ont le même nombre de ressource, on commence par allouer des ressources dans le slot le plus à gauche). Si l'utilisation d'un code fait dépasser le seuil de puissance d'un slot, le code n'est pas utilisé et l'on alloue

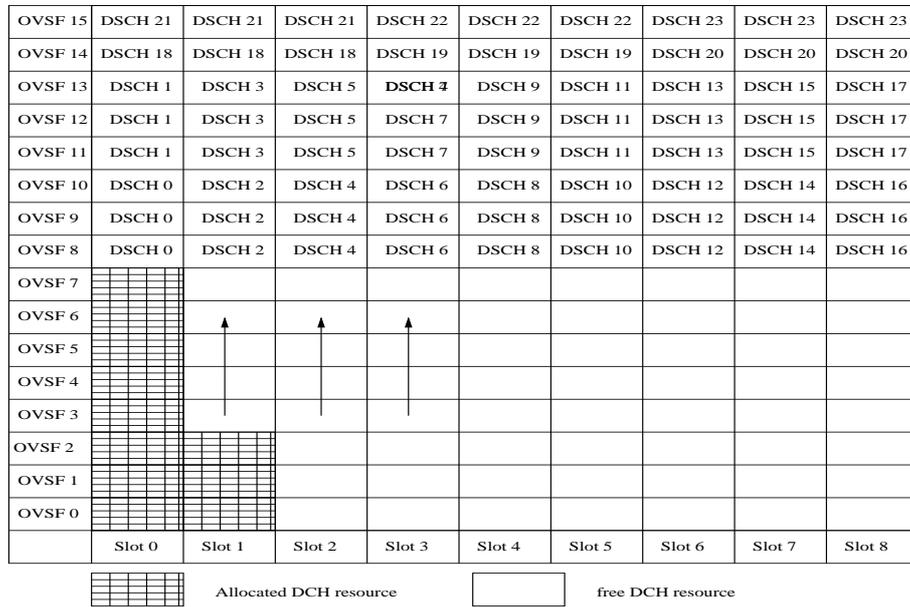


Figure 5.6: Max in slot

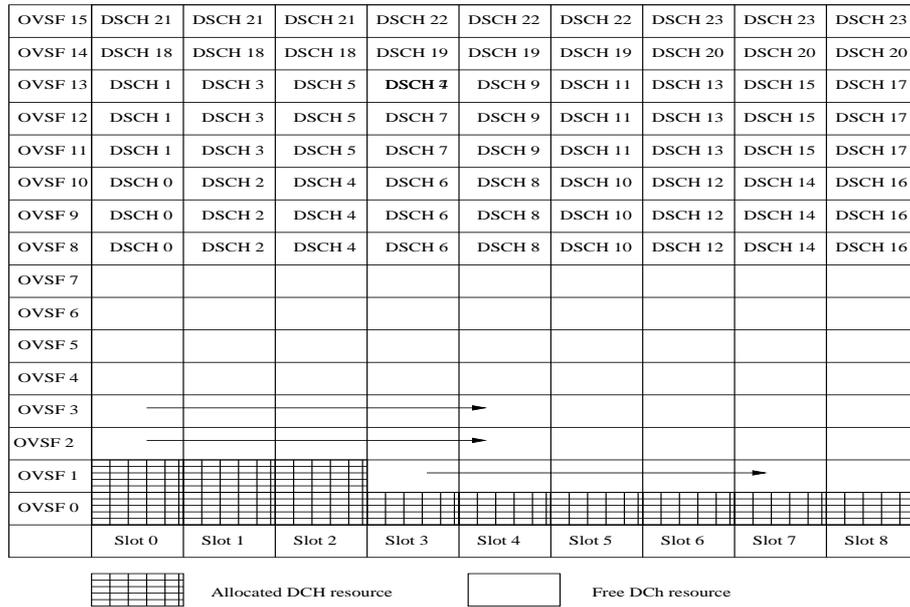


Figure 5.7: Min in slot 1

une ressource du slot suivant qui a le moins de ressources utilisés, et ainsi de suite.

### 5.6.4 Power Based

Cet algorithme mesure la puissance d'émission de chaque slot (somme des puissances d'émission de chaque ressource du slot). Si toutes les ressources du slot qui a la puissance d'émission la plus faible sont utilisées, alors on alloue les ressources du second slot qui a la puissance d'émission la plus faible, et ainsi de suite. Ces algorithmes sont utilisés jusqu'à ce que le

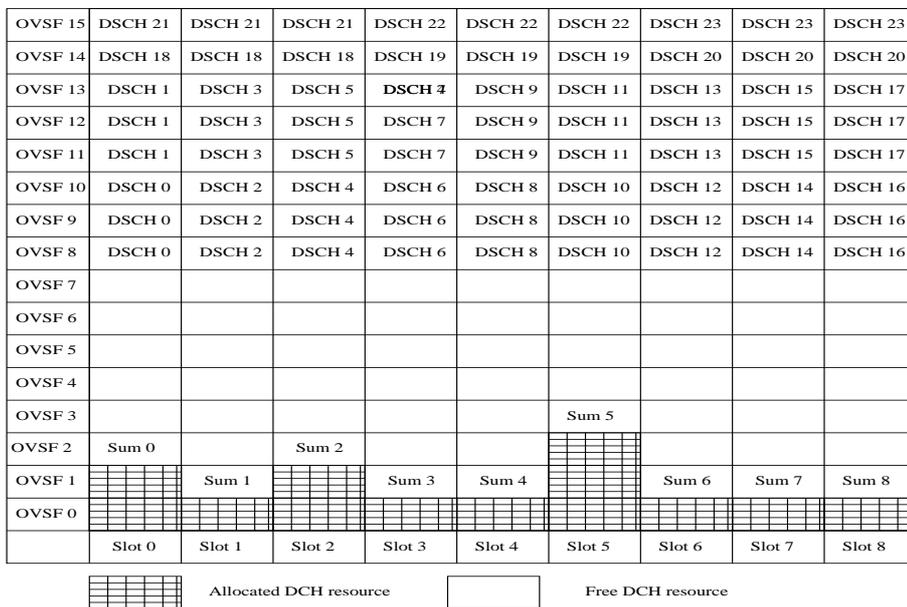


Figure 5.8: Power Based

nombre de ressource demandé par le service ait été alloué. S'il n'a pas été possible de trouver suffisamment de ressources, le service est rejeté, et toutes les ressources qui lui avaient été allouées sont libérées.

Notons qu'aucun de ces algorithmes ne permet une allocation qui face dépasser le seuil de puissance d'un slot.

La figure 5.9 indique l'ordre dans lequel les ressources d'un utilisateur sont allouées en fonction de l'algorithme choisi. L'utilisateur demande 4 ressources d'une puissance normalisée de 0.15. Les ressources déjà allouées ont une puissance normalisée de 0,2.

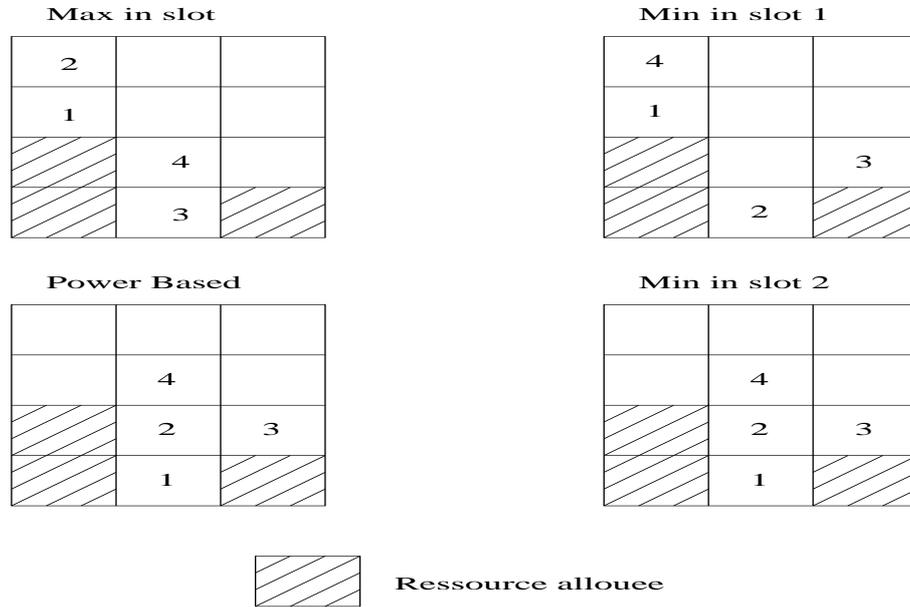


Figure 5.9: Les ressources allouées par les différents algorithmes.

## 5.7 Comparaisons des algorithmes

Nous avons simulé le comportement de ces quatre algorithmes : Nous allons comparer la répartition de la puissance en fonction des algorithmes, et les taux de rejets des utilisateurs.

Des utilisateurs arrivent avec une distribution des temps entre deux arrivées qui suit une loi exponentielle de taux  $\mu$  et un temps de service qui suit une loi exponentielle de taux  $\lambda$ . Pour chaque expérience le nombre de ressources consommé par utilisateurs est noté  $N$ . 9 slots ont été alloués au sens downlink. La table 5.2 indique les différentes valeurs utilisés pour  $\lambda$ ,  $\mu$  et  $N$  au cours des expériences.

Charge	forte	moyenne	faible
$1/\lambda$	15	20	30
$1/\mu$	120	120	120
$N$	12	12	12

Table 5.2: durée moyenne des interarrivées et des services

Les figures 5.10 à 5.12 montrent la puissance moyenne au cours des expériences en fonction des algorithmes.

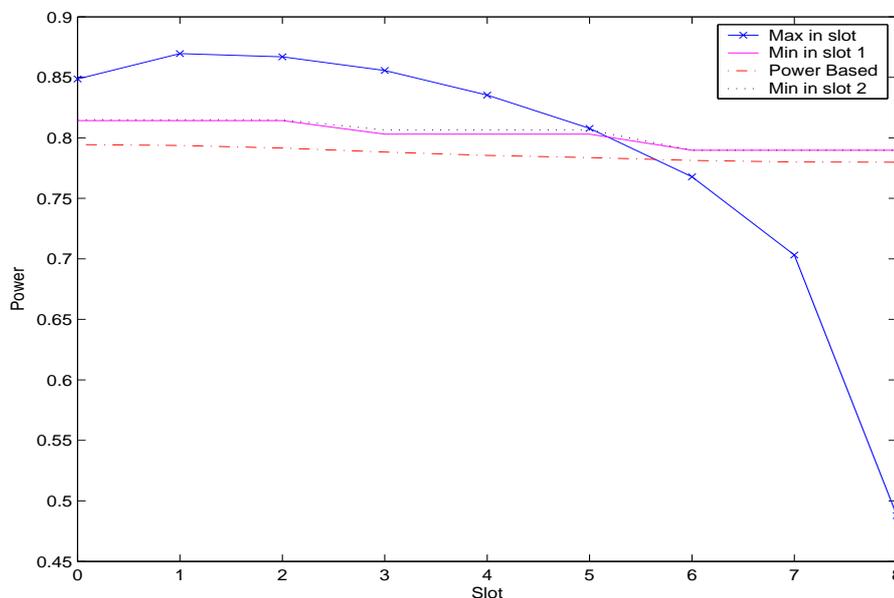


Figure 5.10: puissance moyenne par slot, charge forte

Nous observons que l'algorithme Max in slot induit un gros déséquilibre entre les premiers et les derniers slots d'une trame. Les algorithmes Min in slot 1 et 2 sont beaucoup plus équilibrés, la puissance moyenne oscillant dans un intervalle inférieur à 0.03. L'algorithme Power Based est celui dont la puissance est le mieux répartie. L'écart entre la plus grande et la plus faible moyenne est inférieur à 0.015.

