

Université Abderrahmane Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Ecole Doctorale Réseaux et Systèmes Distribués



Mémoire

Présenté en vue de l'obtention du diplôme de Magistère en Informatique

Transmission par paquets de la vidéo en temps réel sur les
réseaux sans fils

PRÉSENTÉ PAR Rima BOUDJADJA

Le jury est constitué de

Président : Ait Saadi
Rapporteur : Abdelnasser Dahmani
Examineurs : Ali Mallit
Abdelkrim KHIREDDINE
Invité : Mohamed Azni

Remerciements

*Je tiens à exprimer mes plus sincères remerciements et ma profonde gratitude à Monsieur **Mohamed AZNI** et au professeur **Abdelnasser DAHMANI** pour m'avoir proposé cette thèse et m'avoir accompagné tout au long de ces deux années. Leurs aides, leurs encouragements et leurs disponibilités ont été déterminants pour la réalisation de ce mémoire.*

*Je voudrais remercier aussi monsieur **Kamel TARI** et tous le personnel administratif de l'école doctorale Resyd.*

*J'adresse également mes plus sincères remerciements aux Mr **Ali MELLIT**, **Ahmed AIT SAIDI** et **Abdelkrim KHIREDDINE**, qui m'ont fait l'honneur d'évaluer cette thèse et de participer au jury de soutenance.*

Je remercie tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail, par leur aide, leur encouragement ou par leur simple présence.

Je dédie ce travail à :

*Mes chers et tendres parents, **Taher** et **Zoubida**, sans vous, rien n'aurait pu être possible, que dieu vous garde pour moi et vous prête une longue vie pleine de santé et de prospérité.*

Mes s¹/₂urs et mes frères, merci de m'avoir soutenu tout au long de cette aventure, que dieu vous préserve.

Tous les membres de ma famille et à tous mes amis.

Table des matières

1	La compression vidéo	1
1.1	Introduction	1
1.2	Applications de la compression vidéo	1
1.2.1	Vidéo à la demande	1
1.2.2	Vidéo conférence	2
1.2.3	Transmission sans fil de la vidéo	2
1.3	Les défauts typiques de la compression vidéo	2
1.4	Généralités et terminologie	3
1.5	Principe de base de la compression vidéo	4
1.5.1	Redondance subjective	5
1.5.2	Redondance spatiale	6
1.5.3	La redondance temporelle	6
1.5.4	Redondance statistique	7
1.6	Normalisation de la compression vidéo	8
1.6.1	Les normes de l'UIT-T	8
1.6.2	Les normes de l'ISO/IEC	9
1.7	H.264/ MPEG AVC	10
1.7.1	Caractéristiques	10
1.7.1.1	les types d'image	10
1.7.1.2	Tranche ou Slice	11
1.7.1.3	profiles et niveaux	12
1.7.1.4	Format des données codées et l'adaptation au réseau	13
1.7.2	Architecture de la norme H.264	14
1.7.2.1	Encodeur H.264	14
1.7.2.2	Décodeur H.264	15
1.7.3	Les processus de la compression H.264	15
1.7.3.1	Division en macroblocs	15
1.7.3.2	Estimation et compensation de mouvements (Prédiction Inter)	16
1.7.3.3	Prédiction spatiale (Intra)	17
1.7.3.4	Transformée fréquentielle	17
1.7.3.5	Quantification	19
1.7.3.6	Processus de filtrage anti-bloc	19
1.7.3.7	Parcours des données du Macrobloc	20
1.7.3.8	Codage entropique	20
1.7.4	Appréciation des erreurs de compression	22
1.7.5	H.264 et le temps réel	22
1.7.6	Techniques de robustesse de la norme H.264/AVC	24

1.7.6.1	Slice flexible et slice redondante	24
1.7.6.2	La technique FMO (Flexible Macrobloc Ordering)	24
1.7.6.3	Partitionnement de données	24
1.7.6.4	Ensemble de paramètres de séquence	25
1.7.6.5	Sélection de l'image de référence	26
1.7.7	Détection et dissimulation d'erreur dans un flux H.264	26
1.8	Conclusion	27
2	Généralités sur la vidéo sans fil	29
2.1	Introduction	29
2.2	Canal de transmission	29
2.2.1	Canaux stationnaires	29
2.2.1.1	Canal gaussien	30
2.2.2	Canaux non stationnaires	31
2.2.2.1	Probabilité d'erreur sur un canal de Rayleigh	34
2.2.2.2	Modélisation stochastique du canal radio	35
2.2.3	Capacité du canal	36
2.3	Réseaux IP et communication vidéo	36
2.3.1	Contrôle de trafic TCP	37
2.3.2	La chaîne de transmission vidéo	38
2.3.2.1	Codage de la source	38
2.3.2.2	Codage de canal	39
2.3.2.3	Modulation	39
2.3.2.4	Canal de transmission	39
2.3.2.5	Démodulation	40
2.3.2.6	Décodage de canal	40
2.3.2.7	Décodage de la source	40
2.4	Environnement protocolaire	40
2.4.1	La couche physique	41
2.4.1.1	QoS dans la couche physique	41
2.4.2	La couche Mac	41
2.4.2.1	Accès au canal dans la couche MAC 802.11	41
2.4.2.2	Livraison fiable dans la couche MAC 802.11	44
2.4.2.3	QoS au niveau de la couche MAC 802.11	44
2.4.2.4	Le standard IEEE 802.11e	46
2.4.3	La couche réseau	46
2.4.3.1	Routage avec Qualité de Service	47
2.4.4	La couche transport	47
2.4.4.1	Contrôle de débit côté émetteur	48
2.4.4.2	Contrôle de débit côté récepteur	48
2.4.4.3	Contrôle de débit hybride	48
2.4.5	La couche application	49
2.4.5.1	Protocoles multimédia	49
2.4.5.2	Codage source solide	50
2.4.5.3	Gestion de la latence et la gigue	51
2.5	Conclusion	51

3	Les différentes approches de transmission vidéo H.264\AVC sur les réseaux sans fil	53
3.1	Introduction	53
3.2	Réseaux à commutation de paquets vidéo	53
3.3	Contrôle d'erreur par anticipation pour la transmission de paquets vidéos (FEC)	54
3.3.1	Généralités et terminologie	55
3.3.2	Code de Reed_Solomon	57
3.3.2.1	Technique de codage	58
3.3.2.2	Technique de décodage	59
3.3.3	Turbo_codes en blocs	60
3.3.4	Codes raccourcis	61
3.3.5	Codes convolutifs	61
3.3.5.1	La représentation en treillis	62
3.3.5.2	Algorithme de Viterbi	63
3.3.6	Les codes poinçonnés	63
3.3.7	Choix des codes correcteurs	65
3.3.7.1	Temps de codage/décodage compatible avec du temps réel	65
3.3.7.2	Grand choix de taux de rendement	65
3.3.7.3	Systématique	65
3.3.7.4	Utilisation possible de la connaissance de la position des pertes	66
3.4	Les FEC perceptuelles	66
3.5	La protection inégale	66
3.6	Codage par description multiple	67
3.7	Codage en couches	68
3.8	Data Interleaving	68
3.9	ARQ : Demande de retransmission automatique	69
3.10	Approche inter-couches (Cross layer)	69
3.10.1	Les approches du Cross-layer dans les réseaux sans fil	70
3.10.1.1	Les approches ascendantes	71
3.10.1.2	Les approches descendantes	71
3.10.1.3	Les approches mixtes	72
3.11	Codage conjoint source-canal (JSCC)	72
3.11.1	Equilibrage des rendements	73
3.11.2	Décodage conjoint source-canal	75
3.11.2.1	Méthode de décodage séparé	76
3.11.2.2	Méthodes de décodage avec estimation parfaite de la source	76
3.11.2.3	Méthodes de décodage avec estimation paramétrique de la source	77
3.11.2.4	Le schéma de décodage MAP	77
3.12	Conclusion	79
4	Codage conjoint source canal utilisant la UEP et les codes RS	81
4.1	Introduction	81
4.2	Diagramme en bloc du système proposé	81
4.2.1	Le codage de source proposé	82
4.2.2	Allocation de débit de canal	84

4.2.2.1	Algorithme d'affectation de rendement	84
4.2.3	Le codage de canal proposé	87
4.2.4	Ajout de bruit	88
4.2.5	Décodage de canal	89
4.2.6	Le décodage de source	90
4.3	Test de la méthode proposée	90
4.3.1	Environnement de simulation	90
4.3.1.1	Description du CoDec RS utilisé	91
4.3.2	Résultats de simulation	93
4.3.2.1	Utilisation de la protection égale	94
4.3.2.2	Performance de la Méthode Proposée en taux d'erreur bi- naire	95
4.3.2.3	Performance de la Méthode Proposée en taux de perte de paquets	97
4.3.2.4	PSNR et qualité visuelle	100
4.4	Conclusion	101
Conclusion Générale		i
Appendices		1
A Champs de Galoi		3
A.1	Groupe	3
A.2	Anneau	3
A.3	Corps ou champs	4
A.4	Champs de Galoi	4
A.5	Les opérations dans le champs de Galoi	4
A.5.1	Polynômes primitifs	4
A.5.2	Construction d'un $GF(2^q)$	4

Table des figures

1.1	Les trois formats 4.2.0, 4.4.2 et 4.4.4 de la norme H.264	6
1.2	Redondance temporelle	7
1.3	Illustration de la compensation de mouvement basée sur les blocs	7
1.4	Comparaison du débit binaire obtenu par H.264 avec les autres normes vidéo	10
1.5	La relation entre les différents types d'image	11
1.6	La syntaxe d'une tranche	12
1.7	La relation entre les trois profils	13
1.8	Séquence d'unité NAL	14
1.9	Shéma du codec H.264	14
1.10	Le décodeur H.264	15
1.11	Les différentes partitions et sous-partitions de macroblocs dans H.264	16
1.12	Exemple de prédiction entière et sous-pixel	17
1.13	Les modes de prédiction Intra des bloc de 4×4	18
1.14	Le chemin de parcours des données du bloc	20
1.15	Allure typique d'une courbe du rapport débit / distortion	23
1.16	Exemples d'utilisation de FMO	25
2.1	Le modèle d'un système de transmission	30
2.2	Performances des modulations binaires antipodales.	32
2.3	Classification de canaux de transmission	32
2.4	Le canal de rayleigh	33
2.5	Illustration de propagation multitrajets	33
2.6	Loi de rayleigh et son taux d'erreur	35
2.7	Modèle de Markov à deux états	35
2.8	Diagramme d'une chaîne de transmission classique	38
2.9	L'accès distribué au canal	42
2.10	L'accès centralisé au canal	43
3.1	Le schéma général de Reed-Solomon	57
3.2	Mot-code de Reed-Solomon	58
3.3	Principe du codage d'un code produit	61
3.4	Exemple de codage convolutif	62
3.5	La représentation en treillis d'un code	63
3.6	Algorithme de Viterbi	64
3.7	Les modèles de communication Cross-Layer	70
3.8	Les approches du cross-layer	71
3.9	Les architectures JSCC classique	74
3.10	Le paradigme de SHANNON	74
3.11	Turbo-décodage avec information à priori sur la source	79

4.1	L'architecture du système <i>JSCC</i> utilisé	82
4.2	Diagramme en bloc du système proposé	83
4.3	Classification des 9 NALUs d'une image	85
4.4	La taille du bloc de NALUs de type 2 par rapport à la somme des blocs	85
4.5	Variations du TEB selon la valeur du E_b/N_0	89
4.6	Variation du taux de perte de paquets utilisant les code RS	93
4.7	Le vecteur moyen de rendement	94
4.8	Variation du TPP des trois partitions	95
4.9	TEB avant et après la EEP et la UEP	96
4.10	Comparaison du TEB de la UEP et EEP	97
4.11	Le taux de perte de paquets A pour les deux méthode UEP et EEP	98
4.12	Le taux de perte de paquets B pour les deux méthode UEP et EEP	98
4.13	Le taux de perte de paquets C pour les deux méthode UEP et EEP	99
4.14	Le taux de perte effectif des partitions B et C	99
4.15	Variation du TPP global après l'EC pour la UEP et la EEP	100
4.16	La variation du PSNR	100

Liste des tableaux

1.1	Les mots de code Exp-Golomb	21
3.1	Étapes de codage	63
3.2	Exemple de matrice de poinçonnage	65
4.1	Environnement de simulation	91
4.2	Les paramètres utilisés dans la phase de codage H.264	93
4.3	Le vecteur de rendement de la 24 ^{ème} image de la séquence crc1	94
A.1	Polynômes primitifs dans $GF(2^m)^4$	5
A.2	Les éléments de $GF(2^4)$	5

Liste des acronymes

ISO/IEC	International Standards Organisation/ International Electrotechnical Commission
IP	Intenet Protocol
JPEG	Joint Photographic Experts Group
MJPEG	Motion JPEG
MPEG	Motion Picture Expert Group
JVT	Joint Video Team
LLC	Logical Link Control
MAC	Media Access Control
MAQ	Modulations d'Amplitude en Quadratures
NAL	Network Abstraction Layer
PSNR	Peak Signal to Noise Ratio
PCF	Point Coordination Function
QCIF	Quarter Common Intermediate Format
RBSP	Raw Byte Sequence Payload
RGB	Red-Green-Blue
GOP	Group Of Pictures
GF	Galois Field
RS	Redundant Slices
FMO	Flexible Macrobloc Ordering
FEC	Forward Error Control Perceptuelle
FEC	Forward Error Control
Exp-Golomb	Exponential Golomb
EQM	Erreur Quadratique Moyenne
EEP	Equal Error Protection
EC	Error Concealment
DIFS	Distributed InterFrame Space
SIFS	Source Intermediate Format Short
DCT	Discrete Cosine Transform
DCF	Distributed Coordination Function
DC	Direct Current
Codec	COmpression / DECompression
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CAVLC	Context Adaptative Variable Length Coding
CABAC	Adaptative Binary Arithmetic coding

B	Bidirectionnelle
I	Intra
P	Prédite
SI	Switching I
SP	Switching P
AWGN	Average White Gaussian Noise
AVC	
ARQ	Automatic Repeat Request
AR	Admissibility Region
AODV	Ad Hoc On Demand Distance Vector
AIFS	Arbitration IFS
AGB	Additif Gaussien Blanc
ACK	Acknowledgement
ACC	Algorithme de Contrôle de Congestion
AACA	Adaptive Acquisition Collision Avoidance
MAP	Maximum A Posteriori
MDP2	La Modulation de Phase à deux états
MPLS	Multi Protocol Label Switching
MTU	Maximum Transit Unit
QOS	Quality of Service
RAS	Registration, Admission, and Status
RCP/RTCP	Real-time Transport Protocol/Real-Time Control Protocol
RS code	Reed Solomon code
TCP	Transmission Control Protocol
UDP	Usage Datagram Protocol
UEP	Unequal Error Protection
ITU	International Telecommunication Union
VLC	Variable Length Coding
VCL	Video Coding Layer
ASO	Arbitrary Slice Ordering
MDC	Mutiple Description Coding
ECN	Explicit Congestion Notification
JSCC	Joint Source Channel Coding
DCSC	Décodage Conjoint Source Canal
RCPC	Rate Compatible Punctured Codes

Introduction générale

Les applications de transmission de la vidéo numérique sur les canaux sans fil présentent, de nos jours, un enjeu important dans le domaine des transmissions de signaux multimédia. Ces systèmes de communications, combinés avec le développement rapide de la capacité de calcul des ordinateurs et des techniques de compression et de transmission de données, permettent d'accroître de manière significative le nombre d'utilisateurs. Par exemple, *Youtube* et *Dailymotion*, deux sites web de streaming vidéo, rassemblent à eux seuls des centaines de millions d'utilisateurs. Ainsi, il est, par exemple, possible aujourd'hui de télécharger la bande annonce de notre film favori ou de voir le bulletin d'information directement sur le net.

La technique de la visioconférence permet à des participants d'assister à des conférences même s'ils sont très distants (*Skype* étant un représentant très populaire conçu pour ces applications). Ce type de communication vidéo privilégie l'aspect interactif ou temps réel, mais pose souvent des contraintes sévères, tant sur le transmetteur que sur la qualité de services qui doit être offerte par les réseaux de communication. Les téléphones cellulaires 3G présentent un exemple idéal pour l'utilisation de la vidéo sans fil ; ils sont capables d'enregistrer, de visionner et d'échanger des séquences vidéos. Les contraintes qui se posent au niveau du transmetteur sont dues au volume important d'informations qui sont générées par les sources vidéo et dont les signaux sont donc inutilisables à l'état brut, car requérant une bande passante très grande devant la capacité des canaux de transmission existants. Des techniques de compression sophistiquées doivent alors être adoptées. Au niveau du transport de la vidéo, les applications en temps réel doivent faire face aux contraintes sur les délais d'acheminement de l'information de l'émetteur jusqu'au récepteur.

Les techniques de compression vidéo se sont considérablement développées durant les dernières années et plusieurs normes de codage vidéo ont été adoptées. Ces standards visent une vaste gamme d'applications qui exigent une grande variété de débits binaires pour répondre à différentes qualités désirées des images décodées. La norme la plus récente, appelée H.264/AVC, vise à améliorer les standards précédents, en apportant de nombreuses nouveautés. Le rendement de codage a été considérablement amélioré dans cette norme et plusieurs outils, destinés à un codage robuste aux erreurs de transmission, y ont été introduits, en prévision de leurs utilisations dans des applications particulièrement contraignantes.

La transmission des paquets vidéo sur un lien sans fil doit considérer les paramètres principaux qui caractérisent ce type de réseaux. Il est souvent nécessaire de procéder à une régulation de débit pour adapter le débit source à la capacité du canal de transmission. De plus, dans un canal sans fil, des pertes de paquets peuvent avoir lieu due à la congestion dans les réseaux ou dû au taux d'erreur binaire qui est très important. Le délai de bout-en-bout et la gigue sont aussi importants à considérer dans les applications à temps réel. Durant ces dernières années, plusieurs techniques ont été utilisées par les applications multimédia afin de palier aux variations de ces facteurs et de minimiser leurs effets sur

la qualité de la vidéo perçue par le récepteur. Une première approche consiste à ajouter dans le réseau des mécanismes pour pouvoir assurer différentes qualités de service (QoS) avec éventuellement des garanties. A ce sujet, l'IETF (Internet Engineering Task Force) a fait de nombreuses études pour introduire la nouvelle architecture DiffServ (Differentiated Services). Le support de la qualité de service nécessite la mise en oeuvre dans le réseau de plusieurs mécanismes d'ordonnancement, de gestion des files d'attente et de contrôle de trafic. Une seconde approche consiste à transmettre des paquets redondants, qui permet de récupérer les paquets erronés ou perdus. Ce type de redondance peut être introduit à la source ou par un codeur de canal.

Notre objectif dans ce mémoire est de définir un mécanisme capable d'assurer une qualité acceptable de la vidéo reçue pour les applications de transmission de la vidéo en temps réel, dans un environnement sujet aux problèmes de perte de paquets et de délais de transmission caractérisant les réseaux sans fil. Notre proposition consiste en une technique de codage conjoint source canal basée sur l'utilisation en amont d'un codage correcteur d'erreurs associé à une protection inégale des paquets vidéo contre les erreurs de transmission. Notre solution est conçue pour être compatible avec la sémantique d'un flux vidéo codé en norme H.264/AVC.

Le présent mémoire est structuré en quatre chapitres, plus une introduction générale et une conclusion générale.

Le premier chapitre est consacré à la présentation des généralités sur la compression vidéo. Les principes de cette technologie et un historique des différents standards qui ont été publiés par les deux organismes UIT-T et ISO/IEC sont examinés dans ce premier chapitre. La dernière section est dédiée à l'architecture générale du CODEC (Codeur/Décodeur) dans la norme H.264/AVC. Les spécificités de cette norme y sont décrites en détails et, particulièrement, les outils de codage innovants qui sont inclus dans cette norme et qui ont été introduits pour assurer un codage robuste sont mentionnés.

Dans le deuxième chapitre, une étude des modèles de canaux de transmission usuels est proposée. Les modèles de Gauss et de Rayleigh sont examinés. Le canal Gaussien est adapté aux réseaux filaires alors que le modèle de Rayleigh décrit mieux les propriétés des canaux sans fil. On s'intéressera, dans la suite de ce chapitre, aux systèmes de communication vidéo et aux réseaux de transmission, ainsi que l'environnement protocolaire qui les accompagne.

Le troisième chapitre est consacré à une description des différentes techniques de transmission vidéo permettant de minimiser la perte de paquets et leur impact sur la qualité de la vidéo décodée. Ces techniques peuvent être classées en fonction de la couche dans laquelle elles sont mises en place. Notons, toutefois, que la plupart des systèmes proposés dans la littérature sont des combinaisons de plusieurs techniques opérant au niveau des différentes couches du modèle de référence OSI (Open Systems Interconnection).

Une présentation des étapes de la technique proposée ainsi qu'une évaluation de ses performances est présentée dans le quatrième chapitre. Elle s'appuie sur un algorithme d'optimisation débit-distorsion du débit de canal selon le type et la taille des paquets à transmettre. Le codage de canal, adapté aux données inégalement protégées, est effectué au niveau de la couche application en utilisant les codes de Reed-Solomon (RS).

Enfin, une conclusion générale et des perspectives termineront ce document.

Chapitre 1

La compression vidéo

1.1 Introduction

Ce travail étant axé sur la transmission vidéo sur réseaux sans fil. Ces réseaux sont caractérisés par leurs bandes passantes très limitées, par rapport aux réseaux filaires, et par leurs grand taux de perturbations. D'où la nécessité de comprimer le flux vidéo avant la transmission.

Pour tenter à résoudre ce problème, plusieurs normes de codage vidéo ont été adoptées ; CoDec (COmpression / DECompression). Ces standards visent une vaste gamme d'applications qui exige une grande variété de débits binaires pour répondre à différentes qualités désirées des images décodées. La norme la plus récente appelée H.264 vise à améliorer les précédents standards en apportant des nombreuses nouveautés. Généralement, les outils utilisés dans la plupart des normes exploitent la redondance dans les séquences vidéo, qu'elle soit subjective, spatiale, temporelle ou statistique. Ainsi tout ce qui figure dans l'image précédente n'a pas besoin d'être transmis et peut être juste copié par le récepteur.

Ce chapitre sera en grande partie dédié à la compression vidéo et à l'algorithme phare de cette discipline, H.264. Pour cela, nous allons présenter, les outils de prédiction spatiale, estimation et compensation de mouvement, la DCT, la quantification et enfin le codage entropique permettant de réduire les quatre types de redondance qui existent dans une séquence vidéo.

1.2 Applications de la compression vidéo

1.2.1 Vidéo à la demande

La vidéo à la demande est une innovation majeure dans le quotidien des téléspectateurs. L'utilisateur peut choisir son propre programme et le regarder à n'importe quel moment. Autant d'utilisateurs libres de choisir leur propre programme génère indubitablement un trafic énorme sur le réseau. D'où l'absolue nécessité d'utiliser des algorithmes de compression très performants pour répondre à la demande.

Par performances de l'algorithme nous sous-entendons aussi bien, celles au niveau de la compression que celles au niveau de la complexité. En effet, l'algorithme doit être rapidement exécuté à la décompression et cela sur des architectures peu puissantes pour éviter les ralentissements gênants pour l'utilisateur.

1.2.2 Vidéo conférence

La vidéo conférence est une méthode de communication qui connaît une augmentation d'utilisation très importante. Cette technique n'était pas très répandue pendant les dix dernières années; l'insuffisance des capacités de calcul et des débits de transmission ne permettaient pas une meilleure exploitation de cette technique. Mais avec les progrès récents dans ces deux domaines, l'utilisation de la vidéo conférence s'est de plus en plus démocratisée. Il s'agit d'une réception au vol. Une fois les données transmises, le temps d'émission et de réception sont décorrélés à cause de la bande passante et du retard de transmission. Après un certain temps de transition, l'audition ou la visualisation commence en même temps que le stockage des données transmises dans la mémoire tampon.

La performance de la vidéo conférence est toujours limitée par l'insuffisance de la capacité matérielle (vitesse de calcul et débit de transmission), en plus la fréquence n'est plus limitée à 30 images par seconde. Un problème majeur apparaît alors en ce qui concerne le temps réel. En effet, le temps prit pour la compression et la décompression (temps de latence) n'est plus négligeable et induit irrémédiablement à des ralentissements dans le flux vidéo. Dans la norme MPEG-4, ce problème a été considéré et tous les outils relatifs au temps réel ont été optimisés pour éviter les ralentissements.

1.2.3 Transmission sans fil de la vidéo

Les réseaux sans fils qui sont de plus en plus utilisés, se caractérisent par leurs bandes passantes très limitées par rapport aux réseaux filaires, et par leurs grand taux de perturbations. Ainsi, le schéma de compression vidéo destiné aux applications sans fils, ne doit pas seulement offrir un taux de compression assez élevé pour s'adapter aux limites de la bande passante, mais aussi il doit assurer une robustesse contre les erreurs du aux perturbations. La septième section de ce chapitre présente les outils offerts par la dernière norme de compression vidéo qui rendent un flux vidéo plus robuste.

Vu que la taille des équipements dans les réseaux sans fils est très varié, donc résolution variée, l'adaptation du schéma de compression à cette variation est très exigée. Pour cela la compression dans ce type de réseaux doit être aussi flexible que possible. Les téléphones cellulaires 3G (3eme génération) sont un exemple idéal pour l'utilisation de la vidéo sans fils. Cette génération de téléphone portable est capable d'enregistrer, visionner et échanger des séquences vidéo, en plus de toutes ces possibilités d'utilisation, les téléphones 3G offrent la possibilité de faire des conversations audiovisuelles, ou même de regarder la télévision sur l'écran du téléphone portable.

La transmission sans fils de la vidéo est utilisée récemment, dans les cameras de surveillances. Dans ces deux derniers cas, le schéma de compression doit présenter, en plus des caractéristiques expliquées précédemment, la possibilité de faire la compression en temps réel (complexité minimale).

1.3 Les défauts typiques de la compression vidéo

La compression engendre des dégradations trop poussée qui sont visuellement différentes des défauts pouvant affecter un signal analogique. Les principales sont les suivantes :

- Effet de bloc : une structure rappelant une mosaïque carrée apparaît à certains endroits de l'image; du même type que celle que l'on peut voir en parcourant en lecture rapide une cassette numérique. Ce sont les blocs de pixels (voir plus loin) qui deviennent visibles, à cause d'un taux de compression excessif.
- Effet de halo : une sorte de frange apparaît sur les contours des objets, particulièrement visible sur les textes incrustés, du fait d'un débit trop faible.
- Effet de blurring : les détails sont moins nets, les contours moins marqués, avec parfois un effet de traînée. Là encore, une compression trop élevée est en cause.
- Bruit de quantification : un effet de neige ou de vitre sale vient polluer l'image, mais uniformément. Ce défaut est généralement lié à un problème local de conversion analogique/numérique.
- Effet de moustique : du bruit apparaît sur les transitions d'éléments en mouvement, sous la forme de petit points noirs ou blancs qui miroitent autour des objets, rappelant des moustiques. Ce défaut est dû à des erreurs de quantification entre pixels voisins.

1.4 Généralités et terminologie

Vidéo, Image, tranche, Macro-bloc

la vidéo est une succession d'images animées. Son principe fondamental est que l'oeil humain a la faculté de retenir pendant un certain temps, de l'ordre du dixième de seconde, toute image imprimée sur la rétine. Il suffit donc de faire défiler un nombre suffisant d'images par seconde, pour que l'oeil ne se rende pas compte qu'il s'agit d'images distinctes.

Dans un flux vidéo numérique, les données sont hiérarchisées selon une manière bien précise. Tout d'abord, la séquence vidéo commence par un code de début de séquence, contient un ou plusieurs groupe d'images, et se termine par un code de fin de séquence. Chaque groupe d'image débute également par un entête, et comporte une ou plusieurs images.

Une image est formée d'une matrice rectangulaire de 144×176 pixels pour un format QCIF ¹ ou 288×352 pour un CIF. Chaque pixel est composé des trois couleurs primaires rouge, vert et bleu (RVB, ou RGB en anglais pour Red-Green-Blue). Pour obtenir des images en couleur, il faut utiliser au moins 8 bits par pixels, ce qui correspond à 256 couleurs. En fait, avec 8 bits par pixels, on obtient de la vidéo numérique noir et blanc de haute qualité. Pour la vidéo numérique couleur, on utilise 8 bits pour chaque couleur RVB, soit donc 24 bits par pixel, ce qui correspond à environ 16,8 millions de couleurs. Ces trois composantes ont pratiquement la même importance et ne peuvent être sous-échantillonnées. Pour cela, les signaux RVB sont combinés linéairement en un signal de luminance et deux signaux de chrominance; notée YUV, où Y est la luminance, U est la chrominance blanc-rouge et V la chrominance blanc-bleu. Ces deux notations sont

¹Quarter Common Intermediate Format

équivalentes, et on utilisera ici la deuxième, plus économique en place mémoire (voir section 1.5.1).

Chaque image est découpée en tranches ; Slices. Une tranche est constituée de un ou plusieurs macro-blocs, couvrant 16 x 16 pixels, adjacents ordonnés de gauche à droite. Ce sont des éléments importants pour la gestion des erreurs. Si une tranche contient une erreur, le décodeur passe alors immédiatement à la tranche suivante. Bien entendu, plus il y a de tranches dans une image, meilleur est le traitement des erreurs mais plus l'image prend de place également. Tous les traitements utilisés dans la compression vidéo et que nous allons citer dans ce chapitre sont appliqués sur les macro-bloc.

Vidéo entrelacée et vidéo progressive

Même avec un débit de 25 trames par seconde, certaines personnes pouvaient percevoir des papillotements. Pour résoudre ce problème, au lieu d'augmenter le débit des trames, ce qui exigerait plus de bande passante, une technique qui consiste à afficher d'abord les lignes impaires puis les lignes paires. Chacune de ces deux demi-trames est appelée champs. Les expérimentations ont montré que là où des gens notaient des papillotement à 25 trames par seconde, ils n'en notaient plus à 50 champs pas seconde. Cette technique est appelée l'entrelacement ; et on parle alors d'images entrelacées. On dit de la télévision ou de la vidéo non entrelacée qu'elle est progressive.

1.5 Principe de base de la compression vidéo

La compression vidéo est basée sur la suppression de la redondance dans les données vidéo afin de minimiser au maximum la quantité de données nécessaire pour la représentation des informations vidéo, et permettre à la vidéo numérique d'être sauvegardée et envoyée facilement.

Chaque image d'une séquence vidéo brute peut être caractérisée par quatre principaux paramètres : sa redondance subjective, sa redondance spatiale, sa redondance temporelle et sa redondance statistique.

La résolution en luminance détermine le nombre de nuances ou de couleurs possibles pour un pixel. Celle-ci est généralement de 8 bits pour les séquences en niveaux de gris et de 24 bits pour les séquences en couleurs. La résolution spatiale, définit le nombre de lignes et de colonnes de la matrice de pixels. Par contre, la résolution temporelle représente le nombre d'images par seconde.

L'espace mémoire nécessaire pour stocker les images de la séquence vidéo est calculé en fonction des valeurs de ces quatre paramètres. Cet espace est caractérisé par le débit, qui est le coût de stockage pour une seconde ; capacité mémoire pour stocker une seconde de la vidéo.

Le débit d'une séquence vidéo brute est très élevé comparé aux débits et à l'espace offerts par les algorithmes de compression vidéo. Dans ce qui suit, nous allons détailler les quatre types de redondance.

1.5.1 Redondance subjective

Dans un premier temps, une solution pour réduire ce débit est de sous-échantillonner les composantes de chrominance de la séquence. L'oeil humain étant plus sensible aux variations de luminance qu'à celles de chrominance, on utilise l'espace de représentation de couleurs YUV, en sous échantillonnant les composantes U et V.

Historiquement et techniquement, le format YUV est utilisé pour la transmission des signaux vidéos analogiques couleurs. Ces trois informations (Y,U,V) permettent de restituer au final les composantes RVB. Le passage de la représentation (R, V, B) à la représentation (Y, U, V) s'effectue par un simple produit matricielle. Il existe plusieurs matrices de transition selon les différentes recommandation de l'ITU-T. Une matrice de transition souvent utilisée est donnée par :

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & -0,100 \end{bmatrix} \times \begin{bmatrix} R \\ V \\ B \end{bmatrix} \quad (1.1)$$

Cette représentation, YUV ou YCbCr, apporte l'avantage de permettre une compression facile des couleurs et de viabiliser le transport du signal vidéo. Il existe trois formats de représentation de ces trois couleurs YUV selon le taux de réduction de la chrominance par rapport à la luminance.

Le sous-échantillonnage 4 :4 :4 signifie que les trois composantes (Y, Cb et Cr) ont la même résolution. Les nombres indiquent la fréquence relative d'échantillonnage de chaque composante dans le sens horizontal, c'est-à-dire pour tous quatre échantillons de luminance, il y a quatre échantillons U et quatre V. L'échantillonnage 4 :4 :4 préserve toute la fidélité des composantes de chrominance.

Dans l'échantillonnage 4 :2 :2 (parfois dénommé YUY2), les composantes de chrominance ont la même résolution verticale de la luminance, mais la moitié de la résolution horizontale (la notation 4 :2 :2 signifie que, pour tous quatre échantillons de luminance dans le sens horizontal il ya deux échantillons U et deux V). La vidéo 4 :2 :2 est utilisée dans la reproduction de couleur de haute qualité.

Dans l'échantillonnage 4 :2 :0, les composantes de chrominance sont réduits à la moitié de la résolution verticale et horizontale de la composante de luminance, soit 4 composantes de chrominance pour une composante U, et une composante V. La notation 4 :2 :0 n'a pas une interprétation logique, elle a été historiquement choisi pour identifier ce schéma d'échantillonnage et de le différencier de 4 :4 :4 et 4 :2 :2.

Ce sous-échantillonnage correspond à une réduction de la redondance psycho-visuelle. Dans la suite, nous étudierons les techniques permettant de réduire les redondances spatiales, temporelles et statistiques.

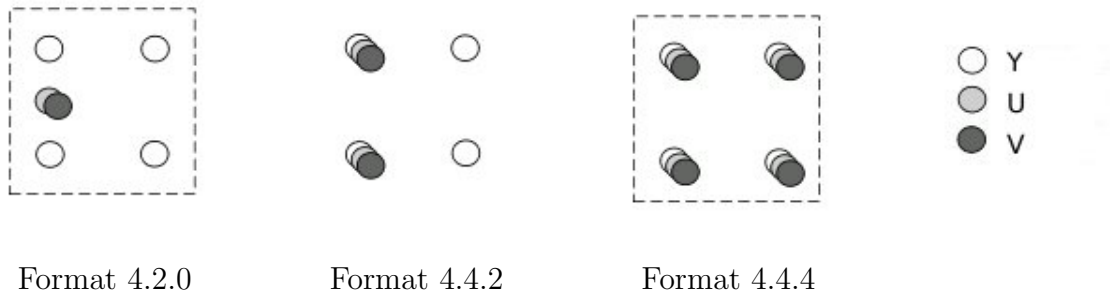


FIG. 1.1 – Les trois formats 4.2.0, 4.4.2 et 4.4.4 de la norme H.264

1.5.2 Redondance spatiale

Toute plage uniforme sur une image renferme des pixels identiques. Il est donc inutile de coder séparément chacun des pixels puisqu'un seul peut les caractériser tous. Il suffit de transmettre deux données, l'une représentant la valeur du pixel, l'autre étant le facteur de répétition. La Transformée en Cosinus Discrète (souvent notée DCT) peut mettre en évidence cette redondance spatiale à l'intérieur de chaque image.

Il est possible de diminuer cette redondance en codant chaque image séparément en JPEG (Joint Photographic Experts Group). Cette approche est parfois utilisée lorsqu'on a besoin de pouvoir accéder de façon aléatoire à chaque image individuellement, comme par exemple lors d'un montage vidéo. On parle alors de MJPEG (Motion JPEG).

1.5.3 La redondance temporelle

Dans une série d'images, il est possible de réduire le volume des données vidéo par des méthodes tel que le codage différentiel, qui est utilisé par la plupart des normes de compression vidéo, y compris H.264. Dans le codage différentiel, une image est comparée à une image de référence (une image I ou P antérieure) et seuls les pixels qui ont changé par rapport à l'image de référence sont codés. Cela permet de réduire le nombre de valeurs de pixels codées et envoyées.

La quantité de données encodée peut être réduite davantage encore si la détection et l'encodage des différences sont basés sur des blocs de pixels (macro-blocs) plutôt que sur des pixels individuels. Par conséquent, de plus grandes surfaces sont comparées et seuls les blocs nettement différents sont codés. Les données associées à l'indication de l'emplacement des zones à modifier s'en trouvent également réduites.

Cependant, le codage différentiel ne réduirait pas les données de manière significative s'il y a beaucoup de mouvement dans une vidéo. Dans ce cas, il est possible de recourir à des techniques telle que la compensation de mouvement basée sur les blocs. Cette technique tient compte du fait qu'une bonne partie de ce qui compose une nouvelle image dans une séquence vidéo se retrouve dans une image précédente, bien qu'à un emplacement potentiellement différent. Cette technique divise une image en série de macro-blocs. Il est possible de composer ou de prédire une nouvelle image (une image P, par exemple) bloc par bloc (bloc cible dans la figure 1.3) en recherchant un bloc identique (bloc correspondant) dans l'image antérieure. Si un bloc référent est trouvé, l'encodeur code tout simplement son emplacement dans l'image de référence ainsi que son déplacement. Le codage du

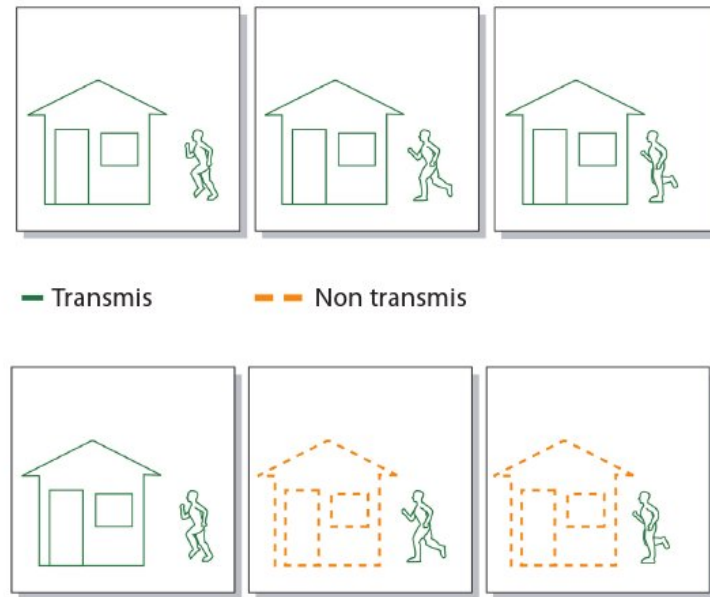


FIG. 1.2 – Redondance temporelle

vecteur de mouvement utilise moins de bits que le codage du contenu réel d'un bloc. Cette technique de mise en correspondance des bloc (bloc-matching) et appelée estimation et compensation de mouvement.

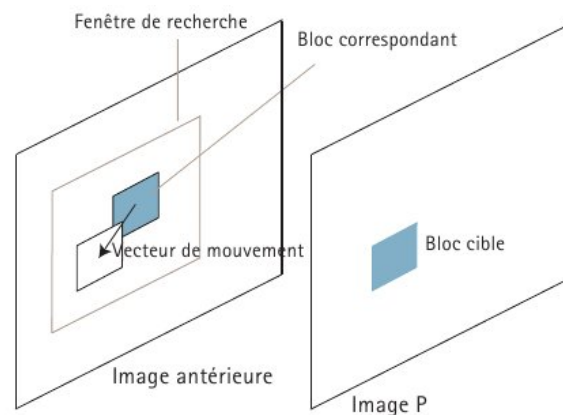


FIG. 1.3 – Illustration de la compensation de mouvement basée sur les blocs

1.5.4 Redondance statistique

Il s'agit d'une notion purement mathématique : si certains codes reviennent plus fréquemment que d'autres, autant leur réserver les mots les plus courts. Cette opération, appelée codage entropique, n'entraîne aucune perte.

Le codage de Huffman est l'algorithme le plus simple du codage entropique, il repose sur la création d'un arbre composé de noeuds où chaque branche représente le code binaire d'un symbole. La norme H.264 utilise trois types de codage, le premier est appelé Exp-Golomb (Exponential Golomb codes), il utilise une table universelle de mot de code. Cette table est utilisée pour coder la plupart des éléments de synchronisation comme les

entêtes. Les autres éléments syntaxiques (coefficients, vecteurs de mouvements) peuvent être codés par deux autres méthodes. Il s'agit d'une part d'un codage VLC adaptatif au contexte (CAVLC - Context Adaptive Variable Length Coding) et d'autre part d'un codage arithmétique adaptatif au contexte (CABAC, Context Adaptive Binary Arithmetic coding).

1.6 Normalisation de la compression vidéo

Il existe plusieurs algorithmes de compression vidéo normalisés, Les familles les plus importantes de ces algorithmes sont publiées par des organismes de normalisation.

Au niveau international, les deux organismes les plus actifs pour la normalisation des systèmes de compression vidéo sont l'UIT-T et l'ISO/IEC. Les travaux techniques de l'ISO/IEC sont menés au sein du groupe MPEG (Motion Picture Expert Group) qui a défini les standards MPEG-1, MPEG-2 et MPEG-4 pour des applications aussi variées que la télévision ou le multimédia. En parallèle des activités de MPEG, le groupe vidéo de l'UIT-T s'intéresse principalement à la définition de recommandations techniques destinées aux applications de visiophonie et de vidéoconférence (normes H.261 et H.263).

Ces deux organismes ont travaillé à l'élaboration d'une norme commune (H.264 / MPEG AVC) dont les performances obtenues permettent de réduire de moitié le débit de transmission ou de stockage pour une qualité visuelle équivalente aux normes précédentes.

Les codeurs vidéo normalisés (H.26x ou MPEGx) utilisent tous le même principe de base dont la connaissance aide à comprendre à la fois leurs potentialités et limites d'utilisation. Cette partie décrit les principaux standards normalisés à l'UIT-T et à l'ISO/IEC.

1.6.1 Les normes de l'UIT-T

Première norme vidéo approuvée, H.261 [E.G03] vise les applications de visiophonie pour le réseau RNIS à des débits multiples de 64 kbit/s. C'est une évolution de la norme H.120. Elle traite les images de format QCIF et CIF. Le schéma de codage de la norme H.261 est constitué de plusieurs modules, l'estimation et la compensation de mouvement, le calcul des résidus, la transformée fréquentielle DCT, la quantification et le codage entropique.

En 1995, la première version de H.263 [LHS07] apparaît. Elle a été développée pour la transmission de la vidéo sur des lignes à très bas débits, pour des applications de visiophonie via le réseau téléphonique commuté. Elle a ensuite été intégrée dans les protocoles de vidéoconférence sur IP. Cette norme repose sur les principes mis en place par la recommandation H.261. Les formats d'images sont des multiples et sous-multiples du CIF. Le codeur H.263 a la possibilité d'effectuer des compensations en mouvement avec une précision au demi-pixel, ce qui améliore grandement la qualité de la vidéo.

La dernière version de H.263 (2000), appelée H.263++ [LHS07], ajoute trois options et une spécification à la version antérieure. Outre l'amélioration en termes de qualité et

de taux de compression, elle prend mieux en compte la transmission vidéo temps réel sur des réseaux à qualité de service non garantie (IP et mobiles) .

1.6.2 Les normes de l'ISO/IEC

La première norme de compression vidéo développée par l'ISO/IEC, MPEG-1 [GJS05] est destinée à l'archivage des données vidéo numériques sur CD. La plupart des encodeurs et des décodeurs MPEG-1 sont conçus pour un débit d'environ 1,5 Mbit/s en résolution CIF. MPEG-1 met surtout l'accent sur le maintien d'un débit relativement constant, au détriment de la qualité d'image, laquelle est variable et comparable à la qualité vidéo VHS.

En 1994, une deuxième norme a été approuvée sous l'appellation MPEG-2 [Tud95]. destinée à la vidéo numérique de qualité supérieure (DVD), à la télévision haute définition (HDTV), aux supports d'enregistrement interactifs (ISM), aux systèmes d'émission vidéo numérique (DBV), et à la télévision par câble (CATV). Elle améliore la technique de compression de la norme MPEG-1 afin d'obtenir des images de résolution plus haute et de meilleure qualité, mais aux dépens d'un taux de compression plus faible et d'un débit d'image plus rapide. La fréquence est plafonnée à 25 images par seconde, tout comme en MPEG-1.

La norme MPEG-4 [TE00] a été spécifiée en octobre 1998 et publiée comme un standard international au début de l'année 2000. Elle représente une évolution substantielle par rapport au format MPEG-2. Cette norme contient de nombreux outils permettant de réduire le débit d'images de manière à préserver une certaine qualité pour une application ou une scène déterminée.

Hétérogénéité des codecs

La méthode utilisée pour réduire la quantité de données vidéo diffère d'un standard à un autre, ce qui explique la différence qu'il y a entre ces standards au niveau du taux de compression, qualité de la vidéo produite et temps de réponse. Cette différence devient un avantage dans certains cas, puisque les applications très variées de la vidéo numérique n'exigent pas toujours la même valeur pour ces trois caractéristiques. Cette différence existe aussi au niveau d'un même standard, puisque on peut adapter le codec à l'application souhaitée en choisissant de ne pas implémenter certaines options offertes par le standard. La seule contrainte qu'il faut satisfaire est d'assurer que le décodeur contient toutes les recommandations du standard, pour pouvoir décoder tout fichier vidéo compressé conforme au format du standard, même si différentes options ont été utilisées pour le codage.

Généralement, il n'y a pas de compatibilité entre les codecs vidéos de différents standards de compression, c'est à dire qu'on ne peut pas utiliser le codeur d'un standard pour la compression, et utiliser le décodeur d'un autre standard pour la décompression ; par exemple le décodeur MPEG-4 Part 2 [TE00] ne peut pas décoder une vidéo compressée grâce au codeur du standard H.264. Ce problème peut être corrigé grâce à la possibilité d'implémenter plusieurs codecs de différents standards dans le même logiciel ou matériel (puisque le codec peut être sous forme d'un Hardware ou Software), chose qui permettra à l'utilisateur de visionner des vidéos de format différent sans être obligé de changer à

chaque fois le dispositif utilisé [E.G03].

1.7 H.264/ MPEG AVC

La toute dernière norme de compression vidéo H.264 [E.G03][SkK05] est appelée à devenir la norme vidéo de référence au cours des prochaines années. Elle a déjà été intégrée avec succès dans des gadgets électroniques tels que les téléphones mobiles et les lecteurs vidéos numériques. Dans le secteur de la vidéo-surveillance, le H.264 offre de nouvelles possibilités en termes de réduction des frais de stockage et de renforcement de l'efficacité globale.

Le H.264, également connue sous l'appellation MPEG-4 Partie 10/AVC, est une norme compatible avec les techniques de compression vidéo les plus efficaces d'aujourd'hui. Un encodeur H.264 peut réduire la taille d'un fichier vidéo numérique de plus de 80 % par rapport à la norme Motion JPEG et de 50 % par rapport à la norme traditionnelle MPEG-4 Partie 2, sans que la qualité d'image ne soit compromise. L'importance de ces gains rend le H.264 extrêmement utile pour les applications de vidéo-surveillance.

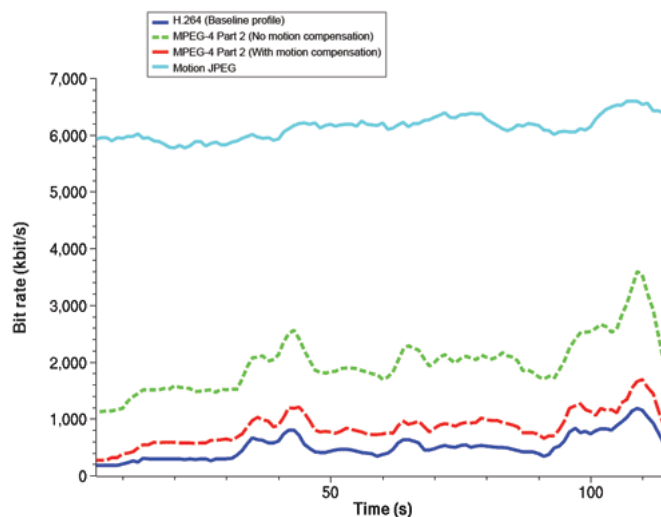


FIG. 1.4 – Comparaison du débit binaire obtenu par H.264 avec les autres normes vidéo

1.7.1 Caractéristiques

1.7.1.1 les types d'image

H.264 utilise les mêmes types d'images que dans les normes précédentes, I : prédiction spatiale (Intra), P : prédiction temporelle simple (Inter) et B : prédiction temporelle bidirectionnelle. En revanche, H.264 apporte deux nouveaux type d'images, SP et SI (switched I et P) [E.G03].

Les images I sont codées sans aucune référence à d'autres images, comme en JPEG, c'est-à-dire qu'elles contiennent tous les éléments nécessaires à leur reconstruction par le

décodeur et sont, de ce fait, le point d'entrée obligatoire lors de l'accès à une séquence. Le taux de compression des images I est relativement faible comparable à celui de JPEG avec perte. Par contre, les images P sont codées par rapport à l'image de type I ou P précédente, grâce aux techniques de prédiction avec compensation de mouvement. La compensation de mouvement n'étant pas parfaite, on ne pourra multiplier indéfiniment le nombre d'images P entre deux images I, car, étant utilisées pour coder d'autres images P ou B, elles propagent donc en l'amplifiant toute erreur de codage. Le taux de compression est nettement plus important que celui des images I.

Les images B sont codées par interpolation entre les deux images de types I ou P précédente et suivante qui les encadrent. Comme elles ne sont pas utilisées pour définir d'autres images, les images B ne propagent donc pas les erreurs de codage. C'est ce type d'images qui offre le taux de compression le plus fort. Selon la complexité du codeur utilisé, on pourra ne coder que des images I, des images I et P ou des images I,P et B, avec bien sûr des résultats très différents au niveau du taux de compression et des possibilités d'accès aléatoire, mais aussi de temps de codage et de la qualité perçue.

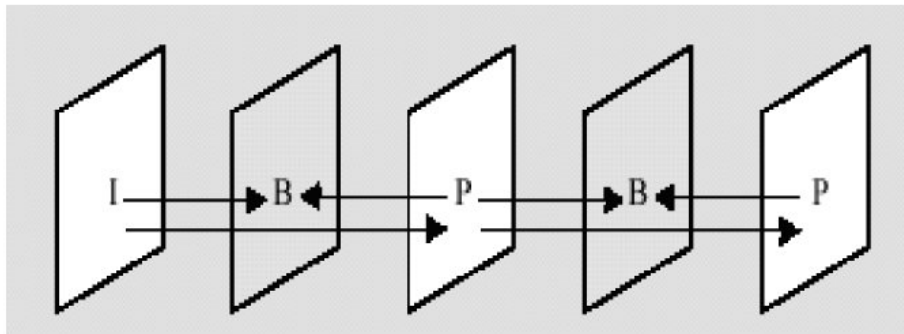


FIG. 1.5 – La relation entre les différents types d'image

Le quatrième type d'images apporté par cette norme, les images SP, viennent remplacer des images P, servent à coder la transition entre deux flux vidéo. Elles permettent, sans envoyer d'images Intra coûteuses en débit, de passer d'une vidéo à une autre. Ces Images peuvent être utilisées dans des contextes variés comme la transmission sur réseaux à QoS non garantie. Les images SI sont définies pour parvenir à une correspondance parfaite des image SP si la prédiction Intra-image ne peut être utilisée en raison d'erreurs de transmission. L'adaptation aux conditions du réseau est permise grâce à ce type d'image.

1.7.1.2 Tranche ou Slice

Une image vidéo est codée en une ou plusieurs tranche (Slice) [AKL03] où chacune contenant un certain nombre de macroblocs. Ce dernier n'est pas forcément fixe et dépend du degré de redondance de l'image codée. L'interdépendance entre les tranches codées est minimale, ce qui permet de limiter la propagation d'erreur. Il existe cinq types de tranches classifiées selon le type de ces macroblocs (I, P, B, SP et SI) et une image peut être codée en plusieurs tranches de différents types.

La figure 1.6 est une illustration simplifiée de la syntaxe d'une tranche codée. Le "header" définit le type de tranche et de l'image codée par cette dernière et peut con-

tenir des instructions relatives à la gestion de l'image de référence. La partie "data" est composée d'une succession de macroblocs codés.