La fonction de répartition (Cumulative Distribution Function, CDF)  $F(x)$  de la variable aléatoire  $X$  se définit par :

$$F(X) = P(X \leq x) \quad (5.1)$$

La CDF des puissances des différents algorithmes est présentée en figure 5.13. Nous notons que la CDF de max in slot ne vaut pas zéro en  $x =$  zéro (mais vaut bien zéro dès que  $x < 0$ , donc respecte la condition de la limite en  $-\infty$  égale à zéro). Cela est dû au fait que la distribution de la puissance de max in slot n'est pas centrée sur la moyenne. Au cours des expériences, 4 % des slots ne sont pas utilisés. Les trois autres algorithmes

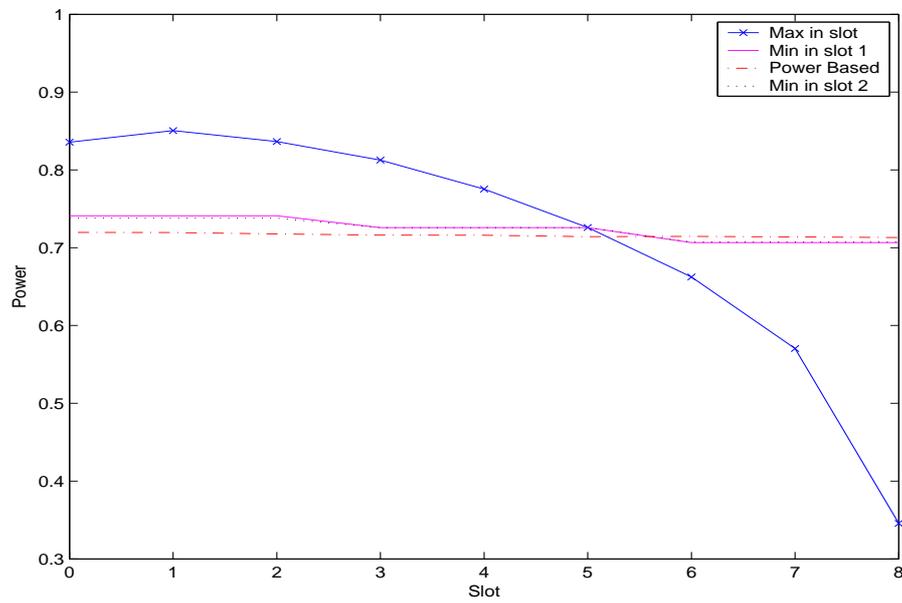


Figure 5.11: puissance moyenne par slot, charge moyenne

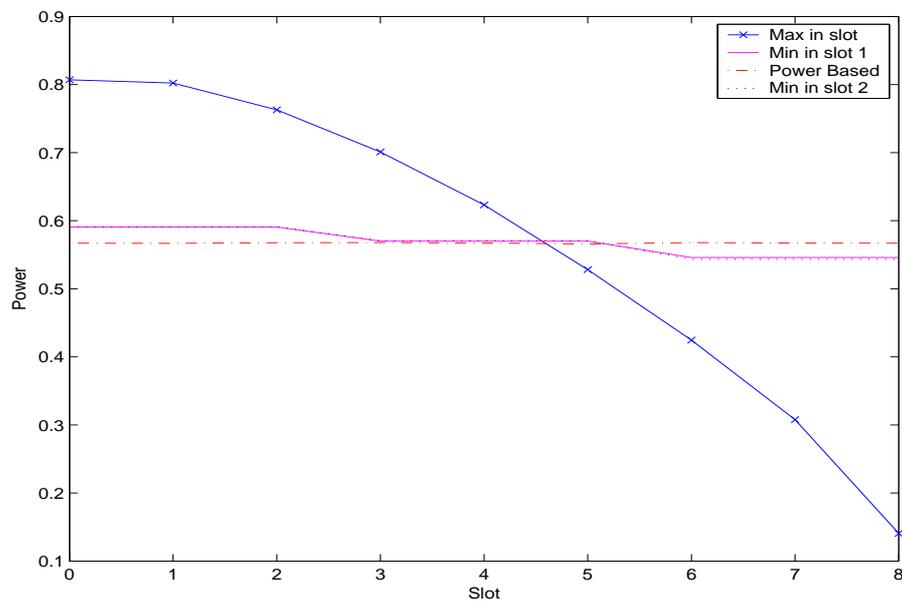


Figure 5.12: puissance moyenne par slot, charge faible

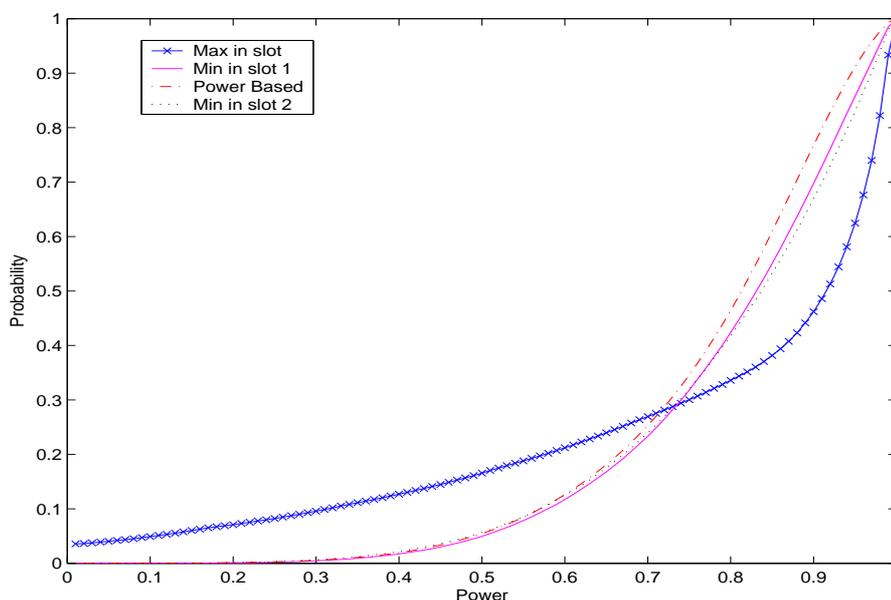


Figure 5.13: cdf charge forte

ayant une meilleure répartition de la puissance, ont bien une probabilité très faible d'avoir des slots inutilisés, et on également une probabilité plus faible d'avoir des slots à la limite du seuil de puissance.

La table 5.3 donne les taux de rejet en fonction de la charge. Le taux de rejet dépend pour beaucoup des lois d'arrivés et de services. Mais nous observons que là encore le taux de rejet diffère selon les algorithmes. Nous observons que le taux de rejet du Max in slot est le plus mauvais. Cela ne nous surprend pas, puisque la répartition de sa puissance est la plus déséquilibrée.

Charge	Max in slot	Min 1	Power Based	Min 2
forte	0.296	0.275	0.282	0.278
moyenne	0.179	0.168	0.171	0.167
faible	0.060	0.058	0.060	0.058

Table 5.3: Taux de rejet en fonction de la charge et de l'algorithme

La figure 5.14 montre un exemple typique où l'utilisation du Max in slot entraîne le blocage du slot 2 par atteinte du seuil de puissance, alors qu'il utilise peu de ressource, le slot 1 utilise en revanche toutes ses ressources, mais est très loin d'atteindre le seuil de puissance. Avec l'algorithme Max in slot, l'utilisateur #2 est rejeté, alors qu'il est accepté avec l'algorithme Min in slot.

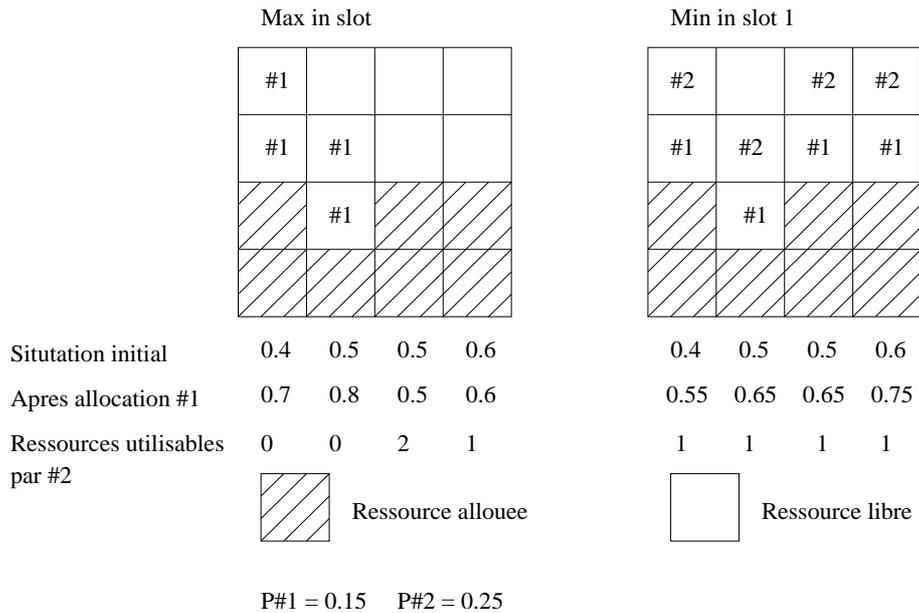


Figure 5.14: Echec du Max in slot

Nous notons que le Min in slot 1 et le Min in slot 2 ont là encore des comportements très proches. Nous notons cependant que le Power Based, bien qu'il a une meilleure répartition de sa puissance, n'a pas le taux de rejet le plus faible, il est même légèrement moins bon. La figure montre un exemple typique où le Power Based ne fait pas le meilleur choix dans l'allocation des ressources. Seul 3 slots sont utilisés dans cet exemple. Le slot 1 est proche de son seuil de puissance dès l'utilisateur #1, et les slots 2 et 3 se remplissent en étant toujours loin du seuil de puissance. On observe que la moyenne de puissance par ressource utilisée est très différente, alors que la somme de puissance est au contraire très proche. Cet exemple d'allocation accepte l'utilisateur si nous utilisons Min in slot 1 ou Min in slot 2.

## 5.8 Conclusion

L'algorithme power based a été optimisé afin de pouvoir gérer de manière efficace la puissance émise. L'idée sous-jacente étant de minimiser la puissance moyenne émise afin de ne pas perturber (par de l'interférence inter-cellule) les cellules avoisinantes et d'améliorer ainsi la performance totale du réseau. A titre de rappel, les systèmes CDMA sont connus pour être des systèmes "interference limited" et toutes les stratégies actuelles (à travers la détection multi-utilisateur et les décodages itératifs) visent à réduire cette interférence au prix d'une complexité accrue. Notre algorithme propose à travers une complexité beaucoup plus réduite (mise en place au niveau de la station de base) et originale une solution possible au problème. En aucun cas, nous prétendons montrer son optimalité. Il peut être tout à fait raisonnable aussi de concentrer la puissance sur certains instants afin de ne pas interférer avec les autres cellules sur les autres instants. Dans le cadre de cellules pouvant se synchroniser, cela sera sans doute le cas. Mais, même dans le cas où les cellules ne sont pas synchronisées, une stratégie d'émission de puissance selon une loi de probabilité est également possible. En fait, le problème de contrôle de puissance dans un contexte multicellulaire est encore complètement ouvert d'un point de vue théorie de l'information et dépend de plusieurs facteurs : détecteur utilisé, type de canal, le path loss. Bien que très difficile, ce sujet attire actuellement beaucoup de chercheurs en théorie de l'information.

Nous allons, dans le chapitre prochain, introduire du trafic NRT. Le trafic NRT devra notamment ne pas perturber le trafic RT présent. Nous pourrions également vérifier que, grâce à sa meilleur répartition de la puissance, l'algorithme d'allocation que nous avons présenté à ce chapitre est celui qui permet à la plus grande quantité de paquets NRT d'être transmis simultanément avec le trafic RT.



## Chapitre 6

---

# Ordonnancement des services NRT en présence de services RT

---

### 6.1 Introduction

Nous avons dans le chapitre précédent montré les mécanismes de contrôle d'accès pour les applications Temps Réel.

Nous voulons maintenant transporter des applications Non Temps Réel en même temps que des applications Temps Réels. Les applications NRT ont pour principales caractéristiques de ne pas avoir de contraintes sur les délais et d'avoir un débit non garanti.

Pour accepter une nouvelle application, le Contrôle d'accès (CAC) tenait compte de la puissance d'émissions des différents flux. N'ayant pas de contraintes dans les délais (les paquets de données pouvant être mis en attente), les applications NRT ne subiront pas de contrôle d'accès.

Nous proposerons un mécanisme garantissant que le transport des applications NRT ne perturbe pas le trafic RT déjà accepté. Si nous réservions des ressources pour les applications RT, nous ne le ferons pas pour les applications NRT, afin de ne pas avoir de ressources non utilisés et non utilisables pendant que les connexions NRT sont silencieuses. Nous utiliserons donc des canaux partagés (DSCH, Downlink Shared Channel) pour

le trafic NRT. L'utilisation de canaux de transports identiques pour des flux différents nécessite une politique d'ordonancement pour le partage des ressources. Nous allons donc commencer par faire un rappel sur les politiques d'ordonancement. Nous présenterons ensuite un algorithme d'ordonnement opportuniste, développé par Liu dans [77]. Nous appliquerons cet ordonnancement au trafic NRT, et montrerons comment nous arrivons à concilier les contraintes du trafic RT, diminuer les pertes pour le trafic NRT, tout en maintenant l'équité entre les applications NRT.

## 6.2 Rappel sur les ordonnancements

A chaque application NRT correspond une entité RLC. Chaque entité RLC contient un espace où elle peut stocker des PDU. Donc, à chaque application NRT, sont associées une entité RLC et une file d'attente. Chaque DSCH peut servir un certain nombre de PDU (selon la configuration du DSCH et la configuration du service) par TTI. Donc nous avons un système de plusieurs files d'attentes avec plusieurs serveurs.

A chaque TTI, un mécanisme d'ordonnement [75] va indiquer aux applications quelles sont celles qui vont pouvoir transporter des PDU et sur quels DSCH.

Plusieurs algorithmes d'ordonnement sont possibles.

### 6.2.1 Round Robin

L'ordonnement parcourt séquentiellement les files d'attente et prend le premier élément. Si la file est vide, l'ordonneur passe instantanément à la suivante.