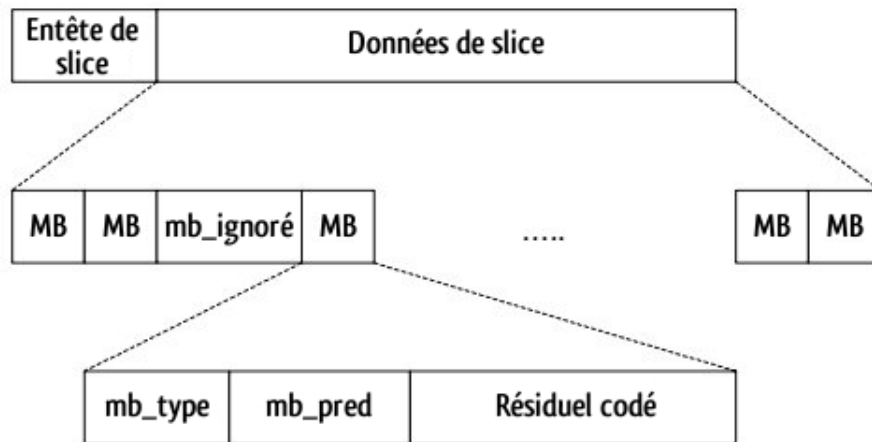


FIG. 1.6 – La syntaxe d'une tranche

Le codage prédictif rend les images dépendantes entre elles puisque lors du décodage, il est nécessaire d'avoir l'image précédente pour décoder l'image en cours. Cette dépendance est gênante si une image est perdue lors d'un transfert, le décodage du reste de la séquence devient impossible. C'est pourquoi le processus de compression regroupe les tranches en GOS (Group Of Slices) composés des trois types d'image.

Un GOS commence généralement par une image I qui constitue l'image-clé du GOS, cette image est suivie par une suite d'images P, intercalées par des images de type B. La fréquence des images P et la taille du GOP sont des paramètres qui influent sur le taux de compression et sur le temps de décodage. En effet, plus le GOS sera long, plus la corrélations entre les images sera importantes, mais en diminue la robustesse.

1.7.1.3 profiles et niveaux

Les profiles et niveaux spécifient des restrictions sur les flux binaires et donc des limites sur les capacités nécessaires pour décoder ces flux. Les profiles et niveaux [E.G03] peuvent aussi être utilisés pour indiquer les points d'inter-fonctionnement entre les implémentations de décodeurs individuels. Chaque niveau spécifie un ensemble de limites sur les valeurs que peuvent prendre les éléments syntaxiques de la présente norme internationale. Le même ensemble de définitions de niveau est utilisé avec tous les profiles, mais des implémentations individuelles peuvent accepter un niveau différent pour chaque profile. Pour tout profile donné, les niveaux correspondent généralement à la charge de traitement du décodeur et à la capacité de mémorisation. La norme H.264 définit trois profiles : de base, principal et étendu.

La figure 1.7 montre les fonctionnalités prises en charge par chaque profile et la relation entre ces trois ensembles. Il ressort clairement de cette figure que le profile de base est un sous-ensemble du profile étendu, mais pas du profile principal. Les détails de chaque outil

de codage seront décrits dans les prochaines sections.

Les limites de performances pour les CoDecs sont définis par un ensemble de niveaux, ces limites sont mis sur des paramètres tels que le taux de transformation de l'échantillon, la taille de l'image codée, débit et de mémoire.

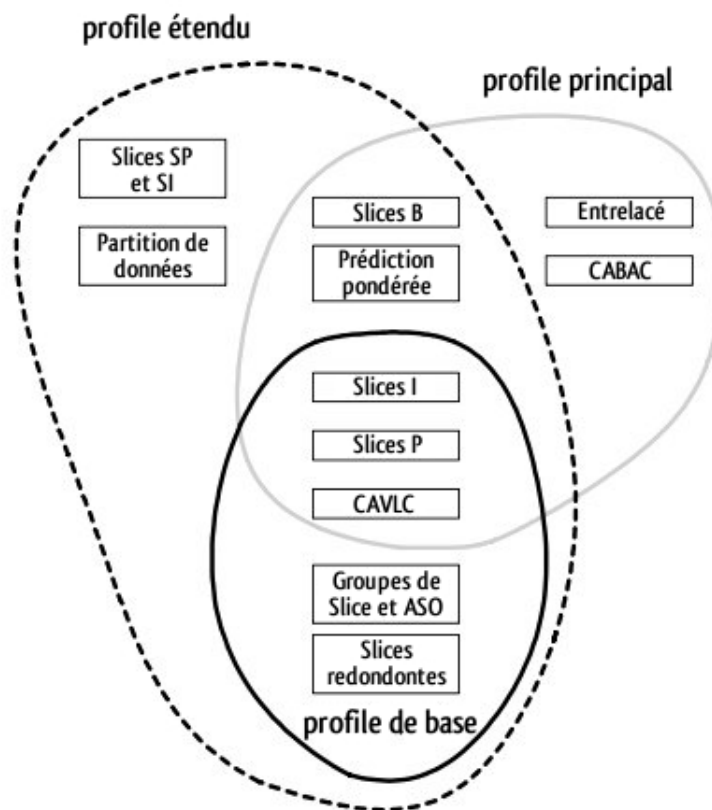


FIG. 1.7 – La relation entre les trois profils

1.7.1.4 Format des données codées et l'adaptation au réseau

H.264 fait une distinction entre une couche de codage vidéo (Video coding layer, VCL) et une couche d'abstraction réseau (Network abstraction layer, NAL). La sortie du processus d'encodage est une trame VCL ; une séquence de bits représentant efficacement le contenu vidéo, qui sont représentés par des unités NAL avant la transmission ou le stockage. La couche NAL s'intéresse plus particulièrement à une mise en forme des données vidéo adaptée au support de transmission.

Chaque NAL contient une séquence de charge utile (Raw Byte Sequence Payload RBSP) [E.G03], un ensemble d'éléments syntaxiques correspondant aux données de la vidéo codée ou aux informations d'entête. Un codage de séquence vidéo est représenté par une séquence d'unités NAL (Figure 1.8) qui peut être transmis sur un réseau ou stocké dans un fichier. Le but de spécifier séparément la VCL et NAL est de faire la distinction

entre les fonctionnalités spécifiques au codage (la VCL), qui seront décrits par la suite, et les fonctionnalités spécifiques au transport (la NAL).

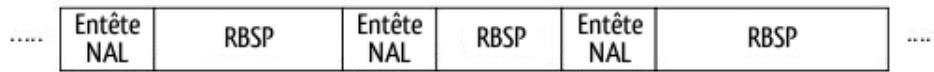


FIG. 1.8 – Séquence d’unité NAL

1.7.2 Architecture de la norme H.264

Par rapport aux anciennes normes de codage, H.264 ne définit pas explicitement un CODEC, mais plutôt définit la syntaxe d’un codage vidéo à un flux de bits ainsi que la méthode de décodage de ce flux [LHS07]. Dans la pratique, l’encodeur et le décodeur sont susceptibles d’inclure les éléments fonctionnels montrés à la Figure 1.9 . A l’exception du filtre antibloc, la plupart des éléments fonctionnels de base (de prévision, de transformé, de quantification, le codage entropique) sont présents dans les normes précédentes, mais des améliorations sont apportées au niveau de chaque bloc tout en maintenant l’architecture globale inchangée.

L’encodeur comprend deux chemins de flux de données, un chemin ”avant” (de gauche à droite) et un chemin de ”reconstruction” (de droite à gauche). Le chemin de flux de données dans le décodeur (figure 1.10) est indiqué de droite à gauche pour illustrer les similitudes entre l’encodeur et le décodeur.

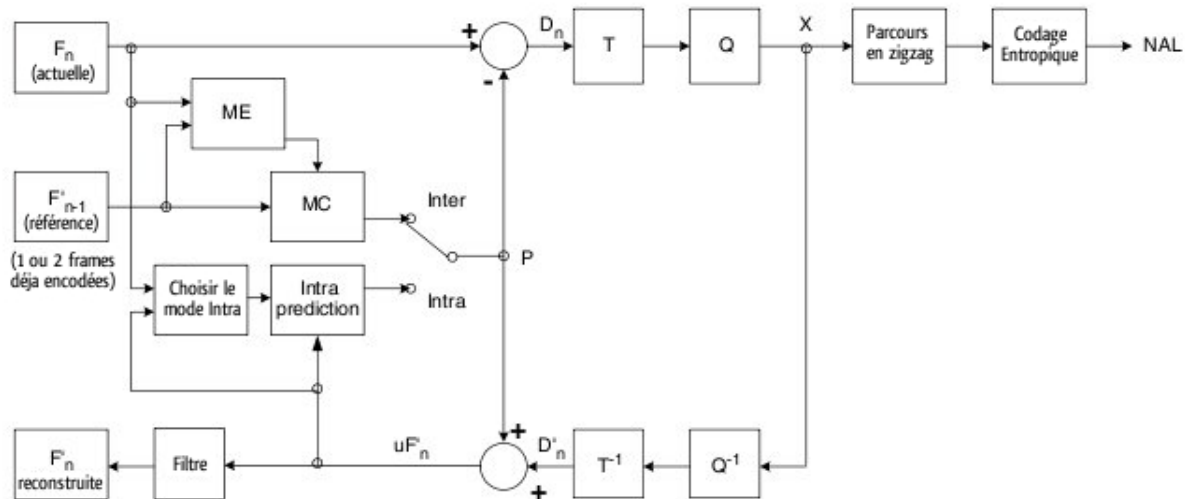


FIG. 1.9 – Shéma du codec H.264

1.7.2.1 Encodeur H.264

La figure 1.9 montre l’architecture de l’encodeur H.264 où F_n représente l’image à coder ; Cette image est composée en macrobloccs. Chaque macrobloc est codé en mode

Intra ou Inter. Dans le mode Intra, l'image prédite est formée à partir de l'image courante alors que pour le mode inter, l'image prédite est formée à partir d'une ou plusieurs images de référence qui sont précédemment codées, décodées et stockée en mémoire. En effet la prédiction est retranchée du macro-bloc courant pour produire D_n . On applique sur cette différence la transformée entière puis la quantification pour passer enfin au codage entropique après le balayage en zig-zag comme le montre la figure 1.14.

Ainsi que le codage et la transmission de chaque bloc dans un macro, l'encodeur décode (reconstruit) ce bloc pour fournir une référence à d'autres prédictions. Les coefficients X sont ajustés (Q^{-1}) et inversement transformés (T^{-1}) pour produire le bloc de différence D'_n . La prédiction du bloc est ajoutée à D'_n pour créer un bloc reconstruit uF'_n (un décodé de la version originale du bloc ; u indique qu'il est non filtrée). Un filtre est appliqué par la suite pour réduire les effets de distorsion des blocs et la reconstruction de l'image de référence est créée à partir d'une série de blocs F'_n .

1.7.2.2 Décodeur H.264

Le décodeur reçoit un flux de bits compressé en NALs. Un décodage entropique de ces données élémentaires est effectué pour produire une série de coefficients quantifiés X . Ces coefficients seront ajustés et inversement transformés pour donner D'_n (identique à D'_n indiqué dans l'encodeur). En utilisant l'entête qui contient les informations de décodage du flux binaire, le décodeur crée une prédiction du bloc identique à la prédiction originale calculée dans l'encodeur. Cette prédiction est cumulée avec D_n pour donner uF'_n qui sera filtré pour créer l'image décodée F_n . Ces étapes sont montrées par la figure 1.10

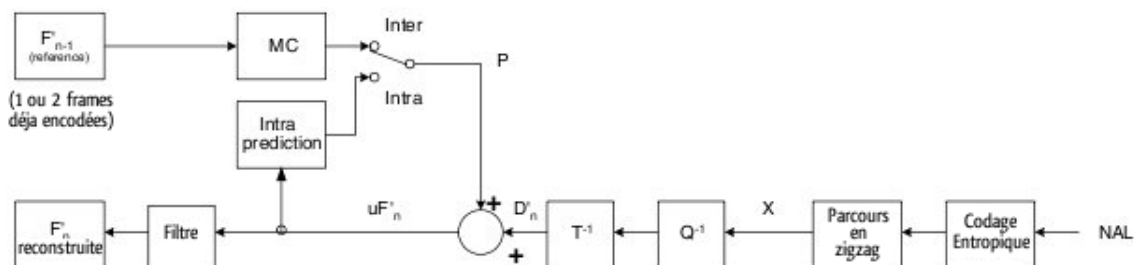


FIG. 1.10 – Le décodeur H.264

Dans ce qui suit, nous allons décrire les éléments fonctionnels de ce standard qui sont la prédiction spatiale et temporelle, la transformée et le codage entropique.

1.7.3 Les processus de la compression H.264

1.7.3.1 Division en macroblocs

Chaque image d'une séquence vidéo est partitionnée en macroblocs de dimension fixe qui couvrent une zone rectangulaire de l'image de 16×16 échantillons de luminance et de 8×8 échantillons des deux composants de chrominance. H.264 nous permet aussi le

passage aux sous-partitions. Il s'agit de blocs pouvant avoir des tailles variées comme l'illustre la figure 1.11 [LHS07].

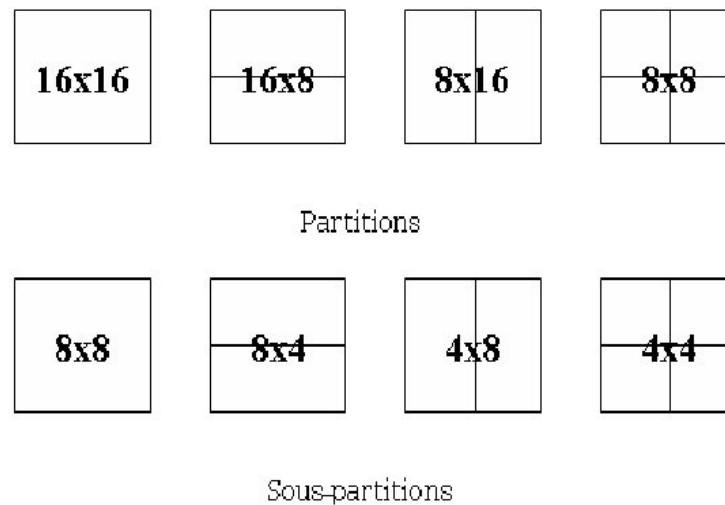


FIG. 1.11 – Les différentes partitions et sous-partitions de macroblocs dans H.264

1.7.3.2 Estimation et compensation de mouvements (Prédiction Inter)

L'estimation et la compensation de mouvements permet d'exploiter les redondances temporelles, fortement présentes dans une séquence vidéo naturelle. La compression par compensation de mouvement [SkK05] consiste pour une image, à ne coder que les erreurs résultantes de l'estimation de mouvement faite à partir d'une image de référence. Cette estimation est faite par mise en correspondance de blocs des images, ou bloc matching, afin d'obtenir les vecteurs de mouvement correspondant. Cette estimation n'est qu'une approximation, car la mise en correspondance n'est pas complète. Les erreurs résiduelles, qui consistent en la différence entre l'image prédite et l'image compensée, sont donc ajoutées aux vecteurs de mouvement. La compensation et ces erreurs sont ensuite transformées, quantifiées et codées, avant d'être transmises.

Une des différences importantes par rapport aux normes antérieures est que l'intervalle de la taille des blocs utilisés est entre 16×16 et 4×4 pixels avec une précision de quart de pixel du vecteur de mouvement. Cette précision nécessite une interpolation entre les pixels voisins de l'image initiale. Un exemple de cette interpolation est montré par le cas (c) de la figure 1.12 où les sous-pixels, point gris, sont générés par interpolation de deux pixels de l'image de référence ; points blanc.

La complexité d'une vidéo est définie par la quantité des redondances spatiales ou temporelles existantes dans celle-ci. Une vidéo représentant des objets à faible mouvement possède une grande redondance temporelle, sa complexité est considérée comme faible. Et à l'opposé, une vidéo qui contient des objets en mouvement permanent, n'a qu'une faible redondance temporelle, et de ce fait sa complexité est grande.

Un faible degré de complexité offre de meilleurs résultats pendant l'étape d'estimation de mouvement (élimination de redondance temporelle). En d'autres termes, moins

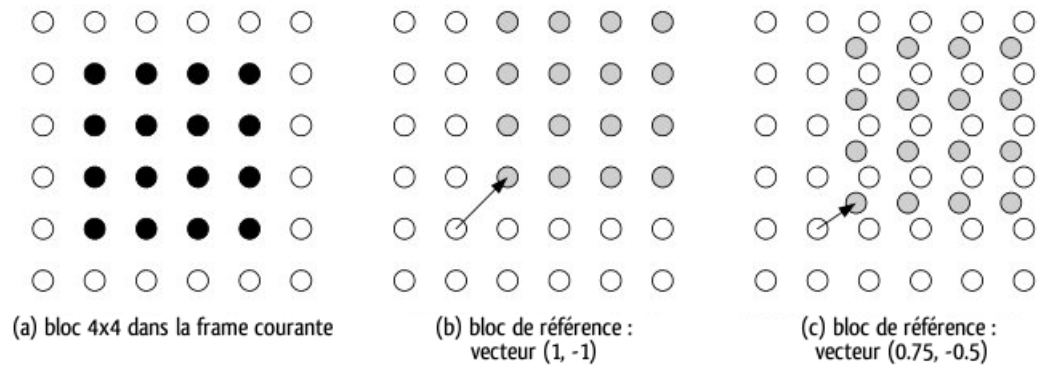


FIG. 1.12 – Exemple de prédiction entière et sous-pixel

de coefficients seront générés, et une faible quantification suffit pour avoir le taux de compression demandé; la qualité de la vidéo compressée sera donc meilleure. Le contraire est vrai, puisque lorsque la vidéo à compresser a une grande complexité, le pas de quantification nécessaire pour avoir un même taux de compression sera plus grand que celui d'une vidéo à faible complexité; l'impacte de quantification est plus significatif dans ce cas, et la qualité de la vidéo compressée générée est de qualité moindre.

Stocker ou transmettre un vecteur de mouvement pour chaque partition peut coûter un nombre important de bits, surtout si des partitions de petite taille sont choisies. Les vecteurs de mouvement des blocs voisins sont très semblables de telle sorte que chaque vecteur de mouvement peut être prédit à partir des vecteurs des partitions voisins calculés précédemment. La différence entre le vecteur courant et le vecteur prédit, est codée puis transmise.

1.7.3.3 Prédiction spatiale (Intra)

H.264 permet de disposer d'un nouveau système de prédiction Intra sophistiqué pour l'encodage des images I. Ce système peut considérablement réduire la taille binaire d'une image I tout en maintenant une qualité élevée et en permettant la prédiction successive de moindre bloc de pixels dans chaque macro-bloc d'une image.

Pour cela, le système essaye de trouver des pixels correspondants parmi les pixels encodés précédemment attendant au nouveau bloc de 4x4 pixels devant être intra-codé. La réutilisation des valeurs des pixels qui ont déjà été encodés peut réduire considérablement la taille binaire. La nouvelle prédiction Intra est un aspect essentiel de la technologie H.264 [SkK05] dont la grande efficacité a été démontrée. En revanche, si seules des images I étaient utilisées dans un flux H.264, la taille du fichier serait nettement plus réduite qu'avec un flux Motion JPEG qui utilise uniquement des images I.

1.7.3.4 Transformée fréquentielle

La transformée dans H.264 est faite de trois manières selon le type des données que représente le bloc, une transformée de Hadamard des matrices de 4×4 composantes de luminance des macroblocs prédits en mode Intra, une transformée de Hadamard des matrices de 2×2 composantes de chrominance (dans n'importe quel macrobloc) et la DCT (Discrete Cosine Transform, en français transformée en cosinus discrète) pour tous les

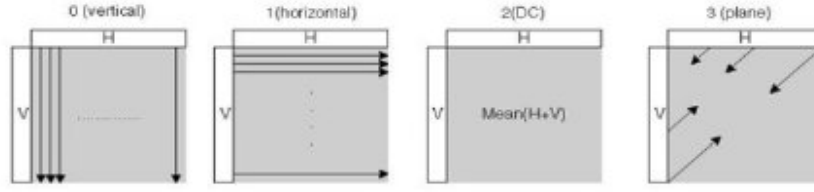


FIG. 1.13 – Les modes de prédiction Intra des bloc de 4×4

autres blocs de 4×4 .

La transformée DCT [E.G03], est une transformation numérique qui est appliquée à chaque bloc et pour chaque couleur. Cette transformée est une variante de la transformée de Fourier qui permet de décrire chaque bloc en une carte de fréquences et en amplitudes plutôt qu'en pixels et couleurs. La valeur d'une fréquence reflète l'importance et la rapidité d'un changement, tandis que la valeur d'une amplitude correspond à l'écart associé à chaque changement de couleur.

A chaque bloc de 4×4 sont ainsi associées 4×4 fréquences. La transformée DCT de la norme H.264 s'exprime mathématiquement par :

$$Y = (C_f X C_f^T) \otimes E_f \tag{1.2}$$

$$Y = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} X \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} \tag{1.3}$$

Où $C_f X C_f^T$ est le noyau de la transformé DCT, et le symbole \otimes indique une multiplication par un scalaire, plutôt qu'une la multiplication de matrices. Cela veut dire que chaque élément de la matrice $C_f X C_f^T$ est multiplié par l'élément qui a la même position dans la matrice E . E est une matrice de transformation.

La transformée inverse de la DCT est définie de manière exacte, précision entière, pour éviter les divergences dues aux arrondis. Elle s'exprime par :

$$X = C_i^T (Y \otimes E_i) C_i \tag{1.4}$$

$$X = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \left(\begin{bmatrix} X \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix} \tag{1.5}$$

Dans les deux cas :

$$a = \frac{1}{2}$$

$$b = \sqrt{\frac{2}{5}}$$

Le calcul d'une DCT est complexe. C'est l'étape qui coûte le plus de temps et de ressources dans la compression et la décompression H.264, mais c'est peut-être la plus importante car elle nous a permis de séparer les basses fréquences et les hautes fréquences

présentes dans chaque image. Dans certains cas, pour diminuer les coefficients isolés, la transformée est faite en deux passes avec une extension de la taille des blocs à 2×2 pixel.

1.7.3.5 Quantification

La DCT retourne, pour chaque bloc, une matrice de 4×4 nombres ; dans l'hypothèse que les blocs de l'image font 4×4 pixels. La quantification [SkK05] consiste à diviser cette matrice par une autre, appelée matrice de quantification, et qui contient 4×4 coefficients savamment choisis par le codeur. Contrairement à la DCT, la quantification est l'étape dans la quelle on perd des informations, donc de la qualité visuelle, mais qui fait gagner beaucoup de place.

Le but ici est d'atténuer les hautes fréquences, c'est à dire celles auxquelles l'oeil humain est très peu sensible. Ces fréquences ont des amplitudes faibles, et elles sont encore plus atténuées par la quantification. Ce processus ramène beaucoup de coefficients à 0, seules quelques informations essentielles sont gardées pour représenter le bloc. L'intérêt est qu'au moment de coder le résultat dans le fichier, la longue suite de zéros nécessitera très peu de place.

Le changement dans la taille du pas de quantification peut efficacement améliorer le taux de compression. Mais ce changement influence aussi la qualité de la vidéo générée, puisque la quantification est un processus avec perte. L'effet du changement du pas de quantification sur la qualité de la vidéo ne dépend pas exclusivement de la performance de l'algorithme utilisé par le codec. La qualité d'un flux vidéo est fortement influencée par son contenu, et spécialement la complexité de la séquence [G.W09].

Le pas de quantification défini dans la norme s'incrémente d'une valeur de 12.5% et leur dynamique est augmentée puisque les vecteurs vont de 1 à 52. Dans les normes vidéos précédentes le pas de quantification augmente par pas fixe, ce qui entraîne des zones inaccessibles pour certains quantificateurs. De plus, afin d'obtenir de meilleurs résultats visuels, la quantification de la chrominance est plus fine que celle de la luminance.

1.7.3.6 Processus de filtrage anti-bloc

La prédiction Intra appliquée sur les blocs ne tient pas compte, toujours, de la corrélation qui peut exister entre différents blocs, ce qui engendre l'apparition du phénomène des effets de blocs. Ce phénomène sera ensuite amplifié suite à l'application de la transformée DCT entière et de la quantification sur les blocs de 4×4 indépendants. Le phénomène des effets de blocs se manifeste même quand on procède à une prédiction Inter pour profiter de la redondance temporelle. En effet, une discontinuité entre les différents types de blocs adoptés par la norme peut apparaître suite à la phase de compensation du mouvement.

Le standard H.264 intègre un filtre qui épaulé l'efficacité de compression en améliorant la qualité visuelle des séquences vidéo suite au gommage de certains effets indésirables du codage tels que les effets de blocs [E.G03]. En effet, suite à la reconstruction, le processus de filtrage prend place pour lisser les bords horizontaux et verticaux au niveau des macroblocs et des blocs de 4×4 . Ce filtre est appliqué en boucle, ce qui améliore nettement

la qualité objective de l'image reconstruite. Cette amélioration affecte énormément l'efficacité de l'estimation du mouvement qui se traduit en une meilleure qualité subjective de l'image décodée.

Pour chaque bord séparant deux blocs 4×4 voisins, on assigne un paramètre B_s (boundary-Strength) [E.G03] ayant une valeur entière entre 0 et 4 indiquant la force du filtrage à appliquer sur ce bord. Cette force de filtrage est principalement affectée par le pas de quantification, les modes de codages des deux blocs adjacents et les vecteurs de mouvements appliqués sur ces derniers.

1.7.3.7 Parcours des données du Macrobloc

Après le filtrage anti-bloc, chaque bloc de 4×4 est converti en une séquence de 16 scalaires selon un ordre des niveaux de coefficient de la plus faible fréquence spatiale jusqu'à la plus forte [AKL03]. Ce type de parcours des données du bloc est appelé le balayage en zig-zag comme le montre la figure 1.14. Un deuxième mode de parcours autorise notamment le parcours du bloc en sens inverse pour fonctionner avec un codage entropique adaptatif. Le résultat est ensuite compressé selon un codage entropique à longueur variable, ce codage nous délivre un flux de bits qui représente la vidéo codée en une suite d'éléments syntaxiques.

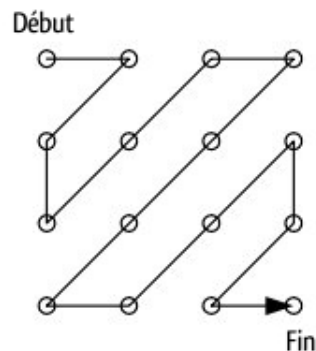


FIG. 1.14 – Le chemin de parcours des données du bloc

1.7.3.8 Codage entropique

Le standard H.264/AVC utilise un codage à longueur variable entropique afin d'encoder les coefficients issus de la transformée fréquentielle quantifiée. Ce codage variable joue un rôle important dans la compression des images ; en assignant des mots de code plus courts aux valeurs et symboles les plus fréquemment rencontrés.

Le codage entropique [E.G03] peut être réalisé de trois manières différentes dans H.264. Une première méthode utilise une table universelle de mot de code utilisant un codage dit Exp-Golomb (Exponential Golomb codes). Cette table est utilisée pour coder la plupart des éléments de synchronisation comme les entêtes. Les deux autres méthodes sont utilisées pour coder presque tous les autres éléments syntaxiques (coefficients, vecteurs

code_num	Codeword
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
...	...