### 6.2.2 Weighted Round Robin

L'ordonnement Weighted Round Robin privilégie certains flux en définissant un poids pour chaque classe de trafic. Ce poids correspond au nombre de paquets en tête de la file d'attente que peut prendre l'ordonneur [76]. Dans notre cas particulier, en supposant que les PDU de chaque file sont de même taille et que chaque DSCH peut transporter un paquet et un seul, l'ordonnement WRR alloue un ou plusieurs DSCH à un flot en fonction de son poids. Le mécanisme est séquentiel : il alloue les DSCH par TTI, Lorsque l'ordonnement a attribué tous les DSCH, il s'interrompt et reprend au TTI suivant.

### 6.3 WCFQ, Un algorithme d'ordonnement opportuniste

Beaucoup d'auteurs ont proposé des mécanismes d'ordonnement pour les réseaux sans fils qui tiennent compte des conditions du canal radio. La plupart de ces algorithmes permettent d'accroître l'efficacité du système, en diminuant les pertes ou en permettant de diminuer la puissance d'émission nécessaire aux utilisateurs préemptés. Cependant, un utilisateur ayant un mauvais canal est pénalisé, car ses paquets ne sont jamais transportés.

WCFQ (Wireless Credit Fair Queing), l'algorithme développé par Liu dans [77] utilise une fonction de coûts qui dépend de l'état du canal, et un mécanisme de crédit [79], [80], [81] [82], qui diminue si l'utilisateur a un paquet transporté ou s'accroît si il est mis en attente.

La fonction qui sélectionne un flux parmi les files est donné par :

$$f_{p+1} = \underset{i \in B(p)}{\operatorname{argmin}} \frac{L_i(p) - K_i(p) + U_i(p)}{\phi_i} \quad (6.1)$$

Où  $f_{p+1}(p)$  est le flot sélectionné au temps  $p+1$ ,  $B(p)$  est l'ensemble des flots ayant des paquets en attente au temps  $p$ ,  $L_i(p)$  est la taille du paquet en tête de la file d'attente  $i$ ,  $K_i(p)$  est la valeur du crédit du flot  $i$  au temps  $p$ ,  $\phi_i$  est le poids du flot  $i$ , et  $U_i(p)$  est la fonction de coût au temps  $p$ .

Le crédit  $K_i$  d'un flot est modifié à chaque fois qu'un paquet est affecté à un DSCH selon les règles suivantes :

$$K_i(p+1) = K_i(p) + \max\left(\frac{L_{f_{p+1}}(p) - K_{f_{p+1}}(p)}{\phi_{f_{p+1}}}, 0\right)\phi_i \quad (6.2)$$

si le flot  $i$  a des paquets en attente et qu'il n'est pas le dernier flot à avoir été sélectionné,

$$K_f(p+1) = \max(0, K_{f_{p+1}}(p) - L_{f_{p+1}}(p), 0) \quad (6.3)$$

pour le flot qui vient d'être sélectionné par l'algorithme d'ordonnement

$$K_i(p+1) = 0 \quad (6.4)$$

si le flot  $i$  n'a plus de paquet en attente.

La fonction de coûts  $U_i$  et l'état du canal  $E_i$  sont reliés par :

$$U_i = -\beta \log(1 - E_i) \quad (6.5)$$

Où  $\beta$  est un coefficient qui permet de modifier le poids de la fonction de coûts dans la fonction de sélection des flots.

La transmission des paquets d'un utilisateur est retardée si l'état de son canal est défavorable et si son crédit est trop bas. Pendant sa mise en attente, son crédit est augmenté. Si son crédit devient trop élevé, il est servi, même si l'état de son canal est encore défavorable. Si son canal s'améliore, il peut également être servi.

En faisant varier la valeur  $\beta$ , de la fonction de coûts, on fait varier l'importance de l'état du canal dans la sélection du flux à transporter. Si  $\beta=0$ , l'algorithme d'ordonnancement ne tient plus compte de l'état du canal et se comporte comme un WRR.

## 6.4 Notion d'équité

L'équité peut être mesurée par :

$$\frac{\left(\sum \frac{x_i}{\phi_i}\right)^2}{n * \sum \left(\frac{x_i}{\phi_i}\right)^2} \quad (6.6)$$

Où  $\phi_i$  est le poids du service  $i$ ,  $x_i$  est la proportion de trafic alloué au service  $i$ , et  $n$  est le nombre de service. Quand ce ratio est proche de 1, l'ordonnancement est équitable.

Si tous les services ont la même priorité, la formule devient :

$$\frac{(\sum x_i)^2}{n * \sum x_i^2} \quad (6.7)$$

C'est l'index d'équité de Jain (*Jain's fairness index*) [83].

## 6.5 Adaptation de WCQF à l'UMTS

Nous voulons transporter les services NRT sur les DSCH. Pour tenir compte des conditions radios et rester équitables, nous allons utiliser WCQF pour l'ordonnancement downlink. Mais, nous allons transporter nos applications NRT en présence d'application RT sur des DSCH. Comme les applications RT sont soumises à des délais, nous ne pouvons prendre le risque de dégrader leurs services, car des pertes sur des applications RT entraîneraient des retards préjudiciables à la fourniture de la QoS. Aussi, lorsque un service sera sélectionné, il faudra parcourir les DSCH, de manière à en trouver un

dont l'utilisation par ce service ne fera dépasser le seuil de puissance à aucun des slots concernés par cette allocation. Si un tel DSCH peut être trouvé, on attribue ce DSCH au premier paquet en attente de ce service, on met à jours les valeurs de  $K_i$  de tous les services en cours, et on sélectionne un nouveau service (qui peut être le même ou différent). L'ordonnancement se poursuit tant qu'il reste des DSCH utilisables. Lorsque tous les DSCH sont utilisés ou que l'utilisation des DSCH restants fait dépasser les seuils de puissance, l'ordonnancement s'arrête. Il reprendra au TTI suivant.

## 6.6 Expériences

Considérons une configuration des ressources comme celle de la figure 6.1. Neuf slots sont alloués pour le transport des données en downlink. (Les 6 autres slots ne sont pas représentés). Trois sont dédiés à la signalisation (BCH, FACH, RACH, et trois autres sont dédiés pour le transport en uplink). En downlink, les 8 premières ressources de chaque slot sont des ressources réservées pour les DCH. Quand un utilisateur se présentera, des ressources lui seront affectées parmi les ressources encore libres de cet ensemble. 24 DSCH sont configurés, tous constitués de 3 ressources. Ces DSCH sont configurés de manière à transporter un Block de Transport (Block de Transport avec un champ de donnée de 320 bits, codée avec le code convolutif de taux 1/2 et une limite de poinçonnage de 0.9).

Nous allons tester cet algorithme avec 10 services NRT. Ces services auront 3 niveaux de priorités. 5 services seront de priorités 3, la plus faible, 3 services seront de priorités 4, et 2 services seront de priorités 5, la plus forte. Nous calculons  $\varphi_i$  par :

$$\varphi_i = \frac{P_i}{\sum P_j} \quad (6.8)$$

$\varphi_i$  est égale au taux de ressources qu'aurait un utilisateur NRT en cas d'ordonnancement WRR.

Le modèle du canal sera le suivant : nous considérons une chaîne de Markov à 10 états [87], modèle généralisé de Gilbert-Elliot [87] [41] [88] (figure 6.2). Ces états représentent un bruit gaussien qui va de 3.3 dB (état 0) à 1.5 dB (état 9). Il y a une différence de 0.2 dB entre chaque état. L'état de chaque utilisateur est fixé aléatoirement au début de l'expérience et il variera indépendamment pour chaque utilisateur.

Tous les seuils de transition ont une probabilité de 1/3, à l'exception de l'état 9 que l'on quitte avec une probabilité de 2/3. La probabilité de

OVSF 15	DSCH 21	DSCH 21	DSCH 21	DSCH 22	DSCH 22	DSCH 22	DSCH 23	DSCH 23	DSCH 23
OVSF 14	DSCH 18	DSCH 18	DSCH 18	DSCH 19	DSCH 19	DSCH 19	DSCH 20	DSCH 20	DSCH 20
OVSF 13	DSCH 1	DSCH 3	DSCH 5	<b>DSCH 7</b>	DSCH 9	DSCH 11	DSCH 13	DSCH 15	DSCH 17
OVSF 12	DSCH 1	DSCH 3	DSCH 5	DSCH 7	DSCH 9	DSCH 11	DSCH 13	DSCH 15	DSCH 17
OVSF 11	DSCH 1	DSCH 3	DSCH 5	DSCH 7	DSCH 9	DSCH 11	DSCH 13	DSCH 15	DSCH 17
OVSF 10	DSCH 0	DSCH 2	DSCH 4	DSCH 6	DSCH 8	DSCH 10	DSCH 12	DSCH 14	DSCH 16
OVSF 9	DSCH 0	DSCH 2	DSCH 4	DSCH 6	DSCH 8	DSCH 10	DSCH 12	DSCH 14	DSCH 16
OVSF 8	DSCH 0	DSCH 2	DSCH 4	DSCH 6	DSCH 8	DSCH 10	DSCH 12	DSCH 14	DSCH 16
OVSF 7	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH
OVSF 6	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH
OVSF 5	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH
OVSF 4	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH
OVSF 3	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH
OVSF 2	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH
OVSF 1	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH
OVSF 0	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH	DCH
	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8

Figure 6.1: Configuration des ressources

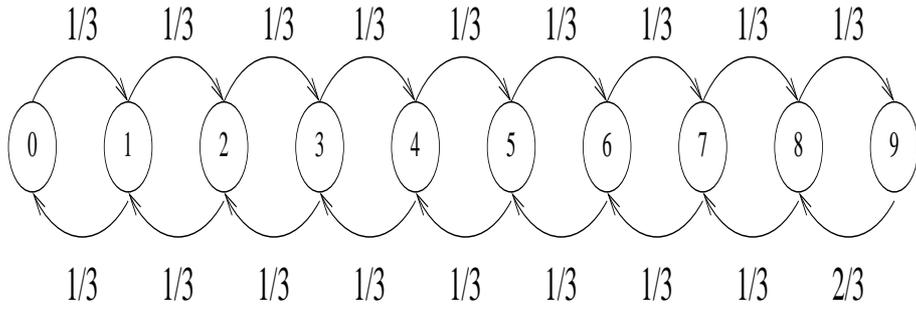


Figure 6.2: Modèle du canal

rester dans un même état est de  $1/3$ , sauf pour l'état 0 où l'on reste avec une probabilité de  $2/3$ .

La fonction de coûts 6.5 se calcule par :

$$U_i = -\beta \log\left(1 - \frac{S_i}{N_{etat}}\right) \quad (6.9)$$

où  $S_i$  est l'état du canal du service  $i$ . et  $N_{etat}$  le nombre d'état que peut prendre le modèle du canal. 3 valeurs de  $\beta$  sont testés : 0, 10 000 et 100 000.

### 6.6.1 RT sur DCH, NRT sur DSCH

Nous voulons maintenant d'une part connaître l'impact des algorithmes d'allocation sur le taux de rejet des appels en présence de trafic NRT, et d'autre part, savoir si certains de ces algorithmes permettent un transport d'un plus grand nombre de trafic NRT.

Pour les services RT, le temps moyen entre deux arrivées est de 20 s et la durée moyenne d'un service est de 150 s. Les 10 services NRT ont leurs files saturées. Il y a toujours des paquets en attente. La figure 6.3 donne les puissances moyennes de chaque slot en fonction des algorithmes vus au chapitre précédent. La mesure est faite en tenant compte des Blocs de Transport des applications NRT transportés sur les DSCH. Nous constatons que l'impact des algorithmes d'allocation sur l'équilibre de la puissance devient minime. Par contre, d'après les mesures de la table 6.1, ces algorithmes ont toujours une influence sur le taux de rejets des appels. Ils ont également un impact sur la charge du DSCH.

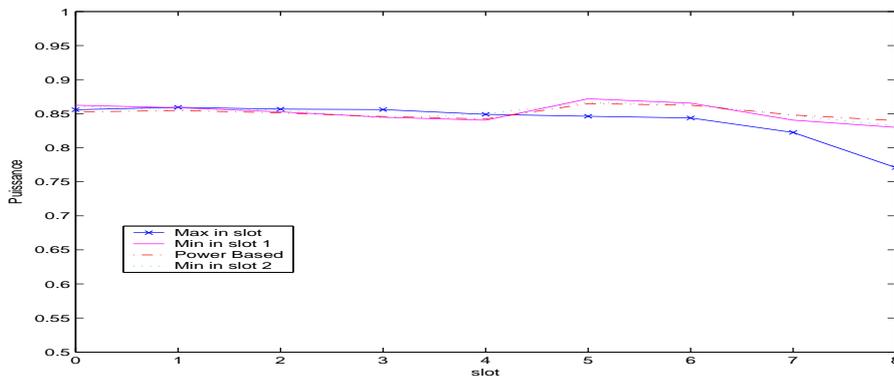


Figure 6.3: Puissance moyenne par slot en fonction des algorithmes DCH

	RCP	DSCH Load
Max in slot	$1.37 * 10^{-1}$	0.418
Min in slot 1	$1.28 * 10^{-1}$	0.418
Power Based	$1.31 * 10^{-1}$	0.4244
Min in slot 2	$1.30 * 10^{-1}$	0.416

Table 6.1: Taux de rejet des applications RT et charges des DSCH

La charge du DSCH se calcule par :

$$DSCHLoad = \frac{Numblock}{nDSCH * nTTI} \quad (6.10)$$

Où Numblock est le nombre total de Block de Transport des applications NRT transportés, nDSCH est le nombre de DSCH dans la configuration des ressources et nTTI est le nombre de TTI de l'expérience.

L'algorithme Min in slot 1 donne le meilleur taux de rejet, mais c'est Power Based qui permet la meilleure utilisation des DSCH.

La table 6.2 donne le Block Error Rate (BLER) pour les Block de Transport des services NRT en fonction de  $\beta$ . Les valeurs du BLER sont dans les limites suggérées par [84]. Nous constatons que l'ordonnancement adaptatif permet de diviser par 4 le nombre de Block de Transport erroné.

$\beta$	Block Error Rate
0	$7.9 * 10^{-2}$
10 000	$5.8 * 10^{-2}$
100 000	$2.1 * 10^{-2}$

Table 6.2: Block Error Rate

Les figures 6.4 à 6.6 montrent le comportement de l'ordonnancement en fonction de  $\beta$  [78] [86]. Les lignes montrent l'évolution de l'état du canal d'un utilisateur au cours du temps. Plus la valeur de l'état est petite, plus le canal est bon. Plus la valeur de l'état est élevée, plus l'état du canal est mauvais. Les bars indiquent combien de paquet lui a été servi. Lorsque  $\beta = \text{zéro}$ , donc lorsque l'ordonnancement ne tient pas en compte de l'état du canal, le comportement est identique à celui d'un Weighted Round Robin. Les Blocs de l'utilisateur sont régulièrement transportés en fonction de sa priorité (2 à 3 par TTI). Lorsque  $\beta = 10\ 000$ , l'ordonnancement est moins régulier. Lorsque le canal est mauvais, l'utilisateur n'est pas sélectionné, mais nous observons des pics dès que son canal connaît une amélioration.

Il y a des intervalles de plusieurs TTI où l'utilisateur n'est pas servi, mais il y a également des pics où l'utilisateur a une dizaine de ces blocs de transportés. Pour  $\beta = 100\ 000$ , l'effet est encore accentué. Les plages sans services sont beaucoup plus longues, et les pics sont de tailles beaucoup plus importantes. Pour un  $\beta$  supérieur à  $100\ 000$ , nos expériences ont montré que le comportement était similaire à celui de  $\beta = 10\ 000$ , l'ensemble des ressources pouvant déjà être attribué à l'utilisateur dont le canal était dans l'état le plus favorable.

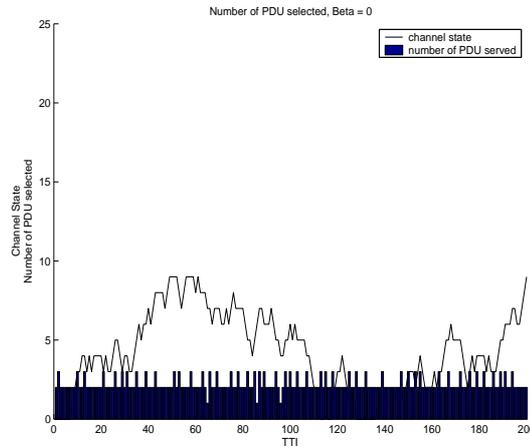


Figure 6.4:  $\beta = 0$ .

L'ordonnement adaptatif préserve-t-il l'équité entre les utilisateurs ? La figure 6.7 montre la quantité de paquets cumulés d'un utilisateur transmise au cours du temps. Pour  $\beta = 0$ , la courbe est assez linéaire, c'est le comportement du WRR. La courbe dont le  $\beta = 10\ 000$  s'en éloigne régulièrement, en ayant servi plus ou moins de paquets que le WRR, mais finit toujours par la recroiser. Pour  $\beta = 100\ 000$ , l'écart est plus important, mais dans la durée, la quantité servie finit toujours par être proche de la quantité servie du WRR. Si le comportement instantané de l'ordonnement pour  $\beta = 10\ 000$  ou  $100\ 000$  est éloigné du WRR, il est en proche en moyenne.

La figure 6.8 montre les quantités de paquets transmises par deux utilisateurs NRT par tranche de 20 TTI en fonction de  $\beta$ . Pour  $\beta = 10\ 000$  et  $100\ 000$ , les quantités servies par unité de temps de deux utilisateurs sont différentes. Cette différence est due à la différence de comportements de

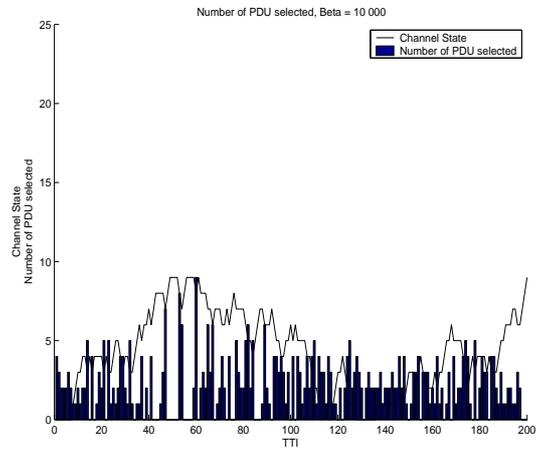


Figure 6.5:  $\beta = 10\ 000$ .

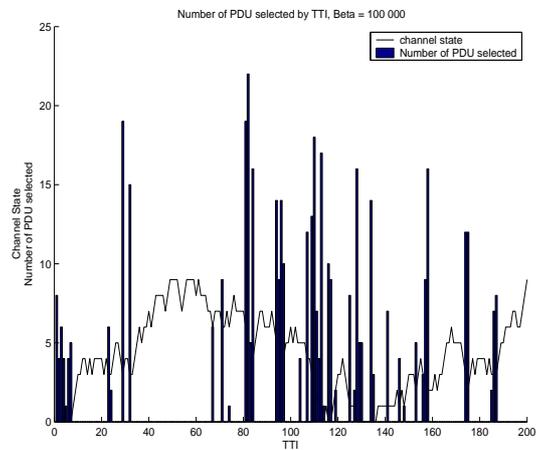


Figure 6.6:  $\beta = 100\ 000$ .

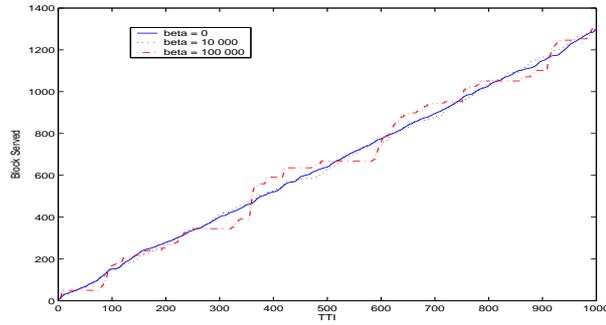


Figure 6.7: Quantité de blocs cumulé d'un utilisateur en fonction de  $\beta$ .

leurs canaux respectifs. On observe là encore que la distribution pour  $\beta = 100\,000$  est la plus éloignée de la moyenne, mais les sommes cumulées de Blocs transmis sur un intervalle de temps suffisamment long montrent que le comportement est équitable.

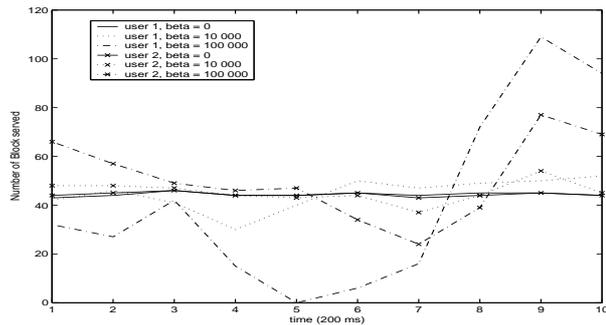


Figure 6.8: Evolution des quantités servies pour 2 services de même priorité au cours du temps.

La table 6.3 donne les proportions de ressources allouées par utilisateurs en fonction de  $\beta$ . Là encore, nous observons que pour  $\beta = 0$ , le comportement de l'ordonnancement correspond parfaitement au WRR. Pour  $\beta = 10\,000$  et  $100\,000$ , on observe cependant des écarts. Nous utilisons l'index 6.6 pour calculer le niveau de l'équité. Pour  $\beta = 0$ , la valeur de l'index est de 1, pour  $\beta = 10\,000$ , la valeur de l'index est de 0.999, et pour  $\beta = 100\,000$  la valeur de l'index est de 0.997. Comme les valeurs de l'index sont très proches

de 1, l'équité de l'ordonnancement est démontrée.

Service Id	$\beta = 0$	$\beta = 10\ 000$	$\beta = 100\ 000$
0	0.081	0.082	0.086
1	0.081	0.081	0.087
2	0.081	0.080	0.084
3	0.081	0.081	0.080
4	0.081	0.081	0.078
5	0.108	0.107	0.105
6	0.108	0.108	0.112
7	0.108	0.109	0.109
8	0.135	0.137	0.137
9	0.135	0.134	0.121

Table 6.3: Proportion de ressources allouées par service

### 6.6.2 RT sur DCH ou DSCH, NRT sur DSCH

La séparation entre ressource DCH et ressource DSCH vu précédemment est cependant trop rigide. Si trop de services RT se présentent par rapport aux dimensionnements choisis, et si les services NRT ont peu de paquets en attente alors il y aura des services RT rejetés pendant que des ressources seront inutilisées [85].

Si un service RT se présente, et qu'il n'est pas possible de lui créer un DCH à partir des ressources réservées pour les DCH, on peut alors décider dans certains cas d'utiliser les ressources de DSCH.

Nous ne créons pas de DCH pour le service RT, mais il sera indiqué au Mobile la liste des DSCH qui lui seront réservés. Ces DSCH lui seront attribués pour la durée de son service et seront donc temporairement supprimés de la liste des DSCH sur le RRM. Le CAC vérifiera que l'utilisation des DSCH qui sont attribués à ce service ne perturbera pas les autres services RT en cours. Les ressources utilisées seront prises en compte par le CAC lors du contrôle d'accès de nouveaux services RT (contrairement aux ressources utilisées pour le trafic NRT).

Du point de vue de l'utilisateur, ce type de configuration est similaire au type de configuration d'un service NRT. On lui indique une liste de DSCH qu'il peut écouter. Cette liste est plus réduite (on lui indique seulement les DSCH qui lui sont réservés) et il sera seul à utiliser ces DSCH pendant son

service.

Sur chaque DSCH, il reçoit un nombre entier de PDU (puisque nous ne faisons pas de multiplexage entre les DSCH). Cette solution ne permet pas une allocation de ressources aussi fine que sur un DCH. Toutefois, un DCH n'aurait pas pu être créé à partir des ressources DSCH, car les utilisateurs NRT en cours se sont vu signaler l'existence de ces DSCH. Or chaque DSCH à un TFCl, alors que le DCH n'en a qu'un seul. Si plusieurs DSCH étaient inclus dans ce DCH, alors des données seraient transportées sur certains TFCl. Si ces données coïncidaient avec un numéro d'utilisateur, celui-ci pourrait croire que ce DSCH lui a été alloué pour ce TTI. L'écoute de ce paquet qui ne lui est pas destiné pourrait perturber le système.

Des conditions pourront être fixées pour l'utilisation de DSCH par un service RT. Ces conditions pourront dépendre par exemple de la priorité du service RT entrant ou du taux de rejet maximum toléré pour les services RT.

La figure 6.9 montre l'algorithme utilisé pour l'allocation de ressources. A chaque TTI :

- nous essayons d'abord d'allouer les nouveaux services RT sur des DCH;
- si ce n'est pas possible, et que le service RT en cours de traitement est autorisé à utiliser des DSCH alors nous essayons de le placer sur des DSCH;
- si ce n'est pas possible nous rejetons ce service RT;
- une fois que tous les nouveaux services RT sont placés ou rejetés, nous ordonnons les PDU en attente des services NRT sur les DSCH restants.

A tout moment, l'utilisation d'une nouvelle ressource ne doit pas perturber les services transportés sur les ressources déjà alloués.

Reprenons l'expérience vu au paragraphe 6.6.1. La loi d'arrivée a une distribution exponentielle de moyenne 20, la loi de services a une distribution exponentielle de moyenne 150, les utilisateurs RT utilisent 7 ressources.