TAB. 1.1 – Les mots de code Exp-Golomb

de mouvements). Il s'agit d'une part d'un codage VLC adaptatif au contexte (CAVLC - Context Adaptive Variable Length Coding) et d'autre part d'un codage arithmétique adaptatif au contexte (CABAC, Context Adaptive Binary Arithmetic coding).

Exp-Golomb Est un codage à longueur variable avec une construction régulière utilisé pour coder les différents paramètres de décodage qui seront ajoutés au flux binaire. Examinons les mots de code présentés par le tableau 1.1, on peut facilement remarquer que ces derniers sont contruits à partir d'un *codenum* comme suit :

$$[M \text{ zero}] [1] [Info] \quad (1.6)$$

Où M représente le nombre de bits nécessaire pour coder *info*. La longueur du mot codé en Exp-Golomb est $2M + 1$ bits et son code est calculé à partir du *codenum* comme suit :

$$M = \text{Int}(\log_2[\text{codenum} + 1]) + 1 \quad (1.7)$$

$$Info = \text{codenum} + 1 - 2^M \quad (1.8)$$

Le processus de décodage d'un mot de code est effectué suivant les étapes :

1. Lire les M zéro qui précèdent le premier bit à "1".
2. Lire les M bits d'information, *Info*, qui suivent le "1".
3. $code - num = 2^M + Info - 1$

CAVLC Pris en charge par les deux profils de base et étendu pour réduire significativement les informations redondantes. Ce codage est dit adaptatif dans le sens où un bloc i sera codé en fonction d'informations issues du bloc $i-1$. Cette adaptation est également utilisée à l'intérieur du bloc lors de l'encodage de ses coefficients où le code d'un coefficient dépend de la valeur du précédent. Dans le cas du premier élément, le nombre de coefficients non nuls du même bloc sont pris en compte. Le codeur CAVLC reçoit en entrée les coefficients des blocs issus de la transformation fréquentielle qui ont été quantifiés et réorganisés en zig-zag. Le codage s'effectue en cinq étapes distinctes, indépendantes l'une de l'autre.

La première étape consiste à encoder le nombre de coefficients non nuls (*TotalCoeff*) dont la valeur est égale à 1 ou -1 appelés *TrailingOnes*. La deuxième étape permet l'encodage de ces derniers où le signe est représenté par un seul bit. Le reste des coefficients non nuls sont ensuite encodés en utilisant des tables de correspondance. L'étape suivante permet de coder le nombre de zéros contenus dans le bloc en fonction du nombre de coefficients non nuls. Il est nécessaire d'indiquer dans le flux binaire final où se trouvent ces zéros par rapport aux coefficients non nuls. Cette étape est appelée *runbefore* car elle indique le nombre de zéros qui précèdent chaque coefficient non nul à l'intérieur du bloc réorganisé en zig-zag. Chacune de ces étapes délivre un élément syntaxique, la concaténation de ces derniers donne le flux binaire final.

CABAC Le codage arithmétique est une technique de compression très puissante. L'intérêt apporté à ce codage trouve ses origines dans son utilisation par les dernières normes de compression d'images et de vidéos; le profile principal de la norme H.264 prend en charge ce type de codage. Il permet, à partir de la probabilité d'apparition des symboles d'une source de créer un seul mot de code qui soit associé à une séquence de longueur variable de symboles. Ceci diffère du code de Huffman qui attribue des mots de codes de longueurs variables à chaque symbole de la source. Le code associé à une séquence est un nombre réel de l'intervalle $[0, 1[$ construit par subdivisions récursives de ce dernier. Un intervalle est subdivisé pour chaque nouveau symbole de la séquence.

1.7.4 Appréciation des erreurs de compression

La quantification est l'opération qui introduit les pertes. Ces pertes sont mesurées par erreur quadratique moyenne (EQM) qui est calculée à partir de l'image originale et de l'image reconstruite.

$$EQM = \frac{1}{N} \sum_{i=0}^N (x_i - \bar{x}_i) \quad (1.9)$$

Où x_i et \bar{x}_i sont respectivement les éléments de l'image originale et reconstruite. De cette EQM, on dérive une autre grandeur, plus significative, appelée *peak signal to noise ratio* (PSNR). Cette grandeur est un gain qui s'exprime en décibels, est s'obtient ainsi :

$$PSNR = 10 \log_{10} \frac{255^2}{EQM} \quad (1.10)$$

Cette mesure de la distorsion couplée avec le débit de la séquence résultante permet d'apprécier le compromis entre le gain de la compression et la qualité de restitution. Ce compromis est visible sur la courbe de la figure 1.15.

Le but étant, en général, d'obtenir la meilleure fidélité (ou la plus petite distorsion) compte tenu de la capacité du canal de transmission, qui détermine la contrainte de débit. Cette optimisation peut être faite à l'aide des techniques de minimisation de Lagrange fondées sur la théorie du débit-distorsion [Eff98].

1.7.5 H.264 et le temps réel

La transmission de la vidéo en temps réel nécessite une paquetisation spécifique, qui évolue avec les standards, pour garantir les exigences de ce type de transport de données.

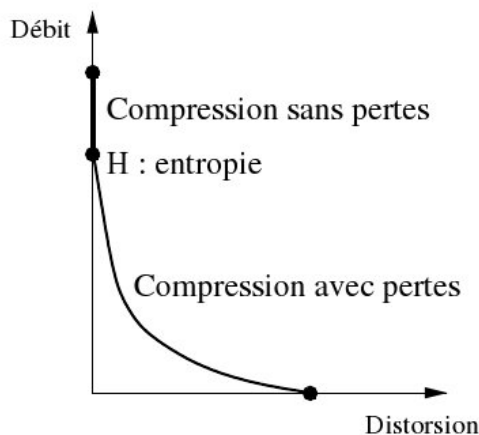


FIG. 1.15 – Allure typique d’une courbe du rapport débit / distortion

Ce type de transmission nécessite l’utilisation du protocole RTP (Real time Transport Protocol). Ce protocole est appliqué sur la charge utile de flux vidéo compressé selon la norme H.264.

En fait, le but de RTP [Ben07] est de fournir un moyen de transport standard sur IP pour les données soumises à des contraintes de temps réel comme le streaming. Le rôle principal de RTP consiste à mettre en oeuvre des numéros de séquences des paquets IP avant l’envoi de ceux-ci et de reconstituer la séquence dans le même ordre à la réception des informations, même si l’ordre des paquets est changé lors du transport de bout en bout sur le réseau IP. De manière générale, RTP permet d’identifier le type de l’information transportée, d’ajouter des marqueurs de temps et des numéros de séquence de l’information transportée et de contrôler l’arrivée à destination des paquets.

De l’autre côté, le protocole RTCP (Real time Transport Control Protocol) fonctionne conjointement avec le RTP, il est utilisé pour contrôler le transport des paquets RTP en utilisant un canal de signalisation en parallèle à celui de transport de données. Il a pour rôle d’envoyer périodiquement des paquets de contrôle à tous les participants d’une session, ce qui permet de contrôler le bon fonctionnement et de garantir la qualité de service associé (QoS).

Les paquets sont structurés d’une manière spécifique pour pouvoir offrir des services de communication en temps réel à travers les réseaux IP. Le format de charge utile est défini selon le type de charge à transporter. Dans notre cas, le flux vidéo H.264 est la charge utile à mettre dans les paquets.

Un flux H.264 est structuré, comme il est précisé dans la norme de codage vidéo H.264, en des unités NAL (Network Abstraction Layer Unit). Chaque unité NAL est formée d’une entête de taille un octet et les données de taille un nombre entier d’octets. L’entête d’unité NAL contient le type de charge utile présenté sur 5 bits, c’est-à-dire, 32 types différents sont possibles.

Puisque le concept de NAL de H.264 a été conçu pour des réseaux IP, beaucoup de contraintes se sont reflétées dans la conception de H.264. En particulier, il n’y a aucun besoin d’une entête additionnelle de charge utile en utilisant la paquetisation simple.

1.7.6 Techniques de robustesse de la norme H.264/AVC

L’une des principales causes de la dégradation de la qualité vidéo au niveau du récepteur est la présence des erreurs de transmission. Malheureusement les réseaux sans fil sont caractérisés par des taux d’erreurs qui sont très élevés par rapport aux réseaux filaires [T.S03]. Pour cela, plusieurs approches sont proposées dans la littérature pour minimiser la dégradation de la qualité vidéo perçue par l’utilisateur final dans de telles situations.

D’une manière générale, ces approches se répartissent en quatre catégories : la re-transmission des paquets perdus (e.g. ARQ) ,voir le chapitre suivant, l’insertion de la redondance (e.g. la FEC) qui fera l’objet du troisième chapitre, les algorithmes de suppression de l’effet des erreurs (error concealment) au niveau du décodeur (voir section 1.7.7) et les outils de robustesse contre les erreurs au niveau de l’encodeur [S.K06b]. Dans cette section, nous nous focalisons sur ces deux dernières catégories offertes par la norme H.264/AVC. Ces dernières peuvent être résumés comme suit :

1.7.6.1 Slice flexible et slice redondante

Comme il a été vu dans la section 1.7.1.2, la slice, qui est une suite de macroblocs (MBs) appartenant à la même image, constitue la plus petite unité qui est indépendamment décodable dans un flux vidéo. Chaque image dans la séquence vidéo est constituée d’une ou de plusieurs slices et chaque MB appartient à une seule slice. H.264 permet d’ajuster librement la taille des slices pour répondre aux exigences en terme de MTU [T.S03] (Maximal Transmission Unit) du réseau de transport sous-jacent.

Une technique dite slices redondantes [E.G03], RS : Redundant Slices, permet d’insérer dans le flux une slice primaire et un ou plusieurs slices secondaires codées avec une qualité moindre que celle du slice primaire. Si la slice primaire arrive à destination, les slices secondaires seront ignorés. Sinon, une slice redondante remplacera la slice perdue.

1.7.6.2 La technique FMO (Flexible Macrobloc Ordering)

La technique FMO [JS08][T.S03] permet de créer des slices avec des MBs qui ne sont pas forcément adjacentes en se basant sur un schéma d’allocation de MBs aux slices, appelé schéma FMO. En utilisant cette technique, chaque image est divisée en groupes de slices et chaque groupe de slices se constitue à son tour d’une ou de plusieurs slices. La Figure 1.16 montre quelques exemples d’utilisation de FMO. Si par exemple une slice de type 0 dans la Figure 1.16(b) se perd dans le réseau, chaque MB de cette slice perdue peut être estimé en se basant sur les Mbs adjacents appartenant à la slice de type 1.

1.7.6.3 Partitionnement de données

Le mode DP (Data Partitionning) est un mécanisme efficace pour augmenter la robustesse. Il permet de séparer les segments d’un flux binaire en fonction de leur impor-

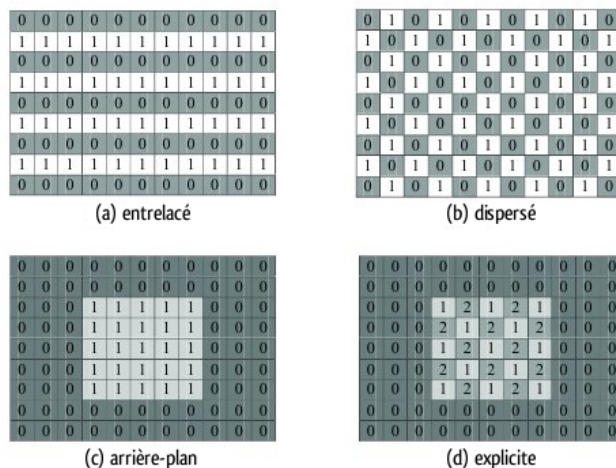


FIG. 1.16 – Exemples d'utilisation de FMO

tance. Des mécanismes externes pourront permettre de mieux protéger ou de privilégier l'acheminement des données les plus importantes. Les entêtes, les vecteurs de mouvement et les coefficients DCT de MB successifs ne sont plus entrelacés dans le flux binaire en suivant l'ordre de parcours de l'image mais sont groupés dans des portions de flux différentes.

H.264 permet de coder chaque slice en trois parties [T.S04] : la partie A qui contient les entêtes des slices, la partie B qui contient les coefficients des MBs I (prédiction spatiale) et la partie C qui contient les coefficients des MBs P et B (prédiction temporelle). De cette façon, même si les parties B et C sont perdues, les informations de la partie A demeurent utilisables par les algorithmes de suppression de l'effet des erreurs ou par une sur-protection de la partie A.

1.7.6.4 Ensemble de paramètres de séquence

Afin de décoder correctement une vidéo à haut débit, le décodeur se synchronise avec l'encodeur en terme de paquet et de syntaxe [S.K06b], par exemple un décodeur doit connaître le format de l'image, le codage entropique utilisé, la résolution du vecteur de mouvement etc. Ces paramètres, utilisés pour décoder un groupe d'images ou une série de tranches, gardent généralement les mêmes valeurs dans une session de streaming.

Grâce à cette propriété, le codeur et le décodeur utilisent la même table de paramètres stockée dans la mémoire locale où il garde les différentes combinaisons possibles de ses derniers ; cette table est appelée *parameter_sets*. Pour signaler ces paramètres de configuration au décodeur, l'index de la combinaison est transmis en lieu et place des paramètres eux-mêmes.

Dans des cas particuliers, pour une surprotection, l'index est transmis en dehors de la bande passante. cette méthode offre plus de sécurité contre l'intrusion ; comme les valeurs actuelles ne sont pas transmises. Une autre copie de ce dernier peut être transmise avec un petit overhead pour une meilleur résistance au bruit.

1.7.6.5 Sélection de l'image de référence

Le principe du mode RPS (Reference Picture Selection) est de permettre la prédiction non seulement à partir de la seule image précédente, mais aussi à partir d'autres images plus lointaines. Ce mode pourra être appliqué à une image entière, à une slice ou à un MB.

L'application à une image ou à une slice a été proposée pour seul objectif d'accroître la robustesse. L'application aux Mbs permet aussi d'accroître le taux de compression. La signalisation dans le flux binaire sera différente suivant l'entité sur laquelle ce mode est appliqué. Dans les deux premiers cas, l'information indiquant l'image de référence à utiliser devra être transmise une fois par image ou par slice. Dans le dernier cas, chaque MB devra contenir cette information ce qui induira des vecteurs de mouvement à trois dimensions, la troisième dimension étant l'information temporelle sur l'image de référence.

Ce mode permet d'éviter que toutes les images soient interdépendantes. Une méthode simple pour augmenter la robustesse consiste à définir dans un même flux vidéo plusieurs groupes d'images décodables indépendamment. Si l'un des groupes subit des erreurs, les autres groupes ne seront pas affectés.

1.7.7 Détection et dissimulation d'erreur dans un flux H.264

A cause de l'utilisation des codes à longueur variable dans le codeur source, n'importe quel dommage à un simple bit peut causer la désynchronisation, conduisant à l'impossibilité de décoder le bit suivant. Cette impossibilité peut être mise à profit pour détecter les erreurs de transmission. Dans la plupart des cas, le VLC employé n'est pas à code complet, c'est à dire, que tous les mots de codes possibles ne sont pas forcément des mots de code autorisés. Par conséquent une fois qu'un décodeur vidéo détecte un mot de code qui n'est pas dans sa table de décodage, une erreur de transmission est déclarée. En outre, la syntaxe incluse dans le flux binaire peut également être employée pour la détection des erreurs.

Dans le cas de l'utilisation du protocole RTP, le flux vidéo codé doit être mis sous forme de paquets, dont chacun contient un champ d'entête et de charge utile. L'entête contient un champ de numéro de séquence qui est consécutif pour les paquets séquentiellement transmis, c'est à dire après chaque envoi d'un paquet une incrémentation du numéro de séquence est effectuée. Au niveau du décodeur, le numéro de séquence peut être employé pour la détection de perte de paquet.

La dissimulation d'erreurs [S.K06b], error concealment EC, offerte par la norme H.264 consiste à limiter au maximum l'effet des erreurs de transmission en exploitant les redondances résiduelles présentes dans le signal vidéo. Il s'agit dans la plupart des cas d'effectuer pendant le décodage des interpolations spatiales ou temporelles utilisant les MB reçus pour retrouver les macroblocs perdus.

Comme vu dans le paragraphe précédant, il existe deux grandes familles de méthodes de dissimulation d'erreurs : la dissimulation d'erreurs temporelle et la dissimulation d'erreurs spatiale. La plupart des méthodes temporelles estiment le vecteur de mouvement associé au MB perdu à partir des vecteurs de mouvement des MB adjacents. On affecte ensuite au MB manquant la valeur des MB correspondant compensées en mouvement.

Une manière simple d'appliquer ce principe consiste à remplacer un MB perdu dans l'image courante par le MB situé à la même position spatiale dans l'image précédente.

Cela revient à considérer le MB perdu comme non codé. Un mécanisme plus élaboré a été proposé dans le Modèle H.264. Celui-ci consiste à affecter à un MB perdu le vecteur de mouvement du MB situé au dessus du MB perdu. Si ce dernier est indisponible, alors un vecteur de mouvement nul est affecté au MB considéré. Le MB indiqué par le vecteur de mouvement est alors recopié à la place du MB manquant. D'autres auteurs proposent de former un vecteur moyen, un vecteur médian ou un vecteur choisi par maximum à posteriori.

1.8 Conclusion

La norme H.264 représente un énorme pas en avant dans le domaine des technologies de compression vidéo. Elle garantit une meilleure efficacité de compression en raison de la plus grande précision des fonctions prédictives et d'une meilleure tolérance des erreurs. Elle offre de nouvelles possibilités pour la création d'encodeurs vidéo offrant des flux vidéo de plus haute qualité, des fréquences d'image plus importantes et des résolutions plus élevées, à débit égal (par rapport aux normes précédentes) ou, à l'inverse, la même qualité vidéo à des débits inférieurs. Dans la suite de notre travail, nous allons nous intéresser aux différentes méthodes de transmission fiable de la vidéo H.264 en temps réel sur les réseaux sans fils ainsi que les différents protocoles avec qualité de service.

Chapitre 2

Généralités sur la vidéo sans fil

2.1 Introduction

Initialement les réseaux locaux sans fil ont été conçus pour mettre en place des transmissions dans des endroits difficiles à câbler et assurer la communication des données pour des applications mobiles. Ces réseaux doivent actuellement supporter des applications plus complexes ayant des besoins contraignants en terme de délai de bout-en-bout, de bande passante, etc. Pour apporter cette Qualité de Service (QoS), les recherches menées dans ce domaine ont permis la mise en place de nombreux protocoles pour garantir la qualité de service. Le but d'un protocole dépend de la couche dans laquelle il est mis en place.

Nous présentons dans ce chapitre un état de l'art sur les mécanismes de QoS qui ont été définis durant ces dernières années au niveau des couches du modèle TCP/IP. Au niveau des couches MAC, réseau et transport, nous nous focalisons particulièrement sur les réseaux sans fil 802.11 et au niveau applicatif, nous nous limitons aux applications multimédias.

2.2 Canal de transmission

Le modèle traditionnel d'un système de communication vidéo est illustré sur la Figure 2.1. Considérons dans cette section la couche physique. Le canal à ce niveau est souvent modélisé par la distribution de probabilité de sa sortie Y pour une entrée X donnée. On distingue : les canaux stationnaires dont les paramètres sont fixes au cours du temps (fibres optiques, câbles métalliques), et les canaux non stationnaires dont les paramètres évoluent au cours du temps (les communications avec les mobiles, communications sans fils).

Dans ce qui suit, nous allons présenter la différence entre ces deux groupes ainsi que leur modèles respectifs, à savoir les canaux gaussiens et de Rayleigh.

2.2.1 Canaux stationnaires

Parmi les canaux stationnaires, le plus utilisé, celui sur lequel l'évaluation des performances des systèmes de communications est aussi la plus simple, est le canal AGB (bruit Additif Gaussien Blanc) ou AWGN (Additive White Gaussian Noise). Ce canal de transmission est rencontré dans les transmissions par faisceaux hertziens à faible débit

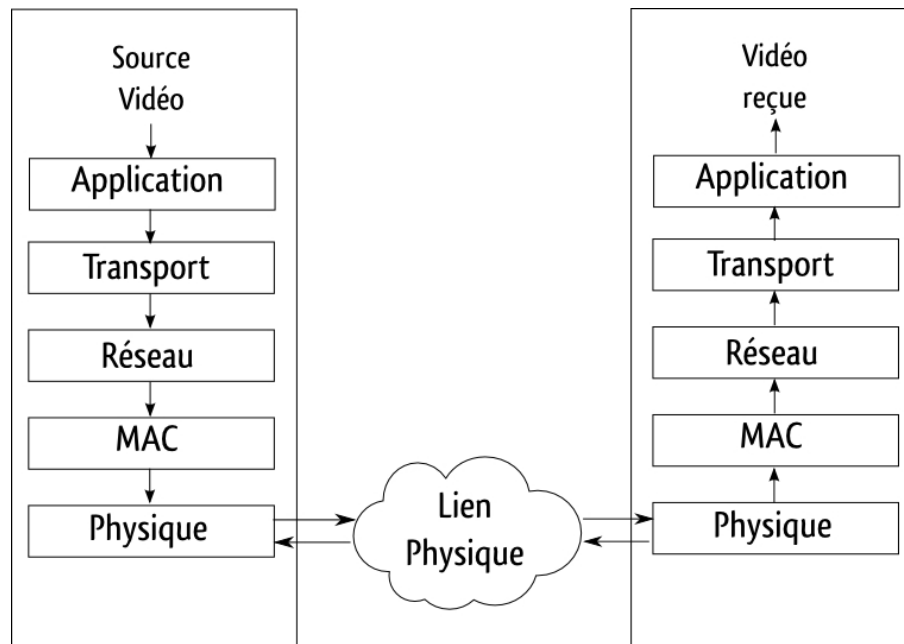


FIG. 2.1 – Le modèle d'un système de transmission

ou dans les liaisons entre des satellites ou des sondes spatiales et des stations terriennes, c'est l'ensemble des transmissions radios électriques en espace libre.

2.2.1.1 Canal gaussien

Considérons d'abord que le canal de transmission est constitué de l'ensemble modulateur, bruit du canal et démodulateur, et que le bruit ajouté par le canal est un bruit blanc gaussien additif. Soient X et Y les alphabets à l'entrée et à la sortie de ce canal, constitués de n symboles $X = \{x_1, x_2, \dots, x_n\}$ et $Y = \{y_1, y_2, \dots, y_n\}$. La fonction de distribution de probabilité $P(X/Y)$ de la sortie Y du canal pour une entrée donnée X est alors donnée par la fonction de Gauss :

$$P(X/Y) = \frac{1}{\sigma_b \sqrt{2\pi}} e^{-\frac{\|X-Y\|^2}{2\sigma_b^2}} \quad (2.1)$$

Où σ_b est la variance du bruit.

Taux d'erreur binaire

Le taux d'erreur binaire (TEB) est une manière d'exprimer la probabilité d'erreurs se produisant en sortie du récepteur. On peut obtenir le taux d'erreur binaire en calculant le nombre d'erreurs observées dans le récepteur par rapport au nombre de bits reçus.

Dans le cas où la transmission est faite sur un canal gaussien et si on considère que les filtres d'émission et de réception sont normés, le signal à l'entrée de l'échantillonneur est défini par :

$$y_n = d_n + b_n \quad (2.2)$$

Où le bruit $\{b_n\}$ est complexe blanc gaussien de variance N_0 et E_b est l'énergie par élément binaire du signal réel passe bande transmis. La distance euclidienne d entre les signaux réels est définie par :

$$d^2 = \int_{-\infty}^{+\infty} \left(\sqrt{E_b}g(t) - \left(\sqrt{E_b}g(t) \right)^2 \right) g^2(t) dt = \left(\sqrt{E_b} + \sqrt{E_b} \right)^2 = 4E_b \quad (2.3)$$

La fonction de mise en forme spectrale $g(t)$ est une fonction en racine de Nyquist, c'est à dire que sa transformée de Fourier $G(f)$ satisfait la condition.

$$\sum_{n=-\infty}^{+\infty} \left| G\left(f - \frac{n}{T}\right) \right|^2 = \text{constante} \quad (2.4)$$

Soit donc :

$$d = 2\sqrt{E_b} \quad (2.5)$$

Après transmission dans un canal AGB, les performances d'une modulation binaire sont définies à partir de la distance entre les signaux à discriminer par :

$$TEB = Q\left(\sqrt{\frac{d^2}{2N_0}}\right) \quad (2.6)$$

Où $Q(x)$ est la fonction de Marcum :

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du \quad (2.7)$$

Pour une modulation binaire à valeurs antipodales, on utilise les relations 2.6 et 2.7 pour obtenir la relation de référence de la probabilité d'erreur par bit est :

$$TEB = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (2.8)$$

Elle ne dépend que de l'énergie par bit et de la densité spectrale $N_0/2$ du bruit additif. La variation du taux d'erreur binaire, pour une modulation binaire antipodale, en fonction du rapport B_b/N_0 est donnée par la figure 2.2

2.2.2 Canaux non stationnaires

Les canaux sélectifs non stationnaires, comportent un ensemble discret ou continu de canaux de Rayleigh [RK01] associés à des temps de propagation différents où le gain complexe du canal est un processus aléatoire gaussien complexe, par exemple les communications radios mobiles à faible débit. Ils peuvent être représentés par un filtre linéaire non stationnaire.

La modélisation du canal est une phase importante est très étudiée dans le domaine de la transmission sans fil, le modèle du canal dépend essentiellement de sa capacité. Dans

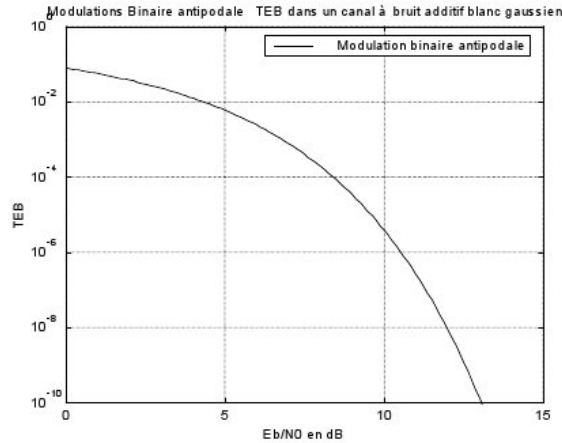


FIG. 2.2 – Performances des modulations binaires antipodales.