Nous fixons maintenant comme condition un taux de rejet des services RT à 10 %. Un compteur est mis en place. Si le rejet d'un appel fait dépasser ce seuil, alors nous essayerons de transporter le service RT sur des DSCH. Sinon, nous le rejeterons. Le tableau 6.4 nous donne les résultats. Nous constatons que cet objectif peut être atteint, et ce quel que soit l'algorithme utilisé pour l'allocation des services RT sur les DCH. Nous constatons par contre que le nombre de PDU transportés pour les services NRT se réduit,

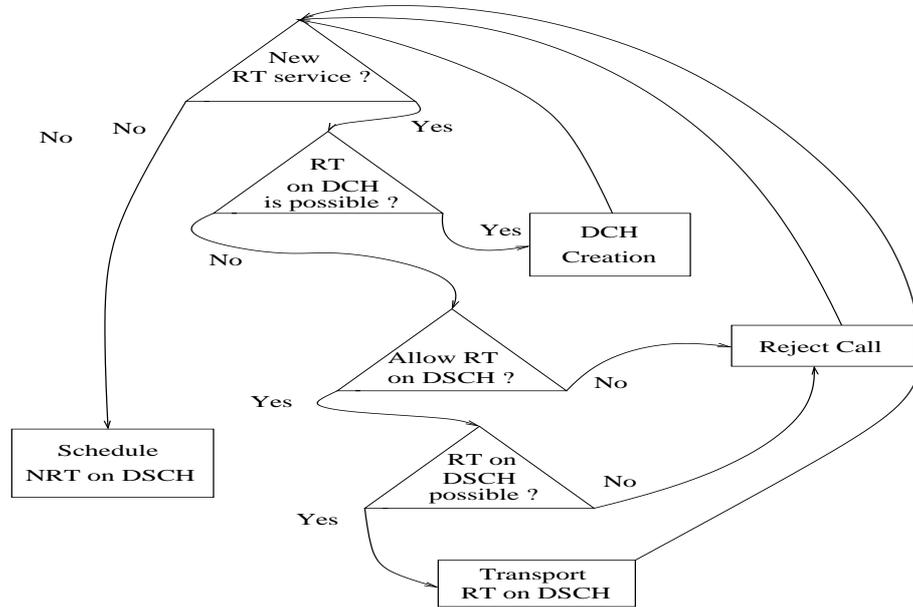


Figure 6.9: Allocation de RT sur DCH et DSCH

ce qui est normal puisqu'une partie de leurs ressources est utilisée par les services RT. Nous constatons que là encore, le MAX in slot transporte moins de paquets que les autres algorithmes.

	RCP	UR DSCH
MAX in slot	$1.0 * 10^{-1}$	0.40
MIN in slot 1	$1.0 * 10^{-1}$	0.41
Power Based	$1.0 * 10^{-1}$	0.41
MIN in slot 1	$1.0 * 10^{-1}$	0.41

Table 6.4: Taux de rejet des services RT et taux d'utilisation des DSCH pour les services NRT, seuil de 10 % de rejet pour le trafic RT

Nous autorisons maintenant l'ensemble des applications RT à utiliser les DSCH. La table 6.5 nous donne les résultats.

L'utilisation des DSCH ne permet pas l'absence de rejet en cas de trafic trop important. On remarque là encore le meilleur comportement des algorithmes Min in slot et Power Based par rapport à l'algorithme Max in slot.

	RCP	UR DSCH
MAX in slot	$5.2 * 10^{-2}$	0.34
MIN in slot 1	$4.5 * 10^{-2}$	0.36
Power Based	$4.9 * 10^{-2}$	0.36
MIN in slot 2	$5.1 * 10^{-2}$	0.36

Table 6.5: Taux de rejet des services RT et taux d'utilisation des DSCH pour les services NRT, les RT utilise les DSCH librement.

### 6.6.3 Conclusion

Nous avons montré l'intérêt d'un ordonnancement opportuniste dans l'UMTS. En effet, un tel ordonnancement, tout en ne perturbant pas le trafic RT transmis en parallèle, et tout en restant équitable, permet de réduire le BLER des applications NRT. Nous avons également montré que les algorithmes d'allocation présentés au chapitre précédent permettent d'accroître le nombre de paquets transmis sur les DSCH par rapport à l'algorithme Max In Slot. Enfin, nous avons proposé un mécanisme qui permet à du trafic RT d'utiliser des ressources DSCH. Cette flexibilité dans l'allocation de ressources permet de réduire la probabilité de rejet d'un utilisateur RT, mais en contre partie, elle ralentit la diffusion des applications NRT.



## Chapitre 7

---

# Intégration sur la plate-forme UMTS d'Eurécom

---

### 7.1 Introduction

Pendant cette thèse, nous avons constamment travaillé avec la plate-forme UMTS PLATON : une partie de nos travaux y a été intégrée, et nous l'avons utilisée pour nos expériences.

Ont été intégrées à la plate-forme, dans le cadre du projet Européen Moby Dick : la génération du QoS IE en fonction des champs Diffserv des paquets IPv6 (chapitre 2) et la configuration des services en fonctions du QoS IE (chapitre 3). La plate-forme peut fonctionner en deux modes : un mode expérimental qui utilise la voie radio et un mode simulation qui utilise les différentes couches de la plate-forme mais qui simule le canal radio. Pour le mode simulation, nous avons développé un serveur et un simulateur de canal.

Nos expériences sur le lien adaptatif (chapitre 4), ont utilisé le code des couches RLC (pour la segmentation et l'assemblage des SDU et le marquage, la transmission et la retransmission des PDU) et la couche L1 (pour l'utilisation des codes correcteurs et la détection des PDU erronés).

Nos expériences des chapitres 5 et 6, sur le contrôle d'accès, l'allocation

de ressource et l'ordonnancement ont utilisé le code de la couche RRM.

La figure 7.1 indique les couches où nous avons effectué nos travaux.

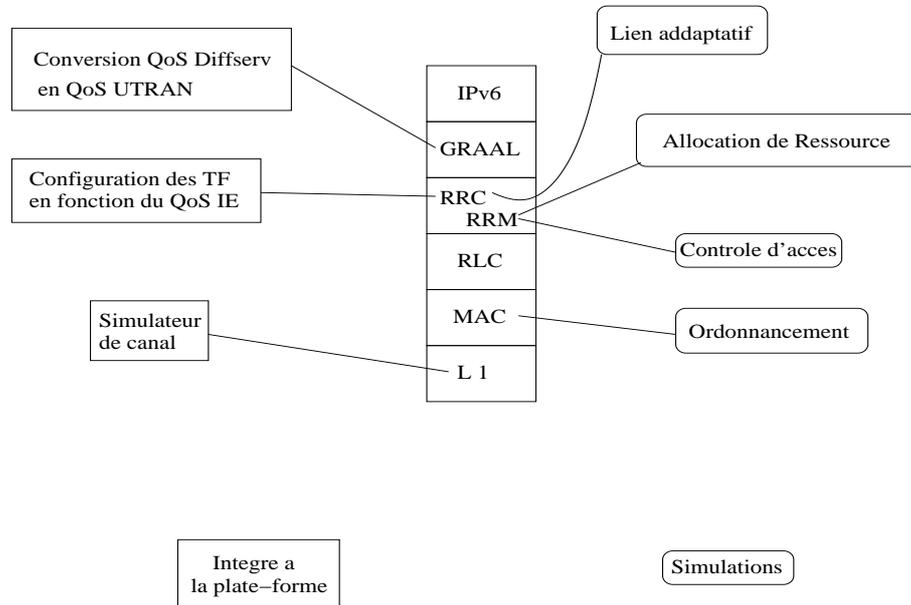


Figure 7.1: Utilisation et intégration de la plate-forme

Nous détaillerons dans ce chapitre, à titre indicatif, nos travaux intégrés à la plate-forme qui ne constituent pas directement un travail de recherche, mais qui ont été des outils indispensables pour ces travaux. Nous ferons également des rappels théoriques sur le traitement du signal.

## 7.2 Le serveur de canal

Pour simuler le canal, nous utilisons un serveur central communiquant avec des clients par des sockets (figure 7.2). Chaque client représente une station de Base ou un Terminal Mobile.

Le serveur de canal simule la chaîne de transmission Radio Fréquence représentée figure 7.3. Le signal transmis est dans un premier temps amplifié par le gain d'antennes  $G_{tx}$  et atténué par le Path Loss dû la propagation du signal Hertzien. Le signal subit également des distorsions (effet doppler dû au déplacement, multi trajets) représenté par un canal  $h(t, x)$  auquel s'ajoute un bruit additif gaussien  $v(t)$ . Le signal est ensuite amplifié en réception

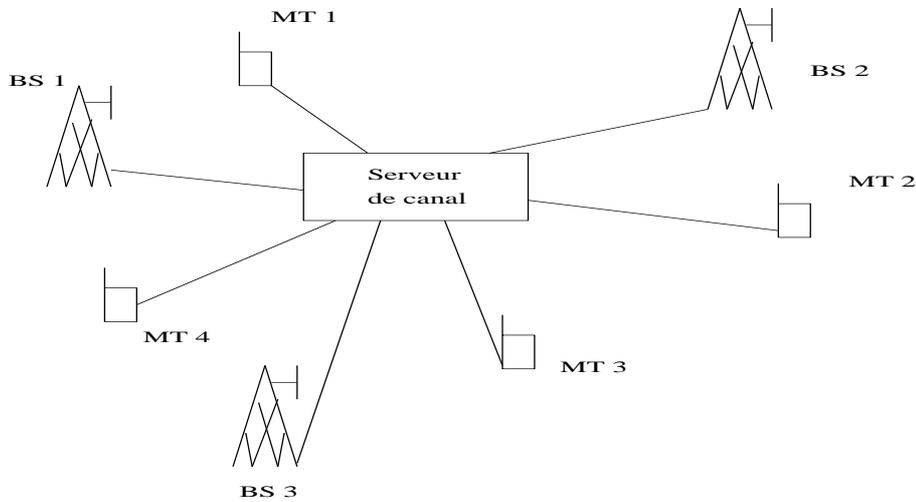


Figure 7.2: Le serveur de canal

par le gain de l'antenne réceptrice  $G_{Rx}$ .

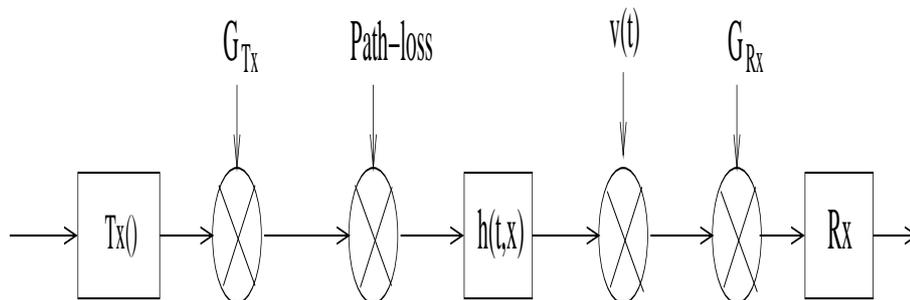


Figure 7.3: La chaîne Radio Fréquence

Le paragraphe suivant fera un rappel sur les principes de propagation.

### 7.3 Principes de la Propagation

Nous allons maintenant introduire des modèles de propagation : L'effet Doppler, les canaux à évanouissements, le path loss, et le shadowing. Les points abordés proviennent essentiellement de : [89] et [90].

### 7.3.1 Effet Doppler et canaux à évanouissement

Considérons le cas de la figure 7.4 où il existe seulement une ligne de vue (LOS, line-of-sight) entre l'émetteur et le récepteur. La position du récepteur change par rapport à celle de l'émetteur. La distance entre l'émetteur et le récepteur au temps  $t$  est donnée par :

$$d_0(t) = -v_0 t + d_0$$

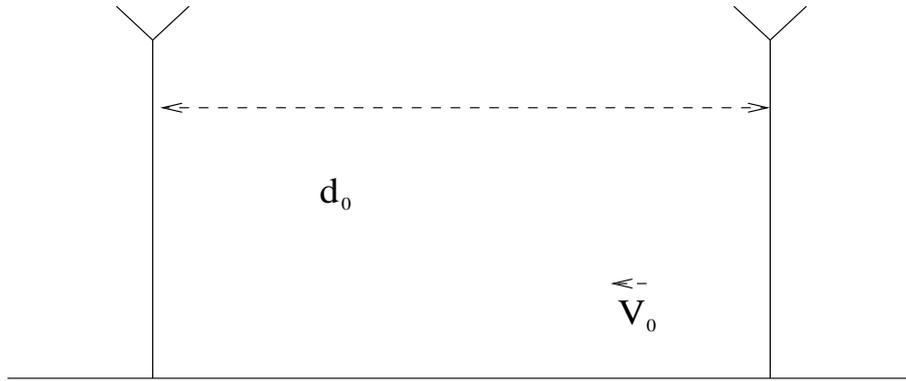


Figure 7.4: Propagation directe avec un terminal mobile.