Canaux de transmission	Non sélectifs	Sélectifs
Stationnaires	<p>Bruit blanc gaussien D.S.P. $\frac{N_0}{2}$</p> <p>Signal \oplus Signal+bruit</p>	<p>Bruit blanc gaussien D.S.P. $\frac{N_0}{2}$</p> <p>Signal \rightarrow [Filtre linéaire $H(f)$] \oplus Signal filtré +bruit</p>
Non stationnaires	<p>Bruit blanc gaussien Processus gaussien complexe D.S.P. $\frac{N_0}{2}$</p> <p>Signal \otimes Signal+bruit</p>	<p>Bruit blanc gaussien D.S.P. $\frac{N_0}{2}$</p> <p>Signal \rightarrow [Filtre linéaire non stationnaire $H(f;t)$] \oplus Signal filtré +bruit</p>

FIG. 2.3 – Classification de canaux de transmission

ce chapitre, nous allons présenter deux modèles, mathématique et stochastique, du canal sans fil.

Généralement, l'évanouissement dans un canal sans fil est modélisé par un processus aléatoire de Rayleigh. L'évanouissement de Rayleigh peut être générés au moyen de deux processus gaussiens comme suit :

$$f_{Rayleigh}(i) = \sqrt{g_1(i)^2 + g_2(i)^2} \tag{2.9}$$

Où $g_1(i)$ et $g_2(i)$ sont deux variables indépendantes de Gauss de variance σ et i représente le temps.

Le schéma de principe d'un canal de Rayleigh, à temps discret, est représenté sur la figure 2.4. Le gain instantané complexe représente l'effet de la transmission par trajets multiples. C'est effectivement un inconvénient sur les performances, qui sont très mauvaises. Mais c'est aussi un avantage très important puisque la présence de réflexions et de diffractions très nombreuses permet de réaliser une transmission radio électrique sans que le récepteur soit en vue directe de l'émetteur. En pratique on utilise un canal de Rayleigh dans une transmission radio électrique sans vue directe entre l'émetteur et le récepteur.

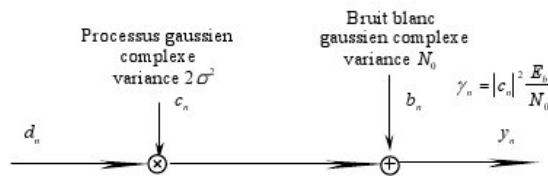


FIG. 2.4 – Le canal de rayleigh

Phénomène de multitrajets

Le canal radio électrique est le siège de la propagation, par multitrajets, des ondes électromagnétiques (voir figure 2.5). Les multitrajets se manifestent lorsque le signal émis par un capteur arrive au récepteur après réflexions et diffractions sur un obstacle : murs, surfaces vitrées ou métalliques en zone urbaine, troncs d'arbres, branche et feuillage en forêt. Ces interactions électromagnétiques modifient des paramètres intrinsèques de l'onde comme sa polarisation et sa phase.

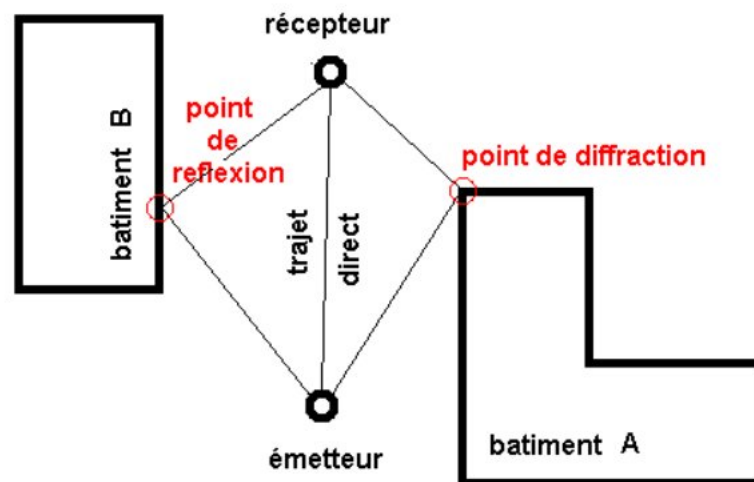


FIG. 2.5 – Illustration de propagation multitrajets

Le canal de propagation est vu comme un système qui transforme le signal d'entrée en un autre signal de sortie. Il peut être modélisé par un filtre linéaire dont les caractéristiques n'évoluent pas dans le temps et dans l'espace. Cette affirmation est due à la position fixe d'émetteur et du récepteur et d'inexistence du mouvement dans notre environnement. Donc, le canal est invariant dans le temps. Le phénomène de multitrajets peut être modélisé par une réponse impulsionnelle complexe qui comprend l'atténuation, le délai des ondes et la phase du signal sur toutes les versions du signal reçu au niveau récepteur (chaque trajet a une autre version du signal).

2.2.2.1 Probabilité d'erreur sur un canal de Rayleigh

On suppose que l'on connaît la valeur c_n du gain du canal pour chaque symbole transmis d_n . Pour une valeur c_n connue, la probabilité d'erreur sur le symbole d_n est donnée par celle obtenue par la même modulation sur un canal AWGN mais avec une énergie par bit reçue .

$$E_{b,n} = |c_n|^2 E_b = \rho_n^2 E_b \quad (2.10)$$

Soit la probabilité d'erreur conditionnellement à gain connu c_n :

$$P_e(c_n) = P_e(\rho_n) = Q\left(\sqrt{2\frac{\rho_n^2 E_b}{N_0}}\right) \quad (2.11)$$

Le rapport signal sur bruit instantané est défini par $\gamma_n = \frac{\rho_n^2 E_b}{N_0}$. D'où l'expression de la probabilité d'erreur pour un rapport signal sur bruit instantané γ_n :

$$P_e(\gamma_n) = Q\left(\sqrt{2\gamma_n}\right) \quad (2.12)$$

La probabilité d'erreur moyenne, peut être évaluée par l'intégrale suivante sur la loi du rapport E_b/N_0 instantané :

$$P_e = \int_0^\infty P_e(\gamma_n) p(\gamma_n) d\gamma_n = \int_0^\infty Q\left(\sqrt{2\gamma_n}\right) p(\gamma_n) d\gamma_n \quad (2.13)$$

Pour effectuer cette intégrale, il faut déterminer la loi du module du gain instantané et ensuite celle du rapport signal sur bruit instantané.

Loi du module du gain instantané

Le module ρ_n du gain instantané suit une loi de Rayleigh :

$$P(\rho_n) = \frac{\rho_n}{\sigma^2} e^{-\frac{\rho_n^2}{2\sigma^2}} \quad (2.14)$$

avec

$$E\{P_n^2\} = 2\sigma^2$$

Loi du rapport signal sur bruit instantané

Le rapport signal sur bruit instantané est défini par :

$$\gamma_n = \frac{\rho_n^2 E_b}{N_0} \quad (2.15)$$

Le rapport signal à bruit moyen est défini par l'espérance mathématique du rapport signal sur bruit instantané :

$$\gamma = \frac{E\{\rho_n^2\} E_b}{N_0} = \frac{2\sigma^2 E_b}{N_0} \quad (2.16)$$

La loi du rapport signal sur bruit est définie à partir de celle du module du gain instantané γ_n par le changement de variable $\rho_n = \sqrt{\frac{\gamma_n N_0}{E_b}}$.

Le rapport signal sur bruit instantané suit une loi exponentielle de paramètre γ :

$$P(\gamma_n) = \frac{1}{\gamma} e^{-\frac{\gamma_n}{\gamma}} \quad (2.17)$$

Cette relation est utilisée pour calculer les performances des modulations numériques transmises sur un canal de Rayleigh. Sur la figure 2.6/(b), on compare les performances d'une modulation numérique binaire antipodale transmise sur un canal AGB et sur un canal de Rayleigh. Pour une probabilité d'erreur de 10^{-3} , il faut un rapport E_b/N_0 de 24 dB sur un canal de Rayleigh alors que 6,5 dB suffisent sur un canal AWGN.

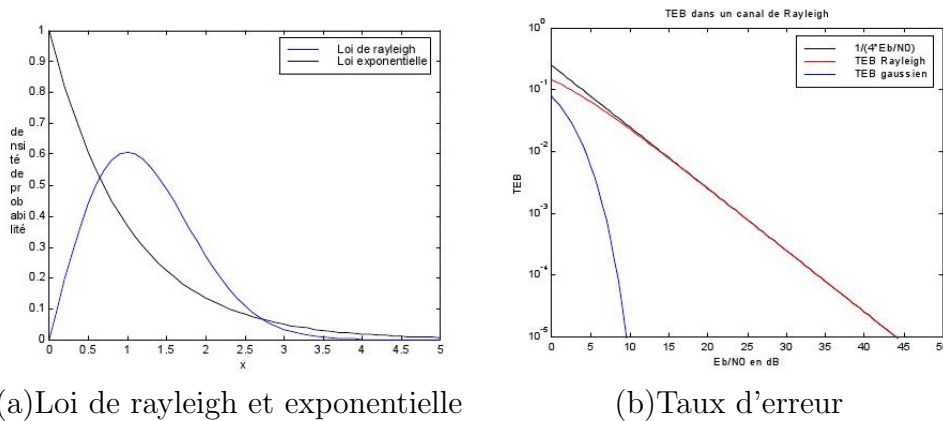


FIG. 2.6 – Loi de rayleigh et son taux d'erreur

2.2.2.2 Modélisation stochastique du canal radio

La deuxième méthode utilisée pour modéliser le canal sans fil, est celle basée sur l'utilisation des chaînes de Markov à deux états et qui représente une simplification du modèle de Gilbert-Elliot [JS08], ce dernier a été utilisé pour modéliser la perte des paquets dans le canal sans fil. Ce modèle a deux états, le bon et la mauvais état, notés par (G) et (B) successivement comme le montre la figure 2.7. Si le canal est dans un état dit "Bon", on peut dire que le paquet peut être transmis sans erreurs. Par contre s'il est dans un mauvais état, le paquet arrive au récepteur avec quelques erreurs.

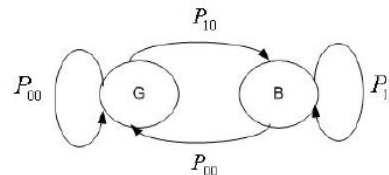


FIG. 2.7 – Modèle de Markov à deux états

P_{00} , P_{01} , P_{10} , P_{11} sont les probabilités de transition d'un état à un autre. La probabilité moyenne de perte est :

$$\overline{P}_e = \frac{P_{10}}{P_{10} + P_{00}} \quad (2.18)$$

Les probabilités de transition entre les états sont données par la matrice P .

$$P = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix} = \begin{bmatrix} 1 - P_{01} & P_{01} \\ P_{10} & 1 - P_{10} \end{bmatrix} \quad (2.19)$$

Les probabilités de transition P_{01} et P_{10} peuvent être dérivées à partir des statistiques du canal, ses valeurs sont données par les deux équations 2.18 et 2.19. Supposons que le canal suit un processus de Markov qui a une distribution géométrique de moyenne $1/P_{10}$.

$$P_{10} = \frac{1}{M} \quad (2.20)$$

$$P_{01} = \frac{P_{10} \cdot PER}{1 - PER} \quad (2.21)$$

Où M représente la longueur du burst, corrélation des erreurs, et PER est le taux de perte de paquet calculé en utilisant un modèle d'erreur généré par un simulateur de canal sans fil.

2.2.3 Capacité du canal

La capacité du canal indique le débit maximal, en bits d'information par seconde, qui peut être transmis sans erreur depuis la source jusqu'à la destination. Cela reste vrai même lorsque le canal de transmission est bruyant, à condition que les exigences de bande passante est de rapport signal sur bruit soit satisfaites. Cela ne signifie pas que le taux d'erreur est nul. Il existera toujours une probabilité finie d'erreur et, par conséquent, le taux d'erreur sera toujours supérieur à zéro.

En 1948, Claude Shannon a publié sa théorie de communication, qui a été renommée *Théorie mathématique de la communication* dans sa première réédition. Sa forme la plus simple, pour un canal gaussien, est donnée par l'équation 2.22, cette loi lie la capacité du canal C au rapport signal sur bruit E_b/N_0 dans la bande passante utilisable du canal de communication W .

$$C = W \log_2 (1 + E_b/N_0) \quad (2.22)$$

Lorsque un système de communication numérique fonctionne à un débit qui est supérieur à la capacité du canal, la probabilité d'erreur sera élevée. Ainsi, il est très important d'avoir un rapport signal sur bruit suffisant pour l'application dans des conditions de fonctionnement en environnement réel afin de garantir un fonctionnement fiable et sans erreur. Il est également très important d'avoir une bande passante suffisante pour garantir le fonctionnement de l'application réseau à utiliser dans les limites de capacité du canal. Dans le quatrième chapitre de ce document, nous allons proposer une bonne utilisation de cette capacité en métrant la sémantique des données transmises comme contrainte dans la phase d'allocation de débit.

2.3 Réseaux IP et communication vidéo

Les réseaux IP peuvent être classés en deux catégories : réseaux IP non contrôlés tel que l'Internet, et réseaux IP contrôlés tels que les réseaux de large étendue ; réseaux privés. Une troisième catégorie émerge, les réseaux sans fil, qui reposent sur les réseaux mobiles

de troisième génération, UMTS¹. Chacun des trois types de réseau a des caractéristiques un peu différentes en termes de taille maximum d'unité de transfert, probabilité d'erreurs en paquets, et nécessité d'obéir au paradigme du trafic du protocole de contrôle de transmission (Transmission Control Protocol : TCP) .

La taille de MTU² représente la taille maximale d'un paquet qui peut être transmis sans être fragmenté ou recombinaison sur les deux couches transport et réseau. Il est généralement recommandé de choisir des tailles de tranches codées proches mais ne dépassant pas la taille de MTU. L'optimisation du rapport entre la taille de la charge utile et l'entête dans un paquet ainsi que la réduction au maximum de la probabilité de perte de la tranche codée sont deux raisons expliquant le choix de la taille de MTU. En effet, les protocoles des couches transport ou réseaux rejettent tous les fragments appartenant à une tranche dans le cas de perte d'un de ces derniers.

La taille bout-en-bout de la MTU à travers un chemin de transmission entre deux noeuds d'un réseau IP est très difficile à identifier, et peut changer dynamiquement pendant une connexion. Cependant, la plupart des recherches considèrent des tailles de MTU d'environ 1500 octets pour des liens d'un réseau IP par câble. Dans un environnement sans fil, la taille de MTU est en général plus petite. Une taille d'environ 100 octets est utilisée dans la plupart des recherches du groupe JVT (Joint Video Team).

2.3.1 Contrôle de trafic TCP

Depuis l'apparition de l'Internet à la fin des années 1980, le protocole de transport TCP, qui est employé pour porter la plupart de contenu d'Internet tel que les courriers électroniques et le trafic web, obéit au paradigme du trafic de TCP [EK04]. De manière plus synthétique, le paradigme du trafic de TCP exige qu'un expéditeur ramène son débit binaire d'envoi à la moitié (en raison d'un ajustement de la taille d'amortisseur de TCP) dès qu'il observe un taux de perte de paquet au-dessus d'un certain seuil.

Une fois que le taux de perte de paquet chute en-dessous du seuil, l'expéditeur peut lentement augmenter son débit binaire d'envoi, jusqu'à ce que le taux de perte de paquet soit de nouveau trop haut et le processus entier se relance. Ce mécanisme réglementaire est simple, mais il est efficace; il empêche la surcharge des routeurs et assure une équité raisonnable entre les expéditeurs.

Dans les réseaux sans fil, où les pertes de paquet dues aux imperfections des protocoles de la couche liaison sont plus communes, cet algorithme ne donne pas les résultats optimaux. Il est néanmoins une des bases des réseaux IP en général et d'Internet en particulier, et devrait être utilisé par tous les éléments du réseau pour empêcher la congestion. Un dispositif d'envoi vidéo basé sur l'IP doit supporter le contrôle de congestion avec ou sans les arrangements de transmission pour assurer une expérience de haute qualité à l'utilisateur. Il est important de signaler le rapport entre le contrôle de congestion et le codage avec résilience d'erreurs; le but des mécanismes de contrôle de la congestion est de réduire la charge de réseau quand on observe des taux d'erreur plus élevés.

¹Universal Mobile Telecommunication System

²Maximum Transfer Unit

Le codage vidéo avec résilience d'erreurs tend à ajouter la redondance au flux binaire pour faire face aux taux d'erreur élevés et augmenter le débit binaire et la charge du réseau en gardant le même niveau de qualité. En conséquence, les deux outils principaux de la communauté qui sont utilisés pour combattre les erreurs sont deux approches contradictoires. Jusqu'ici, la seule solution largement acceptée pour ce problème a été la réduction de la qualité de la vidéo du codeur de source en réduisant le nombre d'images d'envoi, taille d'image, qualité d'image et, dans le plus mauvais cas, l'abandon de la transmission vidéo dans l'ensemble.

2.3.2 La chaîne de transmission vidéo

Le schéma classique d'une chaîne de transmission vidéo est représenté sur la figure 2.8. En télécommunication, on utilise le modèle simplifié.

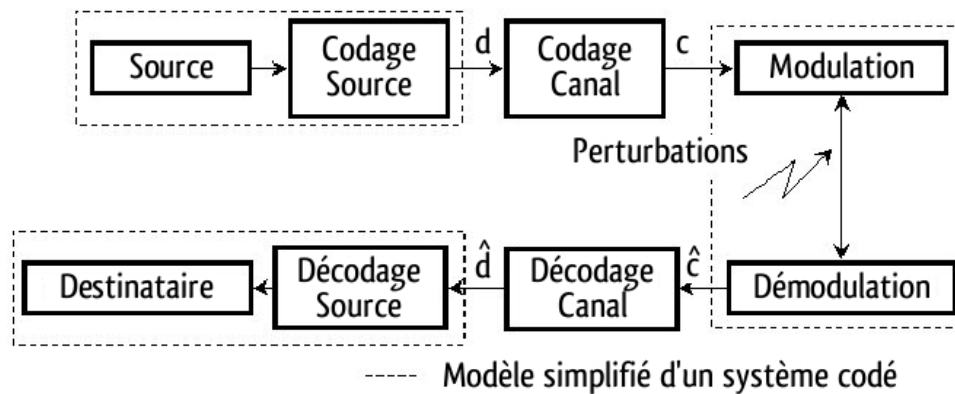


FIG. 2.8 – Diagramme d'une chaîne de transmission classique

2.3.2.1 Codage de la source

De nos jours, il est possible de procéder à une communication vidéo depuis n'importe quel point (ou presque) et à tout moment. Les algorithmes de compression modernes de la vidéo ont permis la réalisation d'applications jugées impensables il y a seulement quelques années, exemple, la radiodiffusion de la vidéo numérique, le streaming vidéo sur Internet ou la visiophonie mobile. Les normes qui ont permis ces progrès (H.264 et MPEG4, introduites dans le chapitre précédent) réalisent la partie compression du processus de codage de la source.

Le codage source consiste d'une part à convertir le message continu en une séquence numérique (utilisation d'un convertisseur analogique numérique : CAN) et à transformer le message de la source en une séquence d'information $D(x)$ de façon à :

- minimiser la taille du message en éliminant les redondances naturelles de l'information source (algorithme de compression).
- pouvoir retrouver le message original à partir de la séquence de substitution $D(x)$ (algorithme réversible).

Le but de cette opération est d'optimiser les ressources nécessaires à la transmission (temps, puissance, bande passante, etc.). Il est à noter que les limites théoriques du codage source sont fixées par le premier théorème de Shannon.

2.3.2.2 Codage de canal

Le codage canal a pour rôle de protéger l'information émise contre les perturbations du canal de transmission susceptible de modifier son contenu. Il s'agit donc de rajouter de la redondance de manière à détecter et éventuellement corriger les erreurs lors de la réception si la stratégie adoptée le permet. Les conditions d'un codage correct sont déterminées par le second théorème de Shannon. L'information $D(x)$ issue du codage source est transformée en séquence codée $C(x)$. Plusieurs techniques de codage de canal existent dans la littérature et seront introduites dans le chapitre suivant.

2.3.2.3 Modulation

Elle associe à chaque symbole émis un signal de durée τ_m . Le type du signal dépend des propriétés physiques du support de transmission (canal) choisi. Ainsi, l'information peut être véhiculée par une tension ou un courant si le support est filaire (cuivre), une onde lumineuse si le support est optique (fibre optique) ou une onde électromagnétique ou radioélectrique dans le cas d'un support aérien (air ou vide). La modulation permet de remédier à tous les inconvénients liés à une transmission en bande de base. Parmi les inconvénients d'une transmission en bande de base :

- les signaux basse fréquence sont les plus atténués sur la ligne ;
- pas de propagation pour les signaux de fréquence en dehors de la bande passante du canal ;
- pertes et affaiblissement proportionnels à la longueur et aux types du support de transmission ;
- impossibilité de différencier plusieurs communications sur un même support ;
- régénération périodique du signal sur une longue distance.

La modulation utilise une onde porteuse qui sert à transposer les données par modification d'une ou de plusieurs caractéristiques de cette onde, amplitude (AM), phase (BPSK, QPSK, etc.)³, fréquence (FM), amplitude et phase (QAM)⁴, etc.

2.3.2.4 Canal de transmission

Il représente la liaison entre l'émetteur et le récepteur et peut être de différentes natures selon le type de grandeur qu'il permet de véhiculer. Le canal de transmission est caractérisé par sa capacité et par sa bande passante (voir paragraphe 2.2.3). Comme vu dans la section 2.2, il existe plusieurs modèles théoriques du canal de transmission en fonction des types d'erreurs les plus fréquents.

Les sources de perturbations sont diverses et dépendent essentiellement du milieu où se trouve le canal de transmission. Les principaux types de bruits sont : les bruits galactiques

³(Binary/Quadrature)Phase Shift Keying

⁴Quadrature Amplitude Modulation

entre 20 MHz et 200 MHz dus aux rayonnements des différentes sources d'énergie de l'espace ; les bruits atmosphériques jusqu'à 20 MHz induit par les éclairs orageux ; le bruit industriel ; le bruit urbain ; les micro-coupures correspondant à de courtes interruptions du signal ; les sauts de phase ; la diaphonie lors de l'acheminement de plusieurs liaisons par un même câble.

2.3.2.5 Démodulation

La démodulation permet de récupérer chaque symbole émis à partir de chaque signal modulé reçu de durée τ_m . Le démodulateur fournit une tension continue ou un symbole binaire dans le cas où sa sortie est quantifiée. Le démodulateur fournit au bloc décodeur une séquence binaire $\tilde{D}(x)$ qui représente l'information émise à laquelle est superposée une séquence d'erreur $E(x)$, $\tilde{D}(x) = D(x) + E(x)$.

2.3.2.6 Décodage de canal

Comme le décrit le théorème fondamental du codage canal, pour se rapprocher de la capacité du canal de transmission, il est nécessaire de coder l'information avant de la transmettre. Au niveau du récepteur, le décodage canal consiste dans un premier temps à détecter la présence d'erreurs dans l'information et puis dans un deuxième temps de les corriger. A partir de ces deux actions découlent trois principales stratégies : les stratégies ARQ⁵ qui se limitent à détecter la présence d'éventuelles erreurs, la correction s'effectuant par retransmission des blocs erronés ; les stratégies FEC (Forward Error Correction) mettant en oeuvre les codes permettant la détection et la correction des erreurs sans aucune retransmission ; enfin les systèmes hybrides combinant les deux techniques.

2.3.2.7 Décodage de la source

Le décodage source consiste à reconstituer, par l'application de l'algorithme de décodage source (décompression), l'information originelle à partir de la séquence de substitution $D(x)$.

2.4 Environnement protocolaire

Pour des applications conversationnelles et de streaming vidéo, une hiérarchie de protocoles commune est utilisée. Cette hiérarchie suit un modèle en couches qu'on va détailler par la suite.

La couche *physique* traite les signaux, et fournit un service pour communiquer des bits. La couche *liaison de données* fournit l'abstraction d'un lien, et la capacité de transmettre et de recevoir des suites de bits, au dessus du lien. La couche *réseau* présente le concept d'un chemin. La couche *transport* fournit un canal bout-en-bout, qui pourrait être fiable ou pas, selon le protocole qui est utilisé. Les trois couches restantes sont moins clairement définies, et ont été fusionnées réellement dans l'architecture de TCP/IP qui s'est développée plus tard. Les interactions entre les couches sont contrôlées, et assurées principalement par les entêtes des protocoles que chaque couche ajoute à l'entête des paquets.

⁵Automatic Repeat Request

2.4.1 La couche physique

Le niveau physique regroupe les règles et procédures qui permettent d'acheminer des éléments binaires non structurés sur un support physique entre l'émetteur et le récepteur. Ce niveau prend en charge les aspects de codage et de modulation de signaux numériques électriques et optiques. La couche physique des réseaux sans fil est définie par la famille des normes IEEE 802.11, 802.11a, 802.11b, 802.11g. Ces différents standards définissent des techniques d'occupation du spectre et des techniques de modulation du signal sur des bandes de fréquences bien définies. La technique de modulation définit le symbole qui sera utilisé au niveau du signal pour représenter les données, elle définit aussi le nombre de bits qui sont codés par un symbole.

2.4.1.1 QoS dans la couche physique

L'apport de la qualité de service pour les communications sans fil ne se limite pas à l'amélioration des techniques du canal, mais nécessite aussi une intégration avec les couches supérieures, comme les algorithmes de compression de la source au niveau applicatif. L'utilisation de schémas de compression élevés peut améliorer la qualité des communications bout-en-bout. Parallèlement, avec plus de protection du canal, moins d'erreurs seront observées ce qui implique un meilleur acheminement des données. Vu que la capacité du canal sans fil est limitée, le codage au niveau de la source et l'état courant du canal doivent être pris en considération.

2.4.2 La couche Mac

La première fonction de la couche liaison de données est le partage équitable du support physique entre plusieurs stations en mettant en oeuvre des mécanismes distribués pour accéder au canal. La deuxième fonction consiste à délimiter les trames et à corriger d'éventuelles erreurs causées par la transmission physique. Dans le modèle de référence IEEE 802, la couche liaison de données est constituée de la couche LLC (Logical Link Control) et de la couche MAC (Media Access Control). La couche MAC IEEE 802.11 couvre trois domaines fonctionnels : le contrôle de l'accès au canal, la livraison fiable de données et la sécurité. Dans la suite de cette section, nous nous intéresserons uniquement aux deux premiers domaines fonctionnels.

2.4.2.1 Accès au canal dans la couche MAC 802.11

Le standard 802.11 se base principalement sur deux méthodes pour partager l'accès au canal : un accès distribué DCF (Distributed Coordination Function) et un accès centralisé PCF (Point Coordination Function). Pour permettre la cohabitation entre ces deux méthodes d'accès, la norme 802.11 a défini des intervalles de temps, appelés "superframe", qui sont partagés en deux périodes : une période sans contention durant laquelle le PCF est utilisé et une période de contention utilisant le DCF.