Le signal transmis peut être noté par :

$$s(t) = \Re\{x(t) \exp(j2\pi f_c t)\}$$

Où  $x(t)$  est l'enveloppe complexe et  $f_c$  est la fréquence de la porteuse. En absence d'autre perturbation, le signal reçu est juste une version retardée du signal transmis, où le délai de propagation est donné par :

$$\tau_0(t) = d_0(t)/c = -(v_0/c)t + d_0/c$$

Où  $c$  représente la vitesse de la lumière. Dans le cas des transmissions terrestres,  $v_0/c$  est très petit. Cependant, le terme  $\xi_0 = f_c v_0/c$  peut être non négligeable car la fréquence porteuse  $f_c$  est normalement très large. Dans l'UMTS,  $f_c \approx 2$  GHz. Ainsi, un mobile se déplaçant à  $v_0 = 100$  km/h a un  $\xi_0 = 184.66$  Hz. Le signal reçu peut être noté :

$$\begin{aligned} r(t) &= \Re\{x(t - \tau_0(t)) \exp(j2\pi f_c (t - \tau_0(t)))\} \\ &\approx \Re\{x(t - \tau_0) \exp(j2\pi (f_c + \xi_0)t) \exp(j\phi_0)\} \end{aligned} \quad (7.1)$$

Où  $\phi_0 = -2\pi f_c d_0/c$  et  $\tau_0 = d_0/c$ , Nous avons supposé que, puisque  $v_0/c$  est petit,  $x(t - \tau_0(t)) \approx x(t - \tau_0)$  sur l'intervalle d'observation. De (7.1) nous notons que la variation dans le temps des délais de propagation provoque un changement de la fréquence porteuse de  $\xi_0$  (appelé *effet Doppler*) et un déplacement de phase de la porteuse de  $\phi_0$ .

Maintenant, considérons la situation de la figure 7.5, où est ajouté au LOS un autre effet de propagation dû à la réflexion du signal contre des obstacles (par exemple un arbre ou un mur).

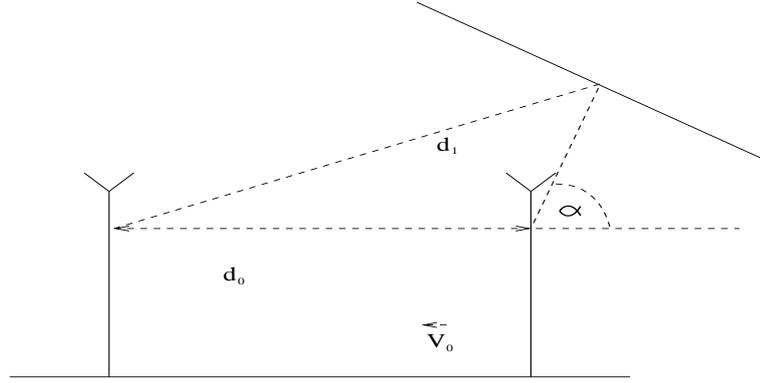


Figure 7.5: Deux chemins de propagation.

Soit  $d_i(t) = -v_i t + d_i$ , pour  $i = 0, 1$ , la distance de propagation, où  $v_0$  et  $v_1$  sont les vitesses relatives du récepteur par rapport à l'émetteur et les réflecteurs. En définissant  $\xi_i = f_c v_i/c$ ,  $\phi_i = -2\pi f_c d_i/c$  et  $\tau_i = d_i/c$ , nous obtenons le signal reçu

$$\begin{aligned} r(t) &\approx \Re\{x(t - \tau_0) \exp(j\phi_0) \exp(j2\pi(f_c + \xi_0)t) + ax(t - \tau_1) \exp(j\phi_1) \exp(j2\pi(f_c + \xi_1)t)\} \\ &= \Re\left\{\left(\sum_{i=0}^1 \rho_i x(t - \tau_i) \exp(j2\pi\xi_i t)\right) \exp(j2\pi f_c t)\right\} \end{aligned} \quad (7.2)$$

où  $a$  est une atténuation complexe due à la réflexion, et nous avons définis le coefficient complexe  $\rho_0 = \exp(j\phi_0)$  et  $\rho_1 = a \exp(j\phi_1)$ . L'enveloppe complexe du signal reçu (7.2) est :

$$y(t) = \sum_{i=0}^1 \rho_i x(t - \tau_i) \exp(j2\pi\xi_i t)$$

Supposons que  $x(t)$  varie lentement, donc que  $x(t) \approx x$ , constant sur l'intervalle d'observation. L'enveloppe du signal reçu peut être sujet à des variations

dans le temps. En fait nous avons :

$$\begin{aligned}
 |y(t)| &= |x\rho_0| \left| 1 + \frac{\rho_1}{\rho_0} \exp(j2\pi(\xi_1 - \xi_0)t) \right| \\
 &= A \sqrt{1 + b^2 + 2b \cos(2\pi\Delta\xi t + \Delta\phi)} x(t) & (7.3) \\
 &= |h(t)| x & (7.4)
 \end{aligned}$$

où  $A = |\rho_0|$ ,  $\Delta\xi = \xi_1 - \xi_0$  et  $\rho_1/\rho_0 = b \exp(j\Delta\phi)$ , avec  $b \in \mathbb{R}^+$ . De (7.3) Si l'intervalle d'observation est plus grand que  $1/\Delta\xi$ , l'enveloppe du signal reçu varie entre une valeur maximale  $A|1 + b|$  et une valeur minimale  $A|1 - b|$ . Le taux de variation est donné par la fréquence d'étalement de Doppler (*i.e Doppler spread*)  $\Delta\xi$ , donné par la différence entre le maximum et le minimum du changement de fréquence de Doppler. La fréquence d'étalement de Doppler caractérise la sélectivité temporelle du canal.

En conclusion, nous avons vu que si nous avons plus d'un chemin de propagation, l'enveloppe du signal reçu change dans le temps même si l'enveloppe du signal transmis reste constante. Cet effet est appelé évanouissement (*fading*), et les canaux caractérisés par ce type de variation temporelle sont appelés des canaux à évanouissement (*fading channels*). En général, le nombre de chemins est assez important et une conséquence du théorème de la limite centrale,  $h(t)$  est souvent modélisé comme une variable aléatoire gaussienne de moyenne nulle.

### 7.3.2 Path loss

Bien que nous ayons déterminé les variations des évanouissements dans le cas d'une propagation multi trajet, une étude plus approfondie est nécessaire pour prendre en compte le profile de la densité de la puissance par rapport au path loss. Supposons par exemple que l'antenne émettrice rayonne de façon isotropique dans un espace libre à un niveau de puissance  $P^T$  Watts. La densité de la puissance à la distance  $r$  de l'antenne est  $P^T/4\pi r^2$  W/m<sup>2</sup>. Si l'antenne émettrice peut faire propager le signal dans une certaine direction, la densité de puissance dans cette direction est accrue d'un facteur appelé le gain de l'antenne et noté  $G^T$ . Dans ce cas, la densité de puissance à la distance  $r$  est  $P^T G^T / 4\pi r^2$  W/m<sup>2</sup>. Le produit  $P^T G^T$  est appelé la puissance radiée effective (*effective radiated power*). Pour  $G_T=1$ , nous retrouvons la puissance radiée relative à une antenne isotropique.

Une antenne réceptrice R pointée dans la direction de puissance radiée recueille une portion de la puissance proportionnelle à l'aire de sa section. La

puissance reçue par l'antenne peut donc être exprimée ainsi :

$$P^R = \frac{P^T G^T A^R}{4\pi r^2} \quad (7.5)$$

où  $A^R$  est l'aire effective de l'antenne réceptrice (*effective area of the received antenna*). [91] indique que la relation entre le gain  $G^R$  de l'antenne et son aire effective  $A^R$  dans un espace libre est :

$$A^R = \frac{G^R \lambda^2}{4\pi} \quad (7.6)$$

avec  $\lambda$  la longueur d'onde du signal reçu.

Si nous remplaçons l'expression de  $A^R$  (7.6) dans (7.5), nous obtenons l'expression du signal reçu :

$$P^R = \frac{P^T G^T G^R \lambda^2}{(4\pi r)^2} \quad (7.7)$$

Le facteur

$$L = \left(\frac{\lambda}{4\pi r}\right)^2$$

est appelé *free space path loss*.

Les expériences dans des espaces **non libres** montrent en fait que  $L$  peut être écrit sous la forme :

$$L \sim \frac{1}{r^\beta}$$

Où  $\beta$  (voir table 1) représente l'exposant du path-loss et a des valeurs typiques de l'ordre de 4 dans les environnements urbains.  $\sim$  indique un facteur multiplicatif.

Environnement		$\beta$
Outdoor	Free space	2
	Shadowed urban area	2.7 to 5
Indoor	Line of sight	1.6 to 1.8
	Obstructed	4 to 6

Table 7.1: Valeurs typiques de l'exposant  $\beta$  du path loss

**Remarque** : Dans les équations précédentes, nous avons supposé que l'antenne réceptrice était très grande par rapport à la longueur d'onde et aux dimensions de l'antenne. La distance  $D$  [92] doit satisfaire la condition  $r \geq \frac{2D^2}{\lambda}$  où  $D$  est la longueur de l'antenne.

### 7.3.3 Effet "Shadowing"

Le *shadowing* est un autre effet de propagation mesuré dans les canaux sans fils. Il est provoqué par des obstacles comme les arbres, les murs ou les bâtiments. Ces obstacles ne dispersent pas le signal, mais atténuent la puissance du signal reçue. Les évanouissements de Rayleigh (ou Rice) varient assez rapidement par rapport au shadowing. Par exemple, le temps de cohérence d'un mobile avec un Doppler spread de  $B_d = 100$  Hz est d'environ  $1/B_d = 10$  ms. Au contraire, le récepteur peut être dans les zones d'ombres d'obstacles pour plusieurs secondes. De plus, des évanouissements à "court terme" due à la dispersion et des évanouissements à "long terme" agissent sur des échelles de temps différents.

Les expériences et l'analyse montrent que dans un champ éloigné de l'antenne émettrice, l'atténuation à long terme pour la propagation terrestre est proportionnelle à  $\zeta$ , où  $\zeta$  est une variable aléatoire en log-normal [93] avec  $E[\zeta] = 1$ . L'usage veut que  $\zeta$  soit exprimé en dB, donc

$$10 \log_{10} \zeta \sim \mathcal{N} \left( -\frac{s^2 \log 10}{20}, s^2 \right)$$

L'écart type  $s$ , exprimé en dB, est appelé *shadowing factor* et a une valeur généralement comprise entre 2 et 8 dB.

### 7.3.4 Expression Générale

La puissance instantanée reçue du signal est donc finalement donnée par :

$$P^R = \frac{|h(t)|^2 G^T G^R P^T \xi}{r^\beta}$$

$r$  est la distance entre l'émetteur et le récepteur,  $\beta$  un exposant qui dépend des caractéristiques du terrain,  $P^R$  et  $P^T$  sont respectivement la puissance émise et la puissance reçue,  $G^T$  et  $G^R$  sont respectivement les gains des antennes émettrices et réceptrices.

## 7.4 Calcul du nombre de ressources d'un CCTRCH

Le nombre de ressources qui va être affecté à un CCTRCH dépend de différents paramètres :

- du débit maximum que l'on souhaite transmettre sur ce CCTRCH. Si un seul flux est transporté sur le CCTRCH, il se déduit à partir du dernier TF du TFS de l'application transportée. En cas de multiplexage, il faut

consulter le TFC et calculer parmi les combinaisons autorisées celle qui a le débit total maximum.

Le débit exprimé au niveau du TF est le débit du point de vue de l'application. Mais le débit qui sera fourni sur le lien physique dépend également de la proportion d'entête (donc de la taille du bloc), du type de codage, de la taille du CRC, et du taux de poinçonnage.

- de paramètres du à la configuration du réseau. Il s'agit du facteur d'étalement, du type de burst (I, II ou III), de la longueur du TFCI et de la longueur du TPC. Ces quatre paramètres, conjugué avec le sens d'émission, downlink ou uplink, définissent le burst format sélectionné.

L'algorithme suivant permet le calcul de  $Nb\_ressource$ , le nombre de ressources nécessaire pour la configuration d'un CCTRCH qui reçoit les données d'un seul DCH (pas de multiplexage) :

1)

$$NB\_Block = \left\lceil \frac{BSS}{BS} \right\rceil \quad (7.8)$$

2)

$$after\_CRC = NB\_Block * (BS + CRC\_length) \quad (7.9)$$

3)

$$cc\_num\_bloc = \left\lceil \frac{after\_crc}{treillis\_size} \right\rceil \quad (7.10)$$

4)

$$Cc\_bloc\_size = \left\lceil \frac{after\_crc}{cc\_num\_bloc} \right\rceil \quad (7.11)$$

5)

$$after\_coding = cc\_num\_bloc * rate * (cc\_bloc\_size + tail\_size) \quad (7.12)$$

6)

$$After\_max\_puncturing = \lceil after\_coding * pl \rceil \quad (7.13)$$

7)

$$Nb\_ressource = \left\lceil \frac{After\_max\_puncturing + tfci\_length + tpc\_length}{number\_of\_data\_symbol} \right\rceil \quad (7.14)$$

est l'opérateur arrondi par excès. BSS est la longueur en bit du dernier Transport Format du Transport Format Set.

NB\_Block représente le nombre de blocs transmis par la couche MAC par TTI.

After\_CRC représente le nombre total de bits dans la couche physique avant l'application du code correcteur (mais après l'application du code CRC).

cc\_num\_bloc représente le nombre d'opérations de codage nécessaire. Ce nombre dépend de la taille du treillis utilisé pour le codage.

after\_coding représente le nombre total de symboles après le codage.

After\_max\_puncturing représente le nombre total de symboles après le poinçonnage.

tfc\_i length et tpc length représente respectivement la taille du TFCI et la taille du TPC dans le premier burst du DCH.

Burst\_usual\_data\_size représente le nombre de symboles dans les champs de données dans le type du burst utilisé et pour le SF sélectionné.

Si on multiplexe des services utilisant le même type de codage, alors il faut supprimer la ligne 1 et remplacer la ligne 2 par :  $\text{after\_CRC} = n \cdot \text{TRCH}$  indique le nombre de TRCH multiplexés sur ce CTRCH. Il y a autant de somme dans cet ensemble qu'il y a de combinaison dans le TFC. Cette ligne permet de sélectionner la combinaison qui produit la quantité de données maximale de ce TFC.

Si les services multiplexés n'utilisent pas le même type de codage, alors il faut pour chaque combinaison calculer le After\_max\_puncturing de chaque service pris isolément et faire la somme du After\_max\_puncturing de chaque TRCH du CTRCH. La combinaison sélectionnée sera celle qui a la somme la plus grande.

## Chapitre 8

---

# Conclusion

---

Les travaux de recherche présentés dans cette thèse portent sur la fourniture de la Qualité de Service et l'allocation de ressource pour des applications multimédias dans les réseaux UMTS-TDD.

Dans le chapitre 2, nous avons montré la nécessité de prédéfinir des tables de correspondances pour traduire la description des critères de QoS de la partie filaire en description de critères de QoS pour le réseau d'accès radio (UTRAN). Nous avons montré une méthode pour réaliser cette traduction.

Une fois les critères connus, il est nécessaire de les faire respecter. Dans le chapitre 3, nous avons étudié le comportement de l'UTRAN et identifié les paramètres clefs dans l'optique de la fourniture de la Qualité de Service. Nous avons proposé un paramétrage pour des applications multimédias caractéristiques.

Comme le canal radio a un comportement variable dans le temps, nous avons proposé au chapitre 4, une méthode pour adapter la configuration en fonction de l'état du canal. Nous avons identifié les paramètres clefs à modifier, dans le but d'accroître l'efficacité du système. Nous avons calculé le gain que pouvait apporter cette méthode en terme d'efficacité pour différents scénarios.

Dans le chapitre 5, nous avons proposé un algorithme d'allocation de

ressources relié au Contrôle d'Accès (CAC) dans le sens descendant pour les applications de type Temps Réels (RT). Cet algorithme permet de mieux répartir la puissance entre les slots. En ayant une puissance plus équilibrée, le système peut accepter plus d'utilisateurs simultanément ce qui réduit le nombre d'utilisateurs rejetés par le CAC.

Dans le chapitre 6, nous avons considéré le transport des applications Non Temps Réels (NRT) simultanément avec des applications Temps Réels. Nous avons utilisé conjointement un mécanisme de Contrôle d'Accès pour les applications RT avec un algorithme d'ordonnancement pour les applications de type NRT. Nous avons montré qu'un ordonnancement qui tient compte de l'état du canal réduit le taux de pertes des paquets transportés tout en préservant une répartition équitable des ressources entre les différents utilisateurs.

Cette thèse a donc proposé un ensemble complet de mécanismes pour la configuration d'applications multimédias dans l'UMTS-TDD.

Ces travaux pourront se poursuivre dans plusieurs directions : Si dans le sens descendant le Contrôle d'Accès se fait à partir de la puissance alloué aux services, dans le sens montant, il doit se faire en tenant compte des interférences. Le facteur d'étalement peut également prendre un plus grand nombre de valeurs. Des mécanismes d'allocation de ressources dans le sens descendant, sélectionnant notamment le facteur d'étalement et tenant comptes des interférences pourront être proposés. De plus, la répartition entre les ressources dédiées et les ressources partagées pourra être examinée plus finement. Une étude sur la taille des DSCH et la position des ressources des DSCH devrait permettre une utilisation des ressources plus efficace. Ces algorithmes ont été étudiés dans un environnement monocellulaire, il pourra être intéressant de préciser nos résultats dans un environnement multicellulaire.

## Chapitre 9

---

# Publications

---

### 9.1 Publications concernant cette thèse

Aawatif Menouni, Illia Racunica, Christian Bonnet, “Access control, Code Allocation, and adaptive scheduling for UMTS-TDD”, IEEE Workshop on Adaptive Wireless Networks (AWiN) (In conjunction with GLOBECOM 2004), Dallas, Texas, 3 décembre 2004

Illia Racunica, Aawatif Menouni, Christian Bonnet, “Access Control, flexible code allocation, and opportunistic scheduling for UMTS-TDD”, 2004 International Conference on Software, Telecommunications and Computer Networks (Softcom), Split (Croatie) - Venice (Italie), 10-13 Octobre 2004

Illia Racunica, Aawatif Menouni, Christian Bonnet, “An opportunistic scheduling with fairness for NRT traffic in presence of RT traffic for UMTS/TDD”, WIRELESS PERSONAL MULTIMEDIA COMMUNICATIONS, WPMC 2004, Abano Terme, Italie, 12-15 Septembre 2004

Illia Racunica, Yan Moret, Christian Bonnet, “Performance of a link adaptive system in UMTS networks”, Winter International Symposium on Information and Communication Technologies 2004 (WISICT 2004), Cancun, Mexique, 5-8 Janvier 2004, pages 86 à 91.

Bonnet, C; Callewaert, H; Gauthier, L; Knopp, R; Menouni, A; Moret, Y; Nussbaum, D; Racunica, I; Wetterwald, M, “Open-Source Experimental B3G Networks Based on Software-Radio Technology”, Software Defined Radio Technical Conference and Product Exposition, Orlando (Floride, USA), Novembre 2003

Christian Bonnet, Yan Moret, Illia Racunica, “Link Quality Control for UMTS TDD”, Rapport RNRT Samu SP2D6, septembre 2002

Illia Racunica, Yan Moret, Christian Bonnet, “Simulation des HARQ1 dans l’UMTS”, CFIP 2002, Montréal, Canada, 27-30 mai 2002, pages 275 à 290.

Neda Nikaein, Illia Racunica, and Christian Bonnet, “QoS Architecture”, Rapport RNRT Samu SP2D2, mai 2001

Illia Racunica, Yan Moret, Christian Bonnet, “Simulation of the Hybrid ARQ”, Rapport RNRT Samu, septembre 2001

## 9.2 Présentation de Poster et atelier

Illia Racunica, Christian Bonnet, “Performance evaluation of HARQ in UMTS-TDD”, WPMC’01, AALBORG, Danemark, septembre 2001

Illia Racunica, Yan Moret, Christian Bonnet, “évaluation des performances des HARQ dans l’UMTS”, Septième Atelier d’évaluation de Performances, 28 mai au 01 juin 2001, école Normale Supérieure, Paris

## 9.3 Autres Travaux

Travaux en Recherche Opérationnelle

Laura Wynter, Illia Racunica, “Localisation Optimal location of inter-modal freight hubs”, article accepté dans Transportation Research-A, publication en 2005.

Illia Racunica, Laura Wynter, “ Localisation optimale des hubs de transport intermodal”, Quatrièmes journées nationales de la ROADEF, ENST, Paris, 20 au 22 février 2002.

Laura Wynter, Illia Racunica, “Localisation optimale des hubs de fret intermodaux”, Francoro III, mai 2001

Illia Racunica, Vesselin Siarov, “Utilisation du MODELE “SIMIQ” pour la traversée Alpine Simulation du potentiel des terminaux du combiné”, Rapport Inrets (Institut National de Recherche sur les transports et leurs sécurités), Janvier 2001

Illia Racunica, Laura Wynter, “Optimal location of intermodal freight hubs ”, Rapport de recherche de l’INRIA No 4088, decembre 2000



## Chapitre 10

---

# Bibliographie

---

Le nombre en début de référence indique le numéro de la référence. Les nombres en fin de références indiquent les pages où ce trouver ces références.



# Bibliographie

- [1] K. Al Agha, G. Pujolle, G. Vivier, *Réseaux de mobiles et réseaux sans fil*, Eyrolles, 2001 [1]
- [2] X. Lagrange, P. Godlewski, S. Tabbane, *Réseaux GSM*, Hermes Science Publications, 2000 [1]
- [3] Bonnet, C; Callewaert, H; Gauthier, L; Knopp, R; Menouni, A; Moret, Y; Nussbaum, D; Racunica, I; Wetterwald, M, “Open-Source Experimental B3G Networks Based on Software-Radio Technology”, Software Defined Radio Technical Conference and Product Exposition, Orlando (Floride, USA), Novembre 2003 [2, 22, 60]
- [4] G.Cizault, *IPv6 - Théorie et pratique*, O’Reilly, 2002 [2]
- [5] IUT-TS General Aspects of Quality of Service and Network Performance in Digital Networks, blue book, 1988 [6]
- [6] 3GPP 23-107, v.4.5.0, “Quality of Service (QoS) concept and architecture”, septembre 2002 [6, 14, 16]
- [7] 3GPP TS 23.101, V4.0.0, “General UMTS Architecture”, avril 2001 [7]
- [8] Guy Pujolle, “Les Réseaux”, 2003, Eyrolles. [9]
- [9] Jean-Louis Mélin, “Qualité de Service sur IP”, 2001, Eyrolles. [9]
- [10] IETF, RFC 2474, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, K. Nichols, S. Blake, F. Baker, D. Black, December 1998. [9]
- [11] IETF, RFC 2460, “Internet Protocol, Version 6 (IPv6) Specification”, S. Deering, R. Hinden, Decembre 1998 [9]

- 
- [12] IETF, RFC 2475, "An Architecture for Differentiated Services", S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang. W. Weiss, December 1998. [10]
- [13] James Kurose, Keith Ross, *Analyse structurée des réseaux*, Pearson Education, 2003 [10]
- [14] IETF, RFC 2598, "An Expedited Forwarding PHB", V. Jacobson, K. Nichols, K. Poduri, June 1999 [11]
- [15] IETF, RFC 2597, "Assured Forwarding PHB Group", J. Heinanen, Telia Finland, F. Baker, W. Weiss, J. Wroclawski, June 1999 [11]
- [16] 3GPP 24-008, "Mobile radio interface layer 3 specification, Core network protocols, stage3." [14]
- [17] Farid Nait Abdesselam, "Gestion de la Qualité de Service des Applications Multimédias Distribuées : Une approche par Agents Logiciels", Thèse de Doctorat de l'Université de Versailles, Janvier 2000 [6]
- [18] Maniatis, S.I.; Nikolouzou, E.G.; Venieris, I.S.; "QoS issues in the converged 3G wireless and wired networks", Communications Magazine, IEEE , Volume: 40 , Issue: 8 , Aug. 2002, Pages:44 - 53 [20, 21]
- [19] Agharebparast, F.; Leung, V.C.M.; "QoS support in the UMTS/GPRS backbone network using DiffServ", Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE , Volume: 2 , 17-21 Nov. 2002 Pages:1440 - 1444 vol.2 [20, 21]
- [20] Borosa, T.; Marsic, B.; Pocuca, S.; "QoS support in IP multimedia subsystem using DiffServ", Proceedings of the 7th International Conference on Telecommunications, 2003. ConTEL 2003. , Volume: 2 , 11-13 June 2003, pages:669 - 672 vol.2 [20, 21]
- [21] Choukair, Z.; Sfar, S.; "Run time adaptation of UMTS services to available resources", 17th International Conference on Advanced Information Networking and Applications, 2003. AINA 2003. 27-29 March 2003, Pages: 33 - 38 [20, 21]
- [22] Gyasi-Agyei, A.; "Performance analysis of a differentiated services over wireless links", 5th IEEE International Conference on High Speed Networks and Multimedia Communications, 3-5 July 2002, Pages:86 - 90 [21]