DCF La technique *DCF* [ZC03] est basée sur le mécanisme *CSMA/CA* (Carrier Sense Multiple Access with Collision Avoidance). L'utilisation de la technique *CSMA/CD* s'avère très coûteuse pour des réseaux sans fils. En effet, la disposition d'un circuit full duplex pour la détection de collision est nécessaire pour l'implémentation la méthode *CSMA/CD* [IA01]. Ainsi, la méthode *CSMA/CA* a été retenue pour les réseaux *WLANs*

pour des considérations économiques. Cette méthode abandonne la détection de collisions, tout en renforçant les mécanismes d'accès pour les éviter.

Avant chaque émission, la station désirant émettre écoute le support. S'il est libre pendant une certaine durée *DIFS* (Distributed Interframe Space) la transmission commencerait immédiatement. Si le support est occupé, la station doit attendre jusqu'à ce que le canal soit inactif durant une période *SIFS* (Short IFS). A la fin de cette période, une procédure de *Backoff* est enclenchée [ZC03]. Le principe général du mode d'accès *DCF* est illustré par la figure 2.9.

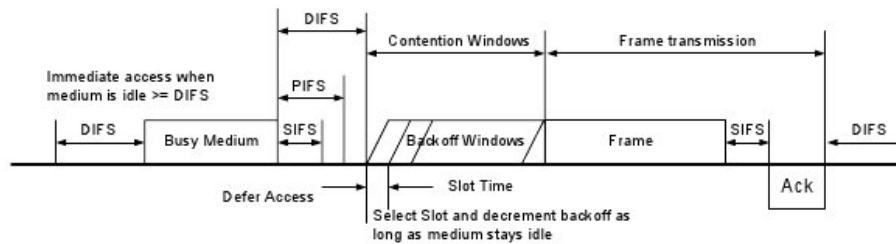


FIG. 2.9 – L'accès distribué au canal

Une station ayant correctement reçu un paquet, renvoie un accusé de réception (*ACK*) à la station émettrice. L' *ACK* indique à l'émetteur qu' aucune collision n'a eu lieu. Par contre, si l'émetteur ne reçoit pas d'acquiescement au bout d'un certain temps, la trame sera retransmise jusqu'à réception d'acquiescement par le récepteur. Enfin, si après un nombre défini de retransmissions, aucun accusé de réception n'est reçu, l'émission est abandonnée.

Dans le cas où la procédure de backoff est enclenchée, une station *S* désirant transmettre les données calcule son temporisateur "*Backoff_Timer*". Ce temporisateur est calculé comme suit :

$$Backoff_Timer = random() \times TS \quad (2.23)$$

Avec :

- *Random()* : nombre pseudo-aléatoire choisi entre 0 et $CW - 1$; où *CW* est la taille de la fenêtre de contention.
- *TS* : durée d'un time-slot définie comme étant l'intervalle de temps nécessaire pour une station pour savoir si une autre a accédé au canal au début du *time_slot* précédent [IA01].

Quand le canal devient libre, et après un *DIFS*, la station commence à décrémenter son temporisateur *time_slot* par *time_slot*. Lorsque la valeur de *Backoff_Timer* est égale à 0, la station peut alors envoyer. Si par contre au cours de la phase de décrément, une autre station *S* termine de décrémenter son temporisateur, la station *S* bloque son temporisateur. Elle pourra continuer de le décrémenter une fois la station *S* termine sa transmission [IA01].

Dans le cas d'une transmission échouée, une éventuelle collision s'est produite. Une collision est due si plusieurs stations transmettent sur le canal radio. Dans ce cas, la taille de la fenêtre de contention se double. Elle est donnée comme suit :

$$CW \leftarrow CW \times 2 + 1 \quad (2.24)$$

Une nouvelle procédure de *backoff* est enclenchée et le processus se termine lorsque la transmission est réussie ou abandonnée. La fenêtre de contention CW est une valeur entière définie par la couche physique, comprise entre CW_{min} et CW_{max} . La taille de la fenêtre est réinitialisée à CW_{min} dans les cas suivants : lorsqu'elle atteint la valeur CW_{max} , après un nombre *retry* de retransmission, ou bien après une transmission correcte.

PCF Il s'agit d'une méthode basée sur un algorithme centralisé permettant de transmettre des données synchrones. Elle est optionnelle et ne peut être utilisée qu'en mode infrastructure. Le point d'accès prend le contrôle et choisit les stations qui peuvent transmettre leurs données. Pour cela, il définit un *PC* (Point coordinator) qui lui permet de communiquer avec station du *BSS*. Le *PC* est chargé donc d'ordonner les transmissions et de distribuer le droit d'utilisation du support. Le *PC* détermine deux périodes de temps [ZC03] :

- La *CP* (Contention Period) : c'est une période de temps avec contention durant laquelle la méthode d'accès *DCF* est utilisée.
- La *CFP* (Contention Free Period) : c'est une période de temps sans contention durant laquelle la méthode d'accès utilisée est *PCF*.

Pour cette technique d'accès, le temps est divisé en intervalle de répétition du *CFP*, appelé aussi super-trame *PCF*.

Au début de la période sans contention, le *PC* acquiert le contrôle du support et garde ce contrôle pendant toute cette période. Si le support est libre pendant une durée *PIFS*, le *PC* commence par envoyer une trame *Beacon* pour informer les stations de la durée de cette période ce qui leur permet de mettre à jour leur compteur *NAV*.

Ensuite, le *PC* attend un *SIFS* ensuite il commence à envoyer les données aux stations destinataires par l'intermédiaire des trames *CF-Down* tandis que les stations émettrices utilisent les trames *CF-Up*. Les différentes trames de données transmises sont espacées par des intervalles *SIFS*. La période *CFP* se termine par l'émission d'une trame *CF-End*. La figure 2.10 illustre le déroulement de la transmission pour les deux périodes *CFP* et *CP*.

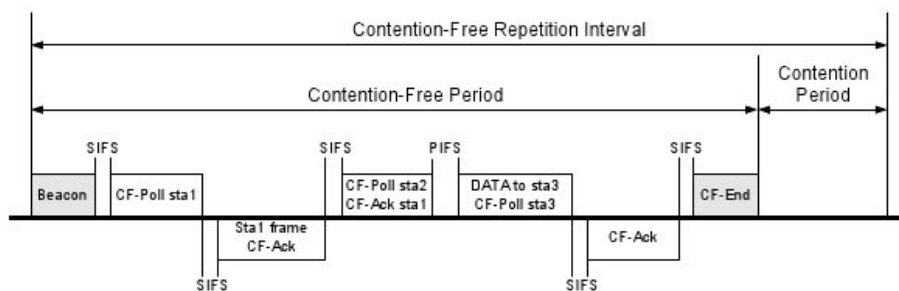


FIG. 2.10 – L'accès centralisé au canal

2.4.2.2 Livraison fiable dans la couche MAC 802.11

Le standard IEEE 802.11 a proposé plusieurs mécanismes pour palier au manque de fiabilité de la couche physique. Un bref descriptif de ces mécanismes est donné ci-dessous :

Retransmission La couche MAC 802.11 se base sur le système de correction d'erreurs ARQ (Automatic Repeat reQuest). Ce dernier emploie les codes de détection d'erreurs, les acquittements et les retransmissions pour assurer la fiabilité des données. En effet, la trame MAC intègre une valeur de contrôle CRC qui permet de vérifier l'intégrité des données au niveau du récepteur. Dans le cas où les données sont correctes, le récepteur envoie un acquittement à l'émetteur. Dans le cas contraire, l'émetteur attend l'acquittement durant un certain temps *timeout* et retransmet la trame en considérant que la précédente est perdue. L'émetteur répète cette opération un nombre de fois limité jusqu'à la réception d'un acquittement.

Utilisation de RTS/CTS Ce mécanisme a été proposé par la norme 802.11 pour éviter le problème de la station cachée engendré par la communication de deux stations éloignées avec une autre au milieu, la requête RTS (Ready To Send) et la réponse CTS (Clear To Send) ont été intégrées au mécanisme CSMA/CA avant un envoi d'une trame de données. Ainsi, lorsqu'une station émettrice détient le canal pour une transmission, elle commence par envoyer une requête RTS à la station réceptrice qui répond par une réponse CTS. Les messages RTS/CTS sont aussi reçus par toutes les autres stations qui doivent différer leurs transmissions pour éviter les collisions.

Fragmentation Afin de faire face à l'évanouissement rapide du signal responsable de la corruption des trames durant leur transmission, la norme 802.11 définit la fragmentation au niveau MAC. La fragmentation permet la décomposition des paquets provenant de la couche LLC en plusieurs trames MAC. Ceci permet de réduire la taille des trames transmises sur le canal sans fil et d'augmenter le nombre de trames livrées sans erreur.

2.4.2.3 QoS au niveau de la couche MAC 802.11

Le DCF défini par le standard 802.11 fournit un service au mieux (*best-effort*) qui n'offre aucune garantie aux stations pour l'accès au canal. Ceci pose inévitablement le problème de la qualité de service exigée par les flux multimédia qui sont très sensibles au délai de transmission et à la variation du débit. Plusieurs travaux ont été menés pour introduire des mécanismes de QoS au niveau de la couche MAC 802.11 en préservant son principe de fonctionnement. Ces mécanismes peuvent être classés en deux catégories : la différenciation de services et la garantie de service.

Différenciation de service au niveau MAC

La différenciation de services au niveau de la couche MAC est assurée par la définition de plusieurs classes de trafic et chacune sa priorité d'accès au canal de transmission. Ces priorités sont créées par la définition de différents paramètres de contention au canal entre les différentes classes. Les paramètres utilisés pour créer les classes sont présentés ci-dessous.

Arbitration IFS (AIFS) Les classes de trafics utilisent des IFS différents. La classe utilisant un petit IFS peut démarrer la transmission ou la procédure du Backoff plus rapidement qu'une classe utilisant un grand IFS. Ceci lui procure une priorité pour accéder au canal de transmission.

La taille d'une trame MAC/TXOP (Transmission Opportunity) Les classes de trafics sont autorisées à émettre une quantité de données différentes une fois le canal de transmission détenu. La quantité de données peut être différenciée en changeant la taille de la trame, ou bien, en envoyant plusieurs trames à la fois durant une opportunité de transmission TXOP. Les classes prioritaires pourront transmettre plus de données, ce qui fait augmenter leur débit de transmission.

Persistence Factor Un facteur P est associé à chaque classe de trafic, les classes prioritaires possèdent un petit facteur P . Dans [GH03], durant la procédure du Backoff, une valeur aléatoire r est générée pour chaque time slot. Le backoff est arrêté pour démarrer la transmission si $r > P$. Dans [BP02], pour chaque retransmission, la taille de la fenêtre de contention est multipliée par le facteur P au lieu de la multiplier par deux comme le propose l'algorithme standard DCF.

Backoff Time (BT) Plusieurs modifications ont été proposées pour l'algorithme du Backoff afin de construire des classes de trafics. L'approche, la plus simple, consiste à choisir des CW_{min} et CW_{max} différents pour chaque classe [BP02]. Ainsi, les classes prioritaires ont de petites valeurs CW_{min} et CW_{max} et par conséquent un BT moins important comparé aux classes moins prioritaires. D'autres approches proposent la modification de la manière dont le BT évolue [MM04]. Des algorithmes multiplicatifs ou bien additifs peuvent être choisis [HJ03]. Un algorithme multiplicatif réagit rapidement par des augmentations et des diminutions importantes. Tandis que les algorithmes additifs réagissent lentement avec un comportement lissé.

Garantie de service au niveau MAC

La différenciation de services fournit un meilleur support de qualité de service pour les trafics multimédia à charges faibles et moyennes. Pour des flux beaucoup plus chargés la différenciation de service est inefficace à cause des défaillances de la couche MAC 802.11 [GH03]. Dans ces conditions, le contrôle d'admission et la réservation de bande passante peuvent être efficaces pour garantir une qualité de service aux trafics existants. A cause de la nature du support sans fils et la méthode d'accès, ces deux approches sont difficiles à réaliser.

Le contrôle d'admission nécessite des moyens pour déterminer l'état du réseau et estimer les ressources disponibles. Les travaux dans [MBV01] proposent d'utiliser un algorithme MAC virtuel qui effectue un contrôle passif en utilisant des trames MAC virtuelles. Ceci afin d'estimer le niveau de service (débit et délais) et de déterminer si le canal peut supporter d'autres demandes de service. Dans [VL02], les auteurs proposent d'utiliser des paquets sondes pour estimer les ressources disponibles et décider de nouvelles admissions.

Concernant la réservation de bande passante, les auteurs dans [BP02] présentent le principe de la région d'admissibilité AR (Admissibility Region) qui se base sur la technique

du saut à jeton pour réserver la bande passante. Lorsqu'un nouveau noeud rejoint le réseau, l'AR réajuste les fenêtres de contention utilisées par tous les noeuds dans le réseau afin d'améliorer les performances.

2.4.2.4 Le standard IEEE 802.11e

Le nouveau standard 802.11e est l'approche principale pour la différenciation de services. Dans cette proposition une nouvelle méthode d'accès HCF (Hybrid Coordination Function) a été introduite. Elle décrit deux fonctions pour l'amélioration de la qualité de service : accès au canal HCF à base de contention et accès au canal HCF à base d'élection. Ces deux fonctions sont respectivement utilisées durant les périodes avec et sans contention pour fournir un support de qualité de service. EDCF (Enhanced DCF) est la méthode d'accès basée sur la contention.

Cette méthode améliore la méthode d'accès DCF par une approche distribuée permettant une différenciation de services. Cette approche définit 8 classes de trafic. Elle affecte des faibles valeurs de fenêtre de contention aux classes de trafics à hautes priorités. Ces trafics à hautes priorités ont alors plus de chances de transmettre sur le canal que celles à plus faibles priorités. Pour renforcer cette différenciation, le 802.11e propose l'utilisation des temps inter trames différents entre les différentes classes. Des faibles valeurs de AIFS (Arbitration IFS) seront utilisées par les classes à hautes priorités.

2.4.3 La couche réseau

Cette couche représente le coeur de l'architecture TCP/IP. Elle se base principalement sur le protocole IP (Internet Protocol) dont la première fonction est l'acheminement de paquets IP de bout-en-bout indépendamment les uns des autres jusqu'à leur destinataire final. Le protocole IP offre un acheminement au mieux (Best-Effort), non fiable et non orienté connexion. La version largement déployée actuellement est la version IPv4 antérieure à la version IPv6 qui a été conçue principalement pour élargir la plage d'adressage IP. L'adoption de IPv6 s'effectue progressivement notamment par les pays à forte population : la Chine, l'Inde, la Corée, etc.

Le service Best-Effort fourni par le protocole IP n'offre aucun contrôle, ni garantie, sur les conditions de transport de bout-en-bout des paquets IP. Grâce à sa simplicité, ce service a rencontré un grand succès dans le passé pour le transport des données asynchrones. Cependant, ce succès et de plus en plus contesté, actuellement, par la multiplication des flux synchrones, ou temps réel, qui exigent une certaine QoS pour leur transport. Cette QoS, appelée QoS réseau, se base sur plusieurs métriques qui caractérisent les performances d'un réseau IP. Les principales métriques sont :

- La bande passante : Elle représente le débit de bout-en-bout disponible sur le réseau.
- Le taux de perte : Il correspond au rapport du nombre de paquets perdus par le nombre de paquets émis durant la transmission.
- La gigue (Variation du délai) : Il s'agit de la variation des délais d'acheminement de bout-en-bout des paquets sur le réseau.
- Le délai de transmission (la latence) : Il représente le temps nécessaire pour traverser le réseau de bout-en-bout de l'émetteur jusqu'au récepteur. Pour répondre aux besoins des flux temps réel.

- Le modèle IntServ : Pour garantir la qualité de service.
- Le modèle DiffServ : Pour offrir une qualité de service différenciée.
- Le protocole MPL (Multi Protocol Label Switching) : Pour une meilleure gestion du trafic dans le réseau IP en considérant la QoS.

2.4.3.1 Routage avec Qualité de Service

Le principe du routage avec qualité de service est de rechercher un chemin entre deux noeuds satisfaisant certaines contraintes. Plusieurs métriques peuvent être utilisées telles que le délai, la bande passante ou encore le coût de transmission comme vue précédemment. Selon le type de contraintes, la recherche de routes optimales peut devenir un problème NP-complet. Le routage avec qualité de service ajoute en général à des protocoles de routage usuels un contrôle d'admission afin de sélectionner parmi les routes disponibles celles qui satisfont les contraintes du flux. Le principal problème de ce type de protocole est le surcoût engendré.

Routage QoS sur DSDV

Dans [CJS99], les auteurs présentent un protocole de routage avec qualité de service tentant de résoudre les problèmes de station cachée par une allocation dynamique d'unités TDMA [CJS99] (division du temps en unités appelées slots). Lors d'une demande de réservation, le protocole proposé évalue la quantité de bande passante disponible sur la route principale fournie par le protocole de routage DSDV [PB94] en évaluant le nombre d'unités TDMA disponibles sur chaque lien tout au long de la route.

Ticket Based Probing

Compte tenu du coût d'accès au médium élevé dans les réseaux sans fil, la recherche de routes par inondation peut devenir très coûteuse. Le but de Ticket Based Probing [CN99] est de limiter ce surcoût et de fournir des garanties de qualité de service. Ce protocole de routage QoS a été conçu pour des réseaux dans lesquels la mobilité est suffisamment faible pour ne pas poser de réel problème (scénario de type salle de conférence). La durée de vie des routes doit être grande devant le temps nécessaire à l'établissement ou à la restauration d'une route. Le protocole utilise une technique de réparation locale des routes.

CEDAR

CEDAR [PSB99], repose sur l'élection dynamique par les noeuds d'un coeur de réseau stable qui approxime un ensemble dominant minimum. Le processus distribué d'élection de ces noeuds est local et dynamique. Le rôle des noeuds du coeur est de propager efficacement des informations sur la bande passante disponible dans les liens, d'assurer le routage dans le réseau en impliquant un minimum de noeuds dans ce processus et de limiter autant que possible les diffusions. La distance de propagation des informations dépend de la qualité du lien en terme de stabilité et de bande passante disponible.

2.4.4 La couche transport

La fonctionnalité principale de la couche transport est le transfert des messages de bout-en-bout de l'émetteur vers le récepteur. Pour cela, elle emploie des protocoles de transport qui utilisent les services de la couche réseau (IP) pour assurer l'acheminement des

messages. Les deux premiers protocoles définis et largement déployés dans les réseaux IP sont : UDP (Usage Datagram Protocol) et TCP (Transmission Control Protocol).

Un Algorithme de Contrôle de Congestion (ACC) pour les flux multimédia peut être exécuté au niveau transport ou au niveau applicatif pour contrôler le débit d'émission ou le débit de réception d'un flux. Ceci réduit la congestion tout en assurant un partage équitable du débit disponible dans le réseau.

2.4.4.1 Contrôle de débit côté émetteur

Dans ce genre d'algorithme, l'émetteur est responsable de l'estimation du débit du réseau en se basant sur des retours d'informations et sur les conditions de transmission. Ils sont généralement utilisés pour des transmissions unicast. L'estimation du débit peut s'effectuer suivant deux approches : en utilisant des sondes, généralement le taux de perte des paquets, ou bien en utilisant des modèles prédéfinis.

Dans la première approche, les taux de perte sont utilisés comme critère de performances. L'émetteur fait varier le débit d'émission pour maintenir un taux de perte inférieur à un seuil qui maintient la QoS du flux.

En ce qui concerne l'approche basée sur un modèle, le débit d'émission est estimé en utilisant un modèle prédéfini, dans la majorité des cas, à partir d'une connexion TCP. Ces modèles se basent sur plusieurs paramètres, comme le taux de perte, le RTT et le MTU (Maximum Transit Unit), pour estimer le débit. Par conséquent, ce type d'algorithme évite la congestion d'une manière similaire à TCP .

2.4.4.2 Contrôle de débit côté récepteur

Ce type d'algorithmes est généralement utilisé pour les transmissions multicast de flux vidéo codés en couche. Le codage en couche d'une vidéo, permet de structurer cette dernière en une couche de base et plusieurs couches d'amélioration (voir chapitre 3). La couche de base peut être décodée sans les couches d'amélioration, mais elle offre une qualité vidéo limitée. L'ajout d'une ou de plusieurs couches d'amélioration durant le décodage permet d'augmenter cette qualité. Pour permettre à des récepteurs hétérogènes (débits de réception différents) de recevoir le flux vidéo, chaque couche est transmise sur un groupe multicast. Le récepteur décide, par la suite, des couches qu'il veut recevoir en s'abonnant ou pas au groupe multicast. Pour cela, le récepteur doit effectuer une estimation judicieuse de son débit de réception.

2.4.4.3 Contrôle de débit hybride

Dans cette catégorie d'algorithmes, le contrôle du débit est effectué par l'émetteur et le récepteur en même temps. Le récepteur décide du nombre de flux qu'il peut recevoir en se basant sur une estimation de son débit de réception. Tandis que l'émetteur ajuste continuellement le débit d'émission de chaque flux indépendamment des autres. Un algorithme de ce type, appelé DSG (Destination Set Grouping) a été présenté dans [SYCL96]. Dans DSG le récepteur s'abonne et se désabonne des groupes en fonction de l'état du réseau, déterminé à partir du taux de perte.

2.4.5 La couche application

La dernière couche du modèle TCP/IP comprend un ensemble de protocoles de haut niveau développés pour de nombreuses applications : la messagerie électronique (SMTP), le transfert de fichiers (FTP), le Web (HTTP) et le streaming (RTSP, RTP/RTCP).

Dans le cadre de notre étude, nous nous intéressons uniquement aux protocoles et aux applications de streaming vidéo qui, à l'opposé du téléchargement vidéo, permettent la visualisation d'un flux vidéo au fur et à mesure de sa réception.

Actuellement, le streaming vidéo sur les réseaux IP, filaires et sans fil, est devenu une réalité. Ceci s'explique par le développement de protocoles multimédia qui permettent la transmission et le contrôle de flux vidéo ainsi que le développement de codecs performants qui fournissent une qualité vidéo satisfaisante avec des débits relativement bas.

2.4.5.1 Protocoles multimédia

Le rôle principal des protocoles multimédia est de fournir des fonctionnalités de base aux applications pour le contrôle, la description et la transmission de flux multimédia. La majorité de ces protocoles sont définis par l'IETF, sauf la suite protocolaire du H.323 qui a été définie par l'ITU-T (International Telecommunication Union - Telecommunications). Une brève description de quelques protocoles est présentée ci-dessous :

Protocole H.323

La première version du H.323 a été publiée par l'ITU-T en 1996. Elle définissait un ensemble de protocoles et d'architectures pour la communication audio/vidéo/données sur les réseaux IP en mode centralisé. Parmi ces protocoles, nous avons le protocole H.225.0/Q.931 et le protocole H.225.0/RAS (Registration, Admission, and Status) pour le contrôle d'une session de communication, le protocole H.245 pour le contrôle des flux audio/vidéo entre entités communicantes et enfin la suite de protocoles T.120 pour la transmission de données.

Protocole SIP (Session Initiation Protocol)

Le protocole SIP est un protocole de signalisation défini par l'IETF pour contrer H.323, caractérisé par sa complexité. Il permet l'établissement, la modification et la libération de sessions multimédia. Une session multimédia peut représenter un appel téléphonique sur IP, une conférence multimédia ou une distribution de contenus multimédia. Pour cela, il se base sur un jeu de transactions requête/réponse codé en format texte (INVITE, REGISTER, BYE, CANCEL, OPTIONS). L'architecture SIP est constituée de plusieurs entités : l'agent utilisateur, le serveur proxy, le serveur d'enregistrement/localisation, le serveur de redirection. Chaque entité possède une fonctionnalité particulière dans l'offre de service multimédia.

RTSP (Real Time Streaming Protocol)

RTSP [HS98] est un protocole de niveau applicatif qui permet l'établissement et le contrôle d'une session multimédia entre un client et un serveur afin de transmettre un,

ou plusieurs, flux audio/vidéo en unicast ou en multicast. À l'instar de SIP, RTSP se base aussi sur un jeu de transactions requête/réponse, codé en format texte, qui offrent des fonctionnalités similaires à celle d'un magnétoscope (OPTION, DESCRIBE, SETUP, PLAY, PAUSE, STOP, TREADOWN). Ces requêtes permettent au client de demander et de contrôler la transmission d'un flux multimédia présent sur un serveur.

RTP/RTCP (Real Time Protocol/Real Time Control Protocol)

RTP [HS03] est un protocole de niveau applicatif qui offre des fonctions de transport de bout-en-bout pour les flux multimédia. Parmi ces fonctions, nous citons l'identification d'une source multimédia, la numérotation des paquets et la synchronisation des paquets. RTP n'offre aucun service de fiabilité et il est indépendant du protocole de transport. Pour contrôler la transmission des paquets RTP, le protocole RTCP est utilisé. Ce dernier définit cinq types de rapports qui peuvent être échangés périodiquement entre l'émetteur et le récepteur.

2.4.5.2 Codage source solide

Plus le codeur vidéo est efficace, plus le paquet individuel vidéo est important. Lorsque la compression est très efficace, la perte d'un seul paquet peut avoir un effet catastrophique. Il faut alors utiliser un puissant mécanisme de récupération, comme des codes FEC complexes, pour en réduire la probabilité. Inversement, lorsque la compression n'est pas très efficace, la perte d'un paquet a peu d'importance. Dans ce cas, des techniques de masquage permettent de réduire, voire de masquer totalement l'effet de la perte. Un mécanisme de récupération léger peut être utilisé.

Il faut donc ajuster soigneusement l'efficacité de la compression en tenant compte de l'effet des pertes et de l'efficacité des techniques de masquage et du mécanisme de récupération. Il est possible alors de diviser de manière optimale la largeur de bande disponible entre les données vidéo et les données redondantes.

En d'autres termes, il est toujours utile d'optimiser conjointement les paramètres du codeur à la source et du codeur de canal (une technique qui sera citée appelée "codage conjoint source canal"). Dans le cas des communications multimédias, il s'agit d'exploiter la résistance aux erreurs qui peut être incorporée dans les flux de données multimédias compressés, au lieu d'utiliser des codes FEC ou des protocoles de communication complexes.

Les codeurs vidéo utilisent différentes techniques pour compresser efficacement la vidéo. Cependant, ces techniques sont déjà détaillées dans le premier chapitre de ce document. La prédiction est l'une des techniques les plus importantes en termes d'efficacité de la compression : l'image en cours est prédite à partir de l'image transmise précédemment. De ce fait, les paquets vidéo dépendent des paquets précédents. S'ils n'ont pas été reçus, le paquet en cours est inutile. C'est la propagation de perte. Il est possible d'échanger l'efficacité de compression par la robustesse en réduisant le taux de prédiction (c'est-à-dire en augmentant le codage intra) : on limitera ainsi les dépendances, ce qui permettra d'arrêter la propagation de perte.

Le retard de transmission peut également constituer une alternative à la robustesse. Les paquets vidéo peuvent être réorganiser dans des tampons spécifiques, entrelacement, de telle sorte que les paquets vidéo consécutifs ne représentent pas des données vidéo adjacentes. Cela permet de délocaliser l'effet des pertes et de faciliter le masquage. Une longue salve de paquets perdus affectera des parties de la vidéo très éloignées l'une de l'autre et il sera alors possible de masquer efficacement les parties perdues à partir des données vidéo adjacentes.

Le masquage est généralement appliqué à l'aveugle dans le récepteur. L'émetteur peut toutefois coder des algorithmes d'optimisation, données de masquage, qui en renforcent l'efficacité. Ce processus absorbe évidemment une partie de la largeur de bande disponible. Toutes ces techniques sont très efficaces, mais il est particulièrement difficile de choisir un ensemble optimal de paramètres, surtout dans une configuration comptant de nombreux récepteurs dont les canaux sont conditionnés différemment.

2.4.5.3 Gestion de la latence et la gigue

Au niveau applicatif, il n'existe pas de mécanisme qui permet de diminuer la latence des transmissions. Cette diminution est prise en charge par des mécanismes de QoS au niveau réseau et au niveau MAC exposés dans le chapitre précédent. Par contre, pour faire face à la gigue qui représente la variation des délais de réception des paquets, un buffer est utilisé au niveau du récepteur afin d'absorber cette variation. L'utilisation d'un buffer permet le fonctionnement des autres mécanismes de QoS, présentés précédemment, comme l'ARQ, la FEC, etc. Ainsi, la taille du buffer est un compromis entre la latence introduite par ce dernier et la fiabilité de la transmission.

2.5 Conclusion

Dans la première section de ce chapitre, nous avons présenté le modèle de deux types de canaux, le canal gaussien et de Rayleigh, ainsi que leur performances en sortie du décodage de canal en terme de taux d'erreur binaire. Dans la deuxième section, et dans le cadre des transmissions multimédias, nous avons introduit la chaîne de transmission vidéo dans les réseaux IP. Enfin, la troisième section donne une description des différents protocoles, dont le but dépend de la couche dans lequel il est mis en place, et qui garantit la fiabilité d'une transmission vidéo.

Les applications et les protocoles de niveau applicatif se contentaient d'exploiter les services qui sont disponibles au niveau des couches inférieures. Étant donné que les conditions de transmission sont dynamiques, débit, taux de perte, latence et la gigue, les applications multimédias subissent les variations de ces conditions avec des réactions limitées à leur niveau en utilisant des mécanismes de QoS présentés dans ce chapitre. Durant ces dernières années, plusieurs techniques ont été utilisées par les applications multimédia afin de palier aux variations de ces facteurs et de minimiser leurs effets sur la qualité de la vidéo perçue par le récepteur. Ces techniques seront abordées dans le chapitre suivant.

Chapitre 3

Les différentes approches de transmission vidéo H.264\AVC sur les réseaux sans fil

3.1 Introduction

Les pertes de paquets dégradent considérablement la qualité de la vidéo décodée à la réception. Cette dégradation est accentuée par l'effet de propagation d'erreurs due à la dépendance du décodage des images I, P et B. En effet, la perte d'un paquet de l'image I provoque une erreur sur cette image et aussi des erreurs sur les images P et B qui appartiennent au même GOP¹. Des mécanismes ont été développés afin de faire face aux pertes des paquets et de minimiser leur impact sur la qualité de la vidéo.

Ce chapitre présentera ces différents mécanismes qu'on peut classer en plusieurs familles selon le niveau (la couche OSI) et l'ordre d'exécution (avant l'émission ou après la réception). Cette classification ne se veut en aucun cas exhaustive, mais nous essayons de distinguer les traits principaux. Il est important de noter que la plupart des systèmes proposés sont des combinaisons des principaux systèmes.

3.2 Réseaux à commutation de paquets vidéo

Internet est un réseau à commutation de paquets dans lequel le concept de connexion n'existe pas. Chaque paquet est routé indépendamment des autres.

Lorsqu'un paquet arrive à un routeur entre la source et la destination, il attend son tour dans une file d'attente pour être transmis. De cette manière, les ressources sont utilisées avec une efficacité maximale, les liaisons étant utilisées dès que les paquets sont disponibles.

Cependant, les connexions ne sont pas indépendantes les unes des autres car le comportement d'une connexion va affecter celui des autres connexions avec lesquelles elle partage des ressources du réseau.

¹Groupe of Pictures

Il n'est donc pas possible dans l'internet traditionnel d'obtenir des garanties d'allocation de ressources, le réseaux se contentant d'acheminer les paquets du mieux qu'il peut.

Il n'y a aucune garantie sur le fait que les paquets atteindront effectivement leur destination, pas plus qu'ils ne l'atteindront dans l'ordre. Pour obtenir un rapport fiable, les paquets doivent être acquittés par un protocole de transport de bout en bout en utilisant par exemple, le protocole TCP².

En cas de non-acquittement d'un paquet, l'émetteur peut réémettre le paquet non acquitté qui devra à nouveau traverser tout le réseau (les routeurs intermédiaires n'en gardant pas de copie). Ce type de service, offert à chaque utilisateur, indépendamment des besoins des applications, est connu sous le nom du meilleur effort possible (ou best effort).

Une conséquence naturelle de cette architecture est que le réseau peut être sujet à des phénomènes néfastes comme la congestion. En effet, lorsque la demande des utilisateurs excède la capacité du réseau, on observe un phénomène de saturation qui se traduit par une dégradation des performances globales du réseau.

Les délais de bout en bout et le taux de perte des paquets augmentent alors rapidement, dégradant ainsi le débit effectif du réseau. Pour éviter ces problèmes, des mécanismes de contrôle de transmission sont nécessaires pour limiter le taux d'utilisation des ressources du réseau en contrôlant le débit des paquets émis par les utilisateurs.

3.3 Contrôle d'erreur par anticipation pour la transmission de paquets vidéos (FEC)

Une méthode simple permettant d'aboutir à des flux résistants aux erreurs de transmission consiste à ajouter de la redondance dans le flux de manière à permettre la détection et la correction des erreurs. Ce type de redondance peut être introduit à la source ou par un codeur de canal, C'est une technique très efficace. Il s'agit de la transmission de paquets redondants, qui permet de récupérer les paquets erronés ou perdus côté récepteur. C'est ce qu'on appelle la FEC³.

Si on connaît le taux de perte, on peut ajouter la redondance juste suffisante pour les compenser. Malheureusement, dans la réalité, non seulement on ne connaît pas le taux de perte, mais il varie beaucoup en fonction du temps. Ce facteur, combiné au fait que le rendement de cette technique est du type "tout ou rien", la rend très difficile à utiliser : en effet, les erreurs sont plus nombreuses ou inférieures aux prévisions.

Si les pertes sont trop nombreuses, la capacité de récupération sera dépassée. La redondance ajoutée ne sera pas suffisante et les pertes ne seront pas récupérées. La qualité après décodage sera très mauvaise (effet tout ou rien). C'est pourquoi, pour ne pas courir de risque, les diffuseurs envisagent généralement le pire scénario et choisissent d'accroître la redondance au détriment de la vidéo, qui est alors davantage compressée ; la qualité

²Transmission Control Protocol

³Forward Error Correction

finale après décodage est alors moins bonne.

S'il y a moins d'erreurs que prévu, ce qui est probable lorsque le système est conçu pour le pire scénario, les pertes seront récupérées. La qualité après décodage sera garantie et ne sera pas affectée par les pertes mais on gaspille cependant de la capacité : on pourrait recourir à moins de redondance, ce qui permettrait d'avoir une vidéo moins compressée et de meilleure qualité. En principe, l'adaptation pourrait servir à équilibrer dynamiquement la redondance ajoutée et la compression vidéo en contrôlant les deux débits de la source et du canal utilisant un codage conjoint.

Cette opération peut être extrêmement complexe : le codage et le décodage des paquets redondants nécessitent de la mémoire et de la puissance de calcul. Les systèmes FEC efficaces doivent traiter de grands nombres de paquets vidéo. Par conséquent, le retard ajouté n'est pas nécessairement important, mais il peut être sensible. Dans ce qui suit, nous allons introduire les codes correcteurs d'erreurs utilisés en FEC.

3.3.1 Généralités et terminologie

On supposera ici que les codes sont constitués d'éléments binaires, prenant leur valeur dans $\{0, 1\}$ noté F_2 . Cette restriction permet de simplifier la présentation des codes sans vraiment nuire à la généralité [A.95].

Une description simple des codes en blocs peut être la suivante [eXC99]. Le codeur d'un code en blocs divise la séquence d'information en blocs de message comportant chacun K bits d'information. Le bloc de message, aussi appelé mot de message, est représenté par un K -uplet binaire de la forme $\bar{m} = (m_0, m_1, \dots, m_{k-1})$. Il y a donc au total $M = 2^k$ mots de message possibles.

Le codeur transforme indépendamment chaque mot de message \bar{m} en un mot de code $\bar{c} = (c_0, c_1, \dots, c_{N-1})$. Ainsi, à chacun des M mots de message correspond un mot de code unique et différent de tous les autres. Cet ensemble de M mots de code de longueur N est appelé un code en blocs noté (N, K) ou $c(N, K)$. Le rendement du code est K/N et le taux de codage est $(N - K)$. Chaque mot de code ne dépendant que des K bits du mot de message, on dira que le codeur est sans mémoire.

Un code en blocs $c(N, K)$ est dit linéaire si les 2^K mots du code constituent un sous-espace vectoriel de F_2^N , de dimension K . Cette définition induit que la somme de deux mots de code est encore un mot de code et que le mot de code nul est toujours élément du code. Un codage en blocs est une application $g()$ de l'ensemble F_2^K constitué par les K -uplets d'information vers l'ensemble F_2^N .

$$g : F_2^K \Rightarrow F_2^N$$

$$\bar{m} \Rightarrow \bar{c} = g(\bar{m}) \tag{3.1}$$

L'application $g()$ peut être mise sous une forme matricielle appelée matrice génératrice du code en blocs (appelée G par la suite). A chaque code en blocs linéaire $c(N, K)$ on peut associer un code en blocs linéaire dual. Tout mot du code dual est orthogonal à tout

mot du code $c(N, K)$. Ainsi tout mot c du code $c(N, K)$ est orthogonal aux lignes de la matrice génératrice H de son code dual :

$$\bar{c}H^T = 0 \quad (3.2)$$

Où $(:)^T$ indique la transposé. Le décodeur du code $c(N, K)$ va donc pouvoir utiliser la matrice génératrice H de son code dual pour détecter la présence des erreurs de transmission. La matrice H est appelée matrice de contrôle de parité. On remarque que :

$$GH^T = 0 \quad (3.3)$$

Cette propriété pourra permettre de trouver H . Une fois détectées, les erreurs seront corrigées en recherchant le mot du code le plus vraisemblable, c'est à dire celui qui sera à la distance de Hamming minimale du mot reçu. La distance minimale d_{min} entre ses mots est donnée par :

$$d_{min} = \min_{c_i, c_j \in c} d_H(c_i, c_j) \quad (3.4)$$

Où $d_H(:, :)$ désigne la distance de Hamming, notons que la distance de Hamming entre deux mots binaires est le nombre de bits différents entre ces deux mots. c_i et c_j sont deux mots du code et c est l'ensemble des mots du code. d_{min} est un paramètre pertinent pour mesurer les performances d'un code en blocs. Si un code en blocs linéaire possède une distance de Hamming égale à d_{min} , il peut détecter toutes les configurations de $d_{min} - 1$ erreurs et corriger toutes les configurations de t erreurs avec $t = \lfloor \frac{d_{min}-1}{2} \rfloor$, pouvoir de correction du code.

Un défaut majeur des premiers codes en blocs linéaires est leur complexité d'implémentation. Les auteurs ont donc cherché d'autres codes qui permettraient de réduire cette complexité. Ces recherches ont conduit aux codes cycliques. Un code en blocs linéaire est dit cyclique si un mot de code obtenu à partir d'une permutation circulaire à gauche d'un élément du code est toujours un élément du code. Parmi les codes en blocs linéaires cycliques nous pouvons citer le code $BCH(N, K, t)$ ayant pour paramètres :

$$\begin{aligned} N &= 2^a - 1 \\ N - K &\leq at \\ d_{min} &\geq 2t + 1 \end{aligned} \quad (3.5)$$

et pouvant corriger jusqu'à t erreurs, a étant un entier positif. Les codes linéaires cycliques sont en général définis dans des corps finis tels que les corps de Galois notés $GF(n)$ (Galois Field of order n). Pour des questions d'efficacité d'implémentation, il est préférable de prendre $n = 2^l$ avec l est entier positif.

D'autre part, on utilise généralement une représentation polynomiale pour les mots de code, ainsi que pour les opérations de codage et de décodage. Il est donc possible d'écrire le mot de code c sous la forme d'un polynôme :

$$c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1} \quad (3.6)$$

L'opération de codage peut alors s'écrire :

$$c(x) = m(x)g(x) \quad (3.7)$$

où $m(x)$ est la version polynomiale de m et $g(x)$ est appelé le polynôme générateur du code. Ce polynôme de degré $N - K$ est unique pour chaque code cyclique et le définit complètement. De la même manière que nous avons pour chaque matrice génératrice d'un code une matrice de contrôle de parité, pour chaque polynôme générateur $g(x)$ nous avons un polynôme de contrôle de parité $h(x)$ de degré K . Ces deux polynômes sont reliés de la manière suivante :

$$x^N - 1 = h(x) g(x) \quad (3.8)$$

Où $h(x)$ et $g(x)$ sont orthogonaux modulo $x^N - 1$, i.e :

$$h(x) g(x) = 0 \text{ mod } (x^N - 1) \quad (3.9)$$

Cette relation est utilisée lors du décodage. Notons que le code généré par $h(x)$ est lui aussi cyclique. Il est possible de remarquer que les codes en blocs en général et les codes linéaires cycliques en particulier peuvent être mis sous forme systématique. Un code est dit systématique si on retrouve le message dans le mot de code.

Les codes cycliques constituent une classe importante de codes. On trouve dans cet ensemble les codes de Reed-Solomon et les codes de Golay. D'autre part, la nature de ces codes a permis d'aboutir à des algorithmes de codage et de décodage très performants, y compris dans le cas de codes en blocs de grande taille. Le paragraphe suivant présente en détail le fonctionnement d'un code de Reed-Solomon.

3.3.2 Code de Reed-Solomon

Les codes de Reed-Solomon (RS) ont été définis par I.S.Reed et G.Solomon en 1958 [IR58]. Les codes RS ont l'avantage par rapport aux codes BCH⁴ de pouvoir coder des symboles n-aires plutôt que binaires, c'est à dire que les mots du code prennent leur valeur dans un corps q -aire de Galois.

Les messages sont divisés en blocs dont on a ajouté des informations redondantes à chaque bloc permettant ainsi de diminuer la possibilité de retransmission. La longueur des blocs dépend de la capacité du codeur.

Le décodeur traite chaque bloc et corrige les éventuelles erreurs. A la fin de ce traitement, les données originales seront restaurées.

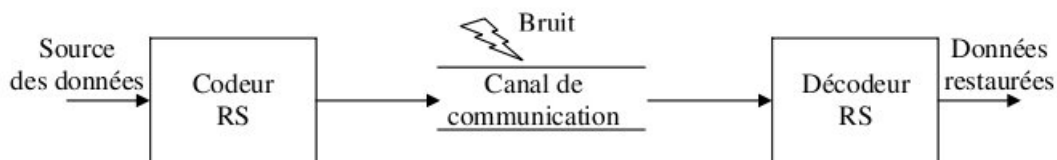


FIG. 3.1 – Le schéma général de Reed-Solomon

Ces codes ont une propriété importante, ils sont linéaires et font partie des codes *BCH*. Le codeur prend m symboles de données (chaque symbole contenant k bits) et calcule les

⁴Reprenant les initiales de ses inventeurs : Bose, ray-Chaudhuri et Hocquenghem

informations de contrôle pour construire n symboles, ce qui donne $n - m$ symboles de contrôle.

Le décodeur peut corriger au maximum t symboles, où $2t = n - m$. La figure 3.2 montre une trame constituée avec le codeur Reed – Solomon :

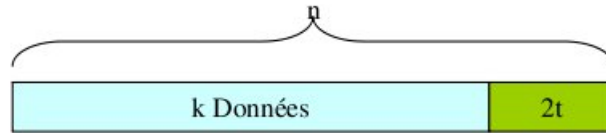


FIG. 3.2 – Mot-code de Reed-Solomon

Les codes de Reed-Solomon $RS(k, t)$ sont formés de n symboles, avec $n = q - 1$ au maximum et $q = 2^k$, chaque symbole appartenant à $GF(q)$ qui est le corps de Galois (Galois Field) à q éléments.

La distance minimale d'un code Reed – Solomon est :

$$d_{min} = 2t + 1$$

Autre propriété des codes Reed – Solomon, ils sont cycliques, c'est-à-dire, que chaque mot de code décalé engendre un autre mot de code. Tous les codes cycliques peuvent être réduits en gardant la même capacité d'erreur, mais le nouveau code formé n'est alors pas cyclique.

De plus, les codes de Reed – Solomon sont des codes non-binaires. Les codes sont représentés sur des "corps de polynômes" de $GF(2^m)$ [PK03] et non pas sur des champs de $GF(2)$. Les symboles sont définis comme les coefficients du polynôme et le degré indique l'ordre. Ainsi, le symbole avec l'ordre le plus élevé est reçu/envoyé en premier et le dernier symbole reçu/envoyé est celui dont l'ordre est moindre. La définition du corps de Galoi et ses opérations sont données dans l'annexe.

3.3.2.1 Technique de codage

Considérons un code de Reed-Solomon avec ses symboles dans $GF(2^k)$, où k est le nombre de bits par symbole. Soit

$$i(x) = c_{m-1}.x^{m-1} + \dots + c_1.x^1 + c_0 \quad (3.10)$$

le polynôme d'information, et soit

$$p(x) = c_{2t-1}.x^{2t-1} + \dots + c_1.x^1 + c_0 \quad (3.11)$$

le polynôme de contrôle, le code de Reed-Solomon sous sa forme polynômiale sera alors

$$c(x) = i(x).x^{2t} + p(x) = c_{n-1}.x^{n-1} + \dots + c_1.x^1 + c_0 \quad (3.12)$$

avec c_i appartenant à $GF(2^k)$ pour $i = 0, \dots, n - 1$.

Un vecteur à n symboles, $(c_{n-1}, \dots, c_1, c_0)$ est un code de Reed-Solomon si et seulement si son polynôme correspondant $c(x)$ est un multiple du polynôme générateur $g(x)$. La méthode courante pour construire un tel polynôme, est de diviser $i(x).x^{2t}$ par $g(x)$ et d'additionner le reste à $c(x)$. En effet,

$$i(x).x^{2t} = q(x).g(x) + r(x) \quad (3.13)$$

où $r(x)$ est le reste de la division de $i(x).x^{2t}$ par $g(x)$,

$$c(x) = i(x).x^{2t} + p(x) = q(x).g(x) + r(x) + p(x) = q(x).g(x) \quad (3.14)$$

pour que $c(x)$ soit un multiple de $g(x)$, soit $c(x) = q(x).g(x)$, il faut que $p(x) = r(x)$. Comme on travaille toujours sur des corps de caractéristique 2, l'opération de soustraction sera toujours égale à l'opération d'addition, soit de manière algébrique $-1 = +1$.

Cela nous donne une méthode pour construire le polynôme de contrôle, il suffit de prendre le reste de la division du polynôme $i(x).x^{2t}$ par $g(x)$.

Il nous reste encore à construire le polynôme générateur $g(x)$. Il est défini de la manière suivante :

$$g(x) = (x + \alpha)(x + \alpha^2) \dots (x + \alpha^{2t}) = x^{2t} + g_{2t-1}.x^{2t-1} + \dots + g_1.x^1 + g_0 \quad (3.15)$$

où les coefficients g_i appartiennent à $GF(2^k)$, et α est un élément primitif de $GF(2^k)$.

3.3.2.2 Technique de décodage

Souvent le terme décodage des codes correcteurs est employé pour désigner l'opération de " correction " dans la mesure où on peut la faire c'est-à-dire où le nombre de symboles erronés est inférieur à t (le pouvoir de correction). C'est en effet cette opération qui est la plus compliquée du point de vue théorique que pratique. Le décodage proprement dit, c'est-à-dire la recherche du mot d'information à partir du mot de code, est souvent évident parce qu'on choisit des codes systématiques.

L'idée de base du décodeur de Reed – Solomon est de détecter une séquence erronée avec peu de termes, qui sommée aux données reçues, donne lieu à un mot-code valable.

Considérons un code de Reed-Solomon $c(x)$ correspondant au code transmis, et soit $d(x)$ le code que l'on reçoit. Le code d'erreur est défini par :

$$e(x) = d(x) - c(x) = d(x) + c(x)$$

car - et + sont équivalents dans $GF(2^k)$.

Supposant que le polynôme des erreurs contienne v erreurs aux positions $x^{j_1}, x^{j_2}, \dots, x^{j_v}$ avec $0 \leq j_1 < j_2 < \dots < j_v \leq n - 1$. On peut donc redéfinir le polynôme des erreurs comme :

$$e(x) = e_{j_1}.x^{j_1} + e_{j_2}.x^{j_2} + \dots + e_{j_v}.x^{j_v} \quad (3.16)$$

avec

$e_{j_1}, e_{j_2}, \dots, e_{j_v}$: valeurs d'amplitude des erreurs.

$x^{j_1}, x^{j_2}, \dots, x^{j_v}$: emplacements des erreurs.

A partir du polynôme $r(x)$ reçu, on peut calculer le polynôme du syndrome $S(x)$ qui nous indiquera la présence d'éventuelles erreurs. Si tous les coefficients du syndrome sont nuls, alors le reste des étapes de décodage n'ont pas lieu d'être car le mot-code reçu ne contiendra pas d'erreurs. Par contre, si le syndrome est non nul, on devra calculer le polynôme de localisation des erreurs et le polynôme d'amplitude des erreurs. Il y a plusieurs méthodes de calcul de ces deux polynômes, par exemple le décodage selon l'algorithme d'Euclide et le décodage selon l'algorithme de Berlekamp – Massey. Une fois les polynômes calculés en utilisant l'algorithme de Forney, on calculera les valeurs à soustraire pour obtenir le mot-code sans erreur.

Les codes RS sont particulièrement adaptés aux 4 critères qui seront cités dans la section 3.3.7. Ils existent sous forme systématique et peuvent être raccourcis. D'autre part, on introduit facilement la connaissance de la position des pertes. Un certain nombre d'auteurs utilisent ces codes pour la génération des FEC soit avec une protection égale EEP⁵[JYC07] ou inégale [NTS06][HH06]

3.3.3 Turbo_codes en blocs

Les turbo-codes, inventés par C.Berrou et al [CB93] à télécom Bretagne, définissent une nouvelle classe de code correcteur d'erreurs dont le pouvoir de correction avoisine, dans des conditions de codage idéales, la limite théorique prévue par Shannon. Ces codes sont obtenus par la concaténation parallèle de deux codes convolutifs systématique récurrents.

Une première réalisation des turbo-codes sur silicium a été menée à son terme avec succès et a permis de valider le concept de turbo-code. Les excellents résultats obtenus par C.Berrou ont incité R.Pyndiah [PYN98] à concevoir un équivalent des turbo-codes convolutifs pour les codes en blocs. Leur choix s'est orienté vers une classe particulière de code en bloc, les codes itérés appelés aussi codes produit.

Les codes produit sont de puissants codes correcteurs d'erreur (ayant une grande distance minimale de Hamming). Ils sont construits par concaténation de deux ou plusieurs codes en blocs linéaires de faible pouvoir de correction. Considérons deux codes en blocs élémentaire C^1 et C^2 systématique de paramètres respectifs (n_1, k_1, d_1) et (n_2, k_2, d_2) , où n_i représente la longueur du code C^i , k_i la longueur du message et d_i la distance minimale de Hamming de C^i . Le code produit $C = C^1 \otimes C^2$ se présente alors sous la forme d'une matrice $[C]$ de n_1 lignes et n_2 colonnes dans laquelle :

Les symboles binaires d'information définissent une matrice de k_1 lignes et k_2 colonnes.

Chacune des k_1 lignes de la matrice $[M]$ est codé par le code C^2 .

⁵Equal Error Protection

Les n_2 colonnes de la matrice ainsi formée sont codées en utilisant le code C^1 pour former la matrice $[C]$.

La figure 3.3 illustre le principe de construction d'un code produit.

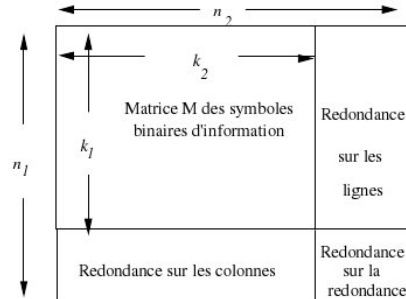


FIG. 3.3 – Principe du codage d'un code produit

Les paramètres (n, k, d) du code produit C s'obtiennent à partir de ceux des codes C^1 et C^2 par $n = n_1 \times n_2$, $k = k_1 \times k_2$ et

$d = d_1 \times d_2$. Le rendement du code produit est égal à $R = R_1 \times R_2$ où R_i est le rendement de C^i .

3.3.4 Codes raccourcis

Les code BCH et RS ne permettent pas de prendre une valeur quelconque de N . Or il est intéressant de pouvoir coder un bloc de taille quelconque. Une technique pour y arriver consiste à coder un bloc de taille N plus grande que celle souhaitée mais remplissant les conditions des paramètres du codeur, puis d'enlever i données au résultat pour atteindre la taille voulue. On obtient ainsi un code $C(N - i, K - i)$ de capacité de correction et de pouvoir de détection au moins égales à ceux du code $C(N, K)$ dont il dérive [Liu97].

3.3.5 Codes convolutifs

Pour ces codes, chaque bloc de n éléments binaires en sortie dépend non seulement des k éléments binaires présents en entrée mais aussi des m blocs de k éléments binaires précédents. $m + 1$ s'appelle la longueur de contrainte, souvent $k = 1$. Le taux de codage est $R = n/k$, le codage se fait avec des registres à décalage et des ou exclusif. La figure 3.4 montre un exemple de codage convolutif.