- [23] Illia Racunina, Michelle Wetterwald, Yan Moret, "IP QoS Class conversion into Quality of Service Information Element (UMTS)", Task Internal Report, Avril 2003 [22]
- [24] Neda Nikaein, Illia Racunica, Christian Bonnet, "QoS Architecture ", rapport RNRT SAMU SP2 D2, février 2001 [7]
- [25] 3GPP 22105, "Services and Service Capabilities" [23]
- [26] 3GPP 25301, "Radio Interface Protocol Architecture" [28]
- [27] 3GPP 25323, "Packet Data Convergence Protocol (PDCP) Specification" [29]
- [28] 3GPP 25331, "RRC Protocol Specification" [28]
- [29] 3GPP 25322, "RLC Protocol Specification" [29, 31]
- [30] Xiao Xu; Yi-Chiun Chen; Hua Xu; Gonen, E.; Peijuan Liu; "Simulation analysis of RLC timers in UMTS systems", Proceedings of the Winter Simulation Conference, 2002. , Volume: 1, Pages: 506 - 512, 8-11 Dec. 2002 [31, 40]
- [31] Xiao Xu; Yi-Chiun Chen; Hua Xu; Gonen, E.; Peijuan Liu; "Performance analysis on the radio link control protocol of UMTS system", Vehicular Technology Conference, Volume: 4 , 24-28 Sept. 2002, Pages:2026 - 2030 vol.4 [40]
- [32] 3GPP 25321, "MAC Protocol Specification" [36]
- [33] 3GPP 25222, "Multiplexing and channel coding (TDD)" [37]
- [34] Illia Racunica, Yan Moret, Christian Bonnet, "Simulations des HARQ1 sur les couches RLC et physique de l'UMTS", CFIP, mai 2002, page 275-290. [38, 40]
- [35] Illia Racunica, Yan Moret, Christian Bonnet, "Simulation of the Hybrid ARQ", rapport RNRT SAMU, fevrier 2002 [40]
- [36] Christian Bonnet, Yan Moret, Illia Racunica, "Link Quality Control for UMTS TDD", rapport RNRT SAMU SP2 D6, septembre 2002 [40]
- [37] Qinqing Zhang; Hsuan-Jung Su;, "Methods for preventing protocol stalling in UMTS radio link control", IEEE International Conference on Communications, 2003. ICC '03. Volume: 3 , 11-15 May 2003, Pages:2246 - 2250 vol.3 [40]

- [38] Lin, Costello and Miller, "Automatic Repeat request error control schemes, a survey of various ARQ and hybrid ARQ schemes, and error detection using linear block code", IEEE Commun. Mag., 1984, 2é, (12) [38]
- [39] 3GPP 25835, "Report on Hybrid ARQ Type II/III" [39]
- [40] Illia Racunica, Yan Moret, Christian Bonnet, "Performance of a link adaptive system in UMTS networks", WISICT, Janvier 2004 [40]
- [41] Neda Nikaein, "QoS Control in Wireless IP Access Networks", Thèse ENST, Eurecom, octobre 2000 [97]
- [42] Enderle, N.; Lagrange, X.; "Radio link control-acknowledged mode protocol performance modeling in UMTS", 4th International Workshop on Mobile and Wireless Communications Network, 2002. 9-11 Sept. 2002, Pages:332 - 336 [58]
- [43] Qinqing Zhang; Hsuan-Jung Su; "Performance of UMTS radio link control" IEEE International Conference on Communications, 2002. ICC 2002. Volume: 5 , 28 April-2 May 2002 Pages:3346 - 3350 vol.5 [58]
- [44] 3GPP 26 101, "AMR Speech Codec Frame Structure" [58]
- [45] Gutierrez, Wigard, Andersen, Damgaard and Mogensen, "Performance of Link Adaptation in GPRS Networks", IEEE VTC 2000, Vol. 2, Page(s): 492-499, [65]
- [46] Hagenauer, J.; "Rate-compatible punctured convolutional codes (RCPC codes) and their applications", IEEE Transactions on Communications, Volume: 36 , Issue: 4 , April 1988, Pages:389 - 400 [65]
- [47] G. Vivier.; "Retransmission hybride : une solution efficace pour TCP dans un environnement sans fil", Thèse de l'Université de Paris 6, juillet 2003 [39]
- [48] Balachandran, Ejzak, Nanda, Vitebskiy, Seth. "GPRS-13 : High Rate packet Data Service for North American TDMA Digital Cellular Systems", IEEE Personal Communication, June 1999 [61]
- [49] Molkdar, D.; Featherstone, W.; Larnbotharan, S. "An overview of EGPRS: the packet data component of EDGE", Electronics and Communication Engineering Journal, Volume: 14, Feb. 2002 [65]

- 
- [50] Ahmed H. Hmimy, Someshwar C. Gupta , “Throughput analysis of ATM traffic over satellite links using ARQ and adaptive concatenated FEC codes” Emerging Technologies Symposium: Broadband, Wireless Internet Access, 2000 [63, 65]
- [51] Bolot J.-C., Fosse-Paris, S., Towsley, D., “Adaptive FEC-based error control for Internet telephony”, INFOCOM '99, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE, Volume: 3 , 1999 Page(s): 1453 -1460 vol.3 [65]
- [52] Khun-Jush, Malmgren, Schramm, Torsner. “Hipelran type 2 for broadband wireless communication” Ericsson review No2, 2000 [64, 65]
- [53] Jumi Lee; Iickho Song; So Ryoung Park; Seokho Yoon “Analysis of an adaptive rate convolutionally coded multicarrier DS/CDMA system”, Transactions on Vehicular Technology, Volume: 50 Issue: 4 ,July 2001 Page(s): 1014 -1023 [65]
- [54] Joachim Hagenauer , “ Rate-compatible punctured convolutional codes (RCPC codes) and their Applications”, IEEE transaction on communications Vol 36, N4, April 1988 [65]
- [55] S. Kumar and S. Nanda. “High Data Rate Packet Communication for Cellular Networks using CDMA: Algorithm and Performance”, IEEE JSAC, Mars 1998 [65]
- [56] 3GPP25302, “Services provided by the physical layer” [56]
- [57] 3GPP 25944, “Channel Coding and Multiplexing examples” [57]
- [58] Kazmi, M.; Godlewski, P.; Cordier, C.; “Admission control strategy and scheduling algorithms for downlink packet transmission in WCDMA”, Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd ,Volume: 2 , 24-28 Sept. 2000. Pages:674 - 680 vol.2 [73]
- [59] H. Holma, A. Toskala, “WCDMA for UMTS, Radio Access for Third Generation Mobile Communications”, *Wiley*, 2000 [74]
- [60] 3GPP 25221, “Physical channels and mapping of transport channels onto physical channels (TDD)” [78, 79]
- [61] 3GPP25223, “Spreading and modulation (TDD)” [78]

- [62] Mihailescu, C.; Lagrange, X.; Godlewski, P.; "Performance evaluation of a dynamic resource allocation algorithm for UMTS-TDD systems" Vehicular Technology Conference Proceedings, 15-18 May 2000, Pages:2339 - 2343 vol.3 [79]
- [63] Fraile, R.; Cardona, N.; "Resource allocation strategies for the TDD component of UMTS" The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002. Volume: 1 , Pages:21 - 25, 15-18 Sept. 2002, [-]
- [64] Ferracioli, M.; Tralli, V.; Verdone, R.; "Channel based adaptive resource allocation at the MAC layer in UMTS TD-CDMA systems" Vehicular Technology Conference, 2000. Pages:2549 - 2555 vol.6, 24-28 Sept. 2000 [-]
- [65] Lehtinen, O.; Kurjenniemi, J.; "UTRA TDD dynamic channel allocation in uplink with slow reallocation" Vehicular Technology Conference, 2003. Pages:1401 - 1405 vol.2, 22-25 April 2003 [82]
- [66] Nasreddine, J.; Lagrange, X.; "Time slot allocation based on a path gain division scheme for TD-CDMA TDD systems" Vehicular Technology Conference, 2003. Pages:1410 - 1414 vol.2, 22-25 April 2003 [79]
- [67] Ammari, S.; Wautier, A.; Braik, A.; Gattoufi, L.; "Multiple access interference model in a TD-CDMA system for capacity evaluation and throughput optimization" Vehicular Technology Conference, 2002. Pages:2298 - 2302 vol.4 , 24-28 Sept. 2002 [-]
- [68] Conti, M.; Gregori, E.; "Traffic and interference adaptive scheduling for Internet traffic in UMTS" Seventh International Symposium on Computers and Communications, 2002. Proceedings. ISCC 2002. , 1-4 July 2002 Pages:391 - 396 [-]
- [69] Forkel, I.; Kriengchaiyapruk, T.; Wegmann, B.; Schulz, E.; "Dynamic channel allocation in UMTS terrestrial radio access TDD systems" Vehicular Technology Conference, 2001. IEEE VTS 53rd , Volume: 2 , Pages:1032 - 1036 vol.2, 6-9 May 2001 [-]
- [70] Capone, A.; Cesana, M.; D'Onofrio, G.; Fratta, L.; "Mixed traffic in UMTS downlink", IEEE Microwave and Wireless Components Letters, Volume: 13 , Issue: 8 , Aug. 2003, Pages:299 - 301 [83]

- [71] Layer 2 and layer 3 of UTRA-TDD Gessner, C.; Kohn, R.; Schniedenharn, J.; Sitte, A.; IEEE 51st Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. Volume: 2 , 15-18 May 2000, Pages:1181 - 1185 vol.2 [83]
- [72] Ching Yao Huang; Yates, R.D.; "Call admission in power controlled CDMA systems" Vehicular Technology Conference, 1996. 'Mobile Technology for the Human Race', 28 April-1 May 1996, Pages:1665 - 1669, Volume: 3 [82]
- [73] Knutsson, J.; Butovitsch, P.; Persson, M.; Yates, R.D.; "Downlink admission control strategies for CDMA systems in a Manhattan environment" Vehicular Technology Conference, 1998. VTC 98. 18-21 May 1998, Pages:1453 - 1457, vol.2 [82]
- [74] Zhao Liu; El Zarki, M.; "SIR-based call admission control for DS-SS-CDMA cellular systems" IEEE Journal on Selected Areas in Communications, , Volume: 12 , Issue: 4 , May 1994, Pages:638 - 644 [82]
- [75] Christian Bonnet, Isabelle Demeure, "Introduction aux systèmes temps réel" 1999, Hermes [94]
- [76] Canton, A.-F.; Tohme, S.; Zeghlache, D.; Chahed, T.; "Statistical analysis of weighted round robin service differentiation at AAL2 layer in UMTS radio access network", Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE , Volume: 2 , 17-21 Nov. 2002 Pages:1554 - 1559 vol.2 [94]
- [77] Yonghe Liu; Gruhl, S.; Knightly, E.W.; "WCFQ: an opportunistic wireless scheduler with statistical fairness bounds" IEEE Transactions on Wireless Communications, Volume: 2 , Issue: 5 , Sept. 2003, Pages:1017 - 1028 [94, 95]
- [78] Illia Racunica, Aawatif Menouni, Christian Bonnet, "An opportunistic scheduling with fairness for NRT traffic in presence of RT traffic for UMTS/TDD", Abano Terme, Italie, 12-15 Septembre 2004 [100]
- [79] B. Bensaou, K. Chan, and D. Tsang, "Credit-based fair queuing (CBFQ): A simple and feasible scheduling algorithm for packet networks," in Proc. IEEE ATM'97 Workshop, May 1997, pp. 589-594. [95]
- [80] T. Ng, I. Stoica, and H. Zhang, "Packet fair queuing algorithms for wireless networks with location dependent errors" in Proc. IEEE INFOCOM' 98, vol. 3, Mar./Apr. 1998, pp. 1103-1111. [95]

- [81] P. Goyal, H. Vin, and H. Chen, "Start-time fair queuing: A scheduling algorithm for integrated services" in Proc. ACM SIGCOMM'96, Palo Alto, CA, Aug. 1996, pp. 157-168. [95]
- [82] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case" IEEE/ACM Trans. Networking, vol. 1, pp. 344-357, June 1993. [95]
- [83] Jain Raj, "The art of computer systems performance analysis : techniques for experimental design , measurement , simulation , and modeling", *John Wiley & Sons*, 1991. [96]
- [84] 3GPP 25102, "UE Radio Transmission and Reception (TDD)" [100]
- [85] Aawatif Menouni, Illia Racunica, Christian Bonnet, "Access control, Code Allocation, and adaptive scheduling for UMTS-TDD", IEEE Workshop on Adaptive Wireless Networks (AWiN) (In conjunction with GLOBECOM 2004), Dallas, Texas, 3 décembre 2004 [104]
- [86] Illia Racunica, Aawatif Menouni, Christian Bonnet, "Access Control, flexible code allocation, and opportunistic scheduling for UMTS-TDD", 2004 International Conference on Software, Telecommunications and Computer Networks (Softcom), Split (Croatie) - Venice (Italie), 10-13 Octobre 2004 [100]
- [87] Jean-Pierre Ebert, Andreas Willig, "A Gilbert-Elliot Bit Error Model and the efficient Use in Packet Level Simulation", TKN Technical Reports Series, Berlin, mars 1999 [97]
- [88] E.O. Elliot, "Estimation of error rates for codes on burst-error channels.", Bell Systems Technical Journal, page 1977, September 1963. [97]
- [89] J.G. Proakis, *Digital Communications*, Mc Graw Hill, New York, USA, 3rd ed., 1995 [111]
- [90] R.H. Clarke, "A statistical Theory of Mobile-Radio Reception", pages 957-1000, Bell Syst. Tech J., July-August 1968. [111]
- [91] Zoya Popovic and Branko D. Popovic, *Introductory to Electromagnetics*, Prentice Hall, 2000. [115]

- [92] Lucien Boithias, *Propagation des Ondes radioélectriques dans l'environnement Terrestre*, Dunod, 1984 [115]
- [93] T.S. Rappaport, *Wireless Communications, Principles and Practice*, Prentice-Hall, New Jersey, 1996 [116]