Dans cet exemple à un éléments binaire d'entrée correspond deux éléments binaire de sortie qui sont déterminés par les équations suivantes :

$$c_k^1 = d_k + d_{k-1} + d_{k-2} \qquad c_k^2 = d_k + d_{k-2}$$

Et on appelle les polynômes générateurs :

$$1 + D + D^2 \qquad 1 + D^2$$

ou en octal

$$[111] = 7 \qquad [101] = 5$$

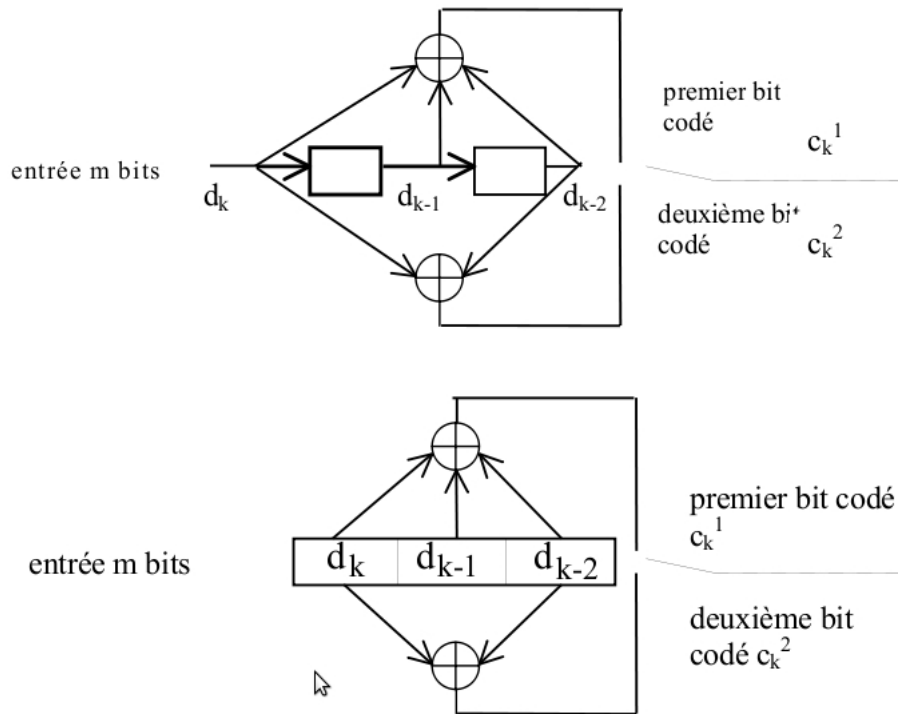


FIG. 3.4 – Exemple de codage convolutif

De façon générale le $j^{ème}$ bit de code a la forme, dans le cas où le bit d'entrée est unique :

$$c^j(k) = \sum_{i=0}^{m-1} g_{ji}d(k_i) \text{ où } k \text{ est l'instant d'observation.}$$

Il y a autant de polynômes générateurs que de sorties du code : ces polynômes sont :

$$g_j(D) = \sum_{i=0}^{m-1} g_{ji}D^i$$

3.3.5.1 La représentation en treillis

La représentation en treillis est la représentation la plus courante du codage facilitant l'algorithme de décodage, il représente l'état du codeur à chaque instant k , cet état est représenté par $\{d_{k-1}, d_{k-2}, \dots, d_{k-m-1}\}$ où m est le nombre de blocs pris en compte dans le codage du bloc actuel et qui le précédent. A chaque arrivée d'un élément binaire d_k , une sortie (un mot de code) est générée, puis juste après le codeur passe dans l'état suivant qui est $\{d_k, d_{k-1}, \dots, d_{k-m}\}$. Le treillis est formé de noeuds reliés par des branches. Les noeuds représentent les différents états du codeur possibles. Il y en a 2^{m-1} s'il y a une entrée et $2^{(m-1)k}$ s'il y a k entrées, les branches représentent les différentes transitions possibles d'un noeud à un autre (ou d'un état du codeur au suivant) lors de l'arrivée d'un bit d'entrée.

La figure 3.5 montre le treillis de l'exemple précédent, prenons les états 00,01,10,11. Partant, par exemple de l'état 00, l'arrivée d'un 0 mène le codeur à l'état 00 (transition en pointillé pour l'arrivée d'un 0) et l'arrivée d'un 1 mène le codeur à l'état 10 (transition en trait plein pour l'arrivée d'un 1). A chaque branche on peut associer le mot codé. Le

bit d'entrée	état du registre $d_{k-1}d_{k-2}$	contenu du registre	bits de sortie
0	00	000	00
1	00	100	11
0	01	001	11
1	01	101	00
0	10	010	10
1	10	110	01
0	11	011	01
1	11	111	10

TAB. 3.1 – Etapes de codage

schéma de treillis correspondant à cet exemple est donné par la figure 3.9.

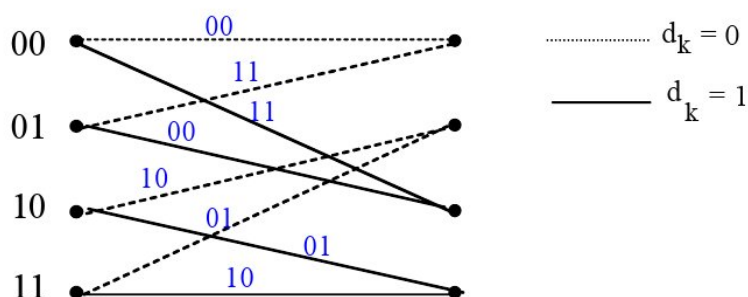


FIG. 3.5 – La représentation en treillis d'un code

3.3.5.2 Algorithme de Viterbi

Il consiste à rechercher dans l'arbre le chemin qui correspond à la séquence la plus probable, c'est-à-dire celle qui est à la distance minimale de la séquence reçue ou encore la séquence la plus probable.

Dans l'exemple qui suit (basé sur le codeur déjà décrit dans la section précédente) on suppose que le démodulateur fournit des 0 et des 1 (décision dure). A chaque fois qu'un groupe de 2 éléments binaires arrive, on examine toutes les branches possibles du treillis, on calcule la distance entre les éléments binaires affectés aux branches et les éléments binaires reçus, on ne garde que les branches donnant lieu à une distance minimale (ces branches forment le chemin survivant), et on affecte l'extrémité du chemin survivant d'une métrique égale à la somme de la métrique précédente et de la distance de la branche retenue. La figure 3.6 montre 3 étapes dans la trace d'exécution du processus de décodage de l'exemple donné dans la section précédente.

3.3.6 Les codes poinçonnés

Les caractéristiques des canaux étant variables, il est préférable que le schéma de codage soit capable de s'adapter à ces changements. Le poinçonnage de code permet à

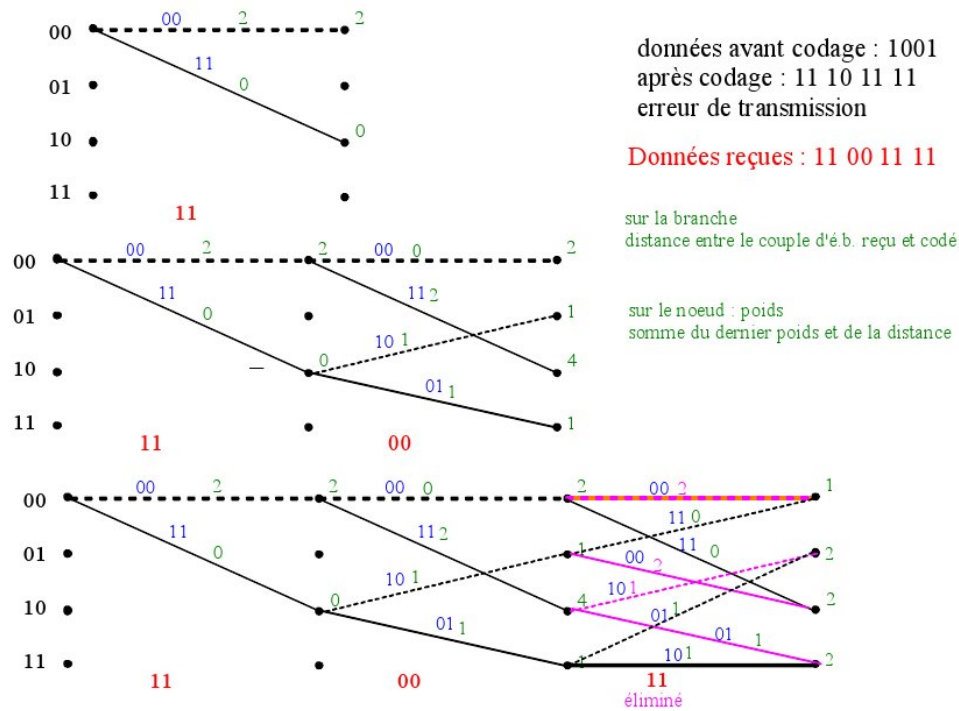


FIG. 3.6 – Algorithme de Viterbi

une paire codeur-décodeur de changer le taux de rendement du code et ainsi ses capacités de correction, sans changer sa structure de base [Liu97]. Les codes en blocs et les codes convolutifs peuvent être poinçonnés. C'est à partir des codes convolutifs qu'a été créée une famille de codes très intéressante, les codes RCPC [P.F97] (Rate Compatible punctured Codes).

Pour réaliser un code convolutif poinçonné, on utilise un code convolutif de faible taux de rendement $1/k$ (code mère) qu'on poinçonne périodiquement à la période pour obtenir une famille de codes de taux de rendement δ/v , ou v peut varier de $\delta + 1$ à δk . Prenons par exemple un code mère de taux de rendement $1/2$ et décidons de prendre une période de poinçonnage de $\delta = 4$. Dans ces conditions les matrices de poinçonnage comme le montre le tableau 3.2 sont des matrices à 4 colonnes et 2 lignes. Les 0 dans les matrices représentent les bits sortant du codeur mère mais enlevés lors du poinçonnage et les 1 ceux qui seront effectivement envoyés. Pour transmettre au taux $4/5$, on utilisera la première matrice. Ainsi, au lieu de transmettre 8 bits pour 4 bits d'information, on n'en transmettra que 5.

On dira de deux codes qu'ils sont compatibles en débit si tous les bits du code de plus haut taux de rendement sont utilisés dans le code de plus bas taux de rendement. Ainsi les codes poinçonnés de taux $4/5$ et $4/6$ sont compatibles en débit puisque tous les bits pris par le premier (code de plus haut taux de rendement) se retrouvent dans le deuxième (code de plus bas taux de rendement). Le décodage d'un code RCPC [P.F97] se fait avec l'algorithme de Viterbi [NS94], qui pour chaque taux changera sa métrique de calcul du meilleur candidat, les bits effacés n'étant pas comptés lors du calcul du code le plus probable.

Un code convolutif de type RCPC a l'avantage de permettre aux auteurs de l'article [Kar09] de générer très simplement différents niveaux de protection en sortie de son codeur de canal, ce qui leur permet d'obtenir facilement des FEC hiérarchiques, ils ont utilisé la taille des unités NAL comme critère de différenciation de protection ainsi que le type de celles-ci.

taux de rendement du code	4/5	4/6	4/7	4/8
Matrices de poinçonnage	1111	1111	1111	1111
	1000	1010	1110	1111

TAB. 3.2 – Exemple de matrice de poinçonnage

3.3.7 Choix des codes correcteurs

La section précédente nous a montré que le décodeur de canal pourra disposer à tout moment de l'information sur la position des pertes. Cette caractéristique importante du canal de transmission doit être prise en compte dans le choix du code. Énumérons les caractéristiques souhaitées pour un code dans le contexte particulier de notre étude :

3.3.7.1 Temps de codage/décodage compatible avec du temps réel

Il est bien évident que puisque nous visons des applications temps réel, en aucun cas l'ajout d'un bloc de codage/décodage canal ne doit rajouter des délais trop importants. Un code en bloc voit sa complexité calculatoire augmenter de manière exponentielle lorsque la taille des blocs augmente.

Pour les codes convolutifs, c'est avec la mémoire du code que la complexité augmente exponentiellement. D'autre part à performance pratiquement égale, un RCPC est beaucoup plus simple à décoder que le code convolutif classique [Liu97]. Tout dépendra donc de la taille des blocs à coder.

3.3.7.2 Grand choix de taux de rendement

La procédure qui fixera la redondance à affecter aux données utiles, le fera de manière relativement précise. Il est donc nécessaire de trouver un code dont les caractéristiques permettent de répondre à cette précision. En d'autres termes, il serait intéressant de ne pas être contraint par les débits possibles du code.

Le code de Reed-solomon par exemple ne peut délivrer un rendement quelconque car ses paramètres sont contraints. Mais, depuis l'arrivée des techniques de raccourcissement (shortening) pour les codes en blocs et du poinçonnage pour les codes convolutifs on dispose pour tous les codes d'un moyen pour obtenir le débit que l'on souhaite.

3.3.7.3 Systématique

Le caractère systématique pour le code que nous devons utiliser nous permettra de ne pas décoder des données dans le cas où aucune erreur ne se serait produite. Cela nous

permettra d'avoir un gain de temps au décodage.

D'autre part dans le cas où la quantité de pertes dépasserait le pouvoir de correction du code, le caractère systématique permettrait de pouvoir au moins utiliser au décodage de source les données reçues alors que pour un code non systématique on ne pourrait rien en faire.

En fin, comme il est possible de supprimer le temps de latence induit par le codage de canal. Les données utiles peuvent en effet être transmises dès leur codage. La redondance sera générée à partir d'une copie et envoyée par la suite. Comme nous l'avons vu plus haut tous les codes ont une forme systématique.

3.3.7.4 Utilisation possible de la connaissance de la position des pertes

Jusqu'à la, peu de codes ont inséré dans leur phase de décodage la connaissance de la position des erreurs. Cette technique est utilisée par les codes RS, elle lui permet de doubler sa capacité de correction. Ils seront capables de décoder tant que la quantité de pertes restera inférieure ou égale à la quantité de redondance. L'utilisation de la connaissance de la position des erreurs est beaucoup moins simple pour les codes convolutifs.

3.4 Les FEC perceptuelles

Il existe une seconde classe de FEC. Il s'agit d'une méthode de FEC dite perceptuelle (FECP) (ou SFEC : Signal Processing FEC) proposée dans plusieurs articles et que les auteurs appliquent dans la transmission vidéo et vidéo sur internet. La différence essentielle par rapport aux FEC précédentes est que cette fois-ci les données redondantes sont des versions codées plus grossièrement des données originales. Ces approches peuvent être vues comme un cas particulier des descriptions multiples (voir la section 3.6). Par exemple un paquet de numéro n peut contenir des informations redondantes des $n - i$ paquets qui le précèdent.

L'intérêt de cette méthode est que contrairement aux FEC à codes correcteurs, la génération des données redondantes peut se faire sans attendre la fin du calcul du flux binaire original. L'idée serait de mettre en parallèle le codeur destiné aux données d'information et les codeurs destinés aux données redondantes. Le seul temps de latence occasionné serait alors dû à l'empaquetage des données.

3.5 La protection inégale

Dans la section précédente, nous avons introduit une manière simple d'utiliser les FEC et qui consiste à ne pas se préoccuper du type de données à protéger et de protéger également tout le flux binaire. On peut toutefois se demander si cette protection égale est réellement judicieuse lorsque l'on sait que dans un flux certaines parties sont plus importantes que d'autres. Prenons par exemple un flux binaire sortant d'un codeur vidéo prédictif tel que H264/AVC.

Nous savons que l'entête d'une image et la partition A, dans le cas de l'utilisation du DP (voir le chapitre I), sont très importantes et que la perte de ces derniers rend impossible le décodage du reste de l'image. Nous savons aussi qu'un groupe d'images sera perdu si l'image Intra servant de référence aux images inter qui la suivent est perdue. Ces deux exemples montrent qu'une protection en fonction de l'importance des données peut s'avérer plus pertinente puisqu'elle permettrait de limiter l'occurrence de cas de figure où on recevrait des données mais que leur décodage s'avérerait impossible sans les informations contenues dans des données perdues. Cette idée est l'une des bases d'une technique de protection inégale appelée UEP⁶ [H.C06].

Des solutions ont déjà été proposées pour exploiter les propriétés de scalabilité présentées dans H.264/AVC. Par exemple, ASO⁷ peut être utilisée pour transmettre de façon hiérarchique les images en fonction du niveau de mouvement du contenu. De plus, FMO⁸ et le partitionnement des données dans le flux compressé fournit une hiérarchisation sémantique et syntaxique des données vidéo.

Ces techniques de partitionnement du code binaire selon le degré d'importance, offertes par la norme de codage vidéo, facilitent l'utilisation d'une protection inégale des différents morceaux du flux. Il existe d'autres techniques dans la littérature qui peuvent être utilisées pour partitionner un flux en sous-flux chacun son degré d'importance. Le codage par description multiple et le codage en couches sont deux exemples qui font l'objet de la section suivante.

Cette inégalité de protection peut être assurée en utilisant un mécanisme FEC avec des taux de redondance différents suivant l'importance des morceaux du flux et/ou un mécanisme ARQ.

3.6 Codage par description multiple

On peut considérer le codage par description multiple (MDC Multiple Description Coding) [AEM99] comme une autre manière de renforcer la résistance aux erreurs sans recourir à des systèmes de codage de canal complexes ; inspiré du codage conjoint source canal (voir section 3.11). L'objectif du MDC est de créer plusieurs descriptions indépendantes pouvant contribuer à une ou plusieurs caractéristiques de la résolution spatiale ou temporelle, du rapport signal/bruit ou du contenu fréquentiel. Ces descriptions peuvent avoir la même importance (systèmes MDC équilibrés) ou des importances différentes (systèmes MDC déséquilibrés).

Plus on reçoit de descriptions, plus est grande la qualité de la vidéo décodée. Il n'existe pas de seuil en dessous duquel la qualité tombe (effet "tout ou rien"). C'est ce qu'on appelle le "mode dégradé". La robustesse vient de l'improbabilité que la même partie de la même image soit endommagée dans toutes les descriptions. L'efficacité du codage est réduite en fonction de la quantité de redondance restante entre les descriptions ; il est cependant vraiment possible de limiter le codage de canal du fait de la résistance améliorée

⁶Unequal Error Protection

⁷Arbitrary Slice Ordering

⁸Flexible Macroblock Ordering

aux erreurs. Des expériences ont montré que la description multiple est très robuste : la qualité livrée est acceptable même lorsque les taux de pertes sont élevés.

Les descriptions peuvent être abandonnées partout où cela peut s'avérer nécessaire : dans l'émetteur si la largeur de la bande passante est inférieure aux prévisions ; dans le récepteur s'il n'est pas nécessaire, ou pas possible, d'utiliser toutes les descriptions reçues. On parle de déployabilité (ou extensibilité). Notons cependant qu'il s'agit d'un avantage secondaire du codage par description multiple, qui ne vise pas à procurer l'extensibilité, mais la robustesse.

Les descriptions de la même partie d'une image doivent être décalées dans le temps autant que possible lorsque les flux sont multiplexés. De cette manière, une salve de pertes à un instant donné n'entraîne pas la perte de la même partie des données dans toutes les descriptions en même temps. En cas d'entrelacement, on peut utiliser le même critère, c'est-à-dire espacer autant que possible les descriptions de la même partie d'une image pour éviter qu'une salve de pertes ne provoque la perte de la même partie des données dans toutes les descriptions simultanément. Il faut tenir compte du retard ajouté du fait du décalage temporel, ou profondeur d'entrelacement.

3.7 Codage en couches

Le codage en couches (LC Layer Coding) est analogue au MDC. La principale différence réside dans la dépendance. Le codage LC vise à créer des couches dépendantes : il comporte une couche de base et plusieurs couches d'amélioration que l'on peut utiliser l'une après l'autre pour affiner la qualité de la couche de base après décodage.

On peut abandonner les couches aux endroits nécessaires, mais pas au hasard : il faut abandonner d'abord la dernière couche d'amélioration, mais jamais la couche de base. En effet, si elle n'est pas reçue, les couches suivantes ne peuvent rien améliorer. Le codage en couches est conçu pour offrir ce type d'extensibilité.

Il faut se doter de mécanismes de réparation pour garantir la fourniture de la couche de base au minimum. De plus, du fait de leur importance inégale, ces mécanismes doivent apporter une protection différente aux diverses couches afin de mieux tirer parti du codage en couches. Tous les réseaux n'offrent cependant pas ce genre de service.

Le codage en couches a besoin d'un codage de canal. Ce dernier peut aussi s'utiliser avec le codage par description multiple. Généralement, il est préférable d'adapter le niveau de protection d'une description ou couche donnée à son importance : il s'agit de la UEP .

3.8 Data Interleaving

L'objectif de cette technique est de minimiser l'effet des pertes consécutives sur la qualité de la vidéo décodée côté récepteur. Pour cela, une réorganisation des paquets vidéo est effectuée avant leur transmission, au niveau de l'émetteur, afin d'éloigner les paquets séquentiellement proches. En cas de perte séquentielle de plusieurs paquets durant la

transmission, cette perte sera répartie sur l'ordre original et minimisera la dégradation de la qualité de la vidéo. L'avantage de cette technique est qu'elle ne consomme pas de débit supplémentaire. Par contre, elle introduit un délai dû à la réorganisation des paquets vidéo du côté émetteur et du côté récepteur. Les auteurs de [JS08] proposent l'utilisation de l'interleaving combiné avec la FEC et la technique FMO.

3.9 ARQ : Demande de retransmission automatique

Une autre technique réactive qui augmente les performance d'une transmission existe dans la littérature sous l'appellation demande de retransmission automatique ou ARQ⁹ [Liu97]. C'est une technique très efficace pour améliorer la fiabilité basé sur la retransmission des paquets perdus. C'est sur cette technique que repose la distribution de contenu basée sur le protocole TCP. Si les pertes sont sporadiques, cette technique est très efficace car les paquets ne sont réellement envoyés qu'une fois.

Si, en revanche, les pertes sont fréquentes, les retransmissions peuvent aggraver encore l'encombrement et le taux de perte, ce qui crée un cercle vicieux. La retransmission est très utile dans les communications de point à point dans lesquelles on dispose d'une voie de retour. Cependant, dans le cas de la diffusion de point à multipoint, le diffuseur ne peut pas gérer toutes les demandes de retransmission individuelles.

Le retard ajouté par la retransmission est d'au moins un temps de propagation aller-retour. Mais chaque retransmission peut également être perdue, ce qui fait que le retard peut être important. Il s'agit d'un point critique pour la vidéo en continu : le retard d'un paquet retransmis peut être beaucoup plus long que le temps entre les arrivées et le flux risque de subir des interruptions. Ce retard s'accumule dans le tampon du récepteur, qui doit être suffisant pour compenser la variation des temps entre les arrivées (gigue).

Puisque les FEC n'assurent pas un taux de pertes nul. Plusieurs auteurs ont proposé des mécanismes combinant les FEC avec l'ARQ pour éliminer les pertes résiduelles dans la même couche ou deux couches différentes. Ces schémas sont alors regroupés sous l'appellation Cross-layer qui sera abordé par la section suivante.

3.10 Approche inter-couches (Cross layer)

Les techniques de transmission fiable vues dans les sections précédentes ainsi que les protocoles du chapitre précédent consistent à proposer une solution qui améliore les performances du réseaux et la qualité des données transmises, toutes ces solution sont basées sur le principe d'isolation des couches de la pile OSI.

Cette section présente un nouveau paradigme qui a émergé récemment sous l'appellation "Cross Layer" ou communication inter-couches. Le principe de base de ce concept est de permettre l'échange d'informations entre les couches adjacentes et non adjacentes afin d'améliorer les performances de transmission. Cet échange d'informations peut être mis en oeuvre suivant différents schémas. Parmi toutes les architectures Cross-layer proposées dans la littérature, deux modèles de communication peuvent être distingués

⁹Automatic Repeat reQuest

[SM05][NSM06] : La communication directe entre les couches et une base de données partagée entre les couches. La Figure 3.7 présente la différence entre ces deux modèles.

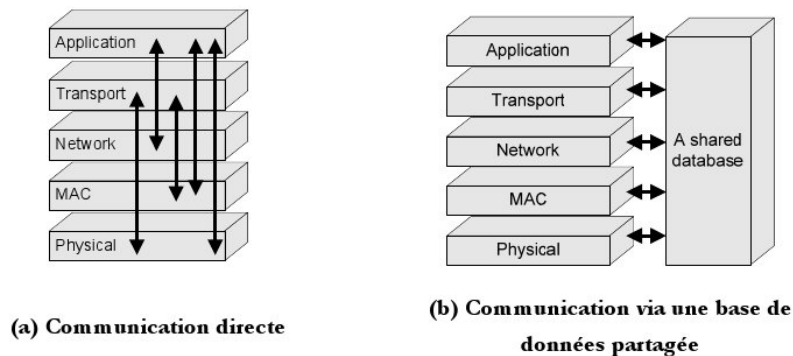


FIG. 3.7 – Les modèles de communication Cross-Layer

Le modèle le plus utilisé dans la littérature est celui qui utilise une communication directe entre les couches. Il permet de faciliter l'accès aux données en permettant à une couche d'accéder directement aux paramètres et aux variables d'une autre couche sans passer par un intermédiaire.

Il existe deux types de communication directe ; "in band" et "out of band". Le premier type est basé sur l'utilisation des informations portées par les entêtes ou des extensions de ces derniers, la couche IP par exemple peut accéder au champs de l'entête TCP pour indiquer une congestion dans le réseau. Le second type utilise un protocole dédié dit de signalisation. Les auteurs de [QW03] présentent un exemple parfait de ce type de protocole.

La communication "out of band" peut s'effectuer en définissant de nouvelles interfaces (API : Application Programming Interface) au niveau d'une couche qui seront utilisées directement par d'autres couches pour récupérer et configurer des paramètres de fonctionnement. Un exemple de cette API est proposé dans [V.T06].

La Figure 3.7 (b) illustre un autre modèle de communication Cross-Layer. Il utilise une base de données partagée qui contient des paramètres utilisés par toutes les couches de la pile qui peuvent s'informer de l'état des autres ou récupérer des paramètres de configuration. La base de données est aussi accessible par un système d'optimisation qui a le rôle d'initialiser les paramètres avec les valeurs adéquates. Ce modèle est utilisé par plusieurs architecture Cross-Layer [R.W06][S.K06a].

3.10.1 Les approches du Cross-layer dans les réseaux sans fil

Dans la littérature, plusieurs techniques Cross-layer ont été proposées pour améliorer les performances des transmissions sans fil. Au début, ces mécanismes étaient limités à l'interaction entre la couche physique et la couche liaison de données. De plus, les mécanismes proposés étaient indépendants et visaient l'amélioration d'une imperfection précise. Par la suite, nous allons introduire des travaux proposant des interactions avec les couches

supérieures et aussi à l'apparition d'architectures Cross-layer qui tentent de faire collaborer plusieurs couches, prenant en charge plusieurs paramètres, pour une optimisation globale.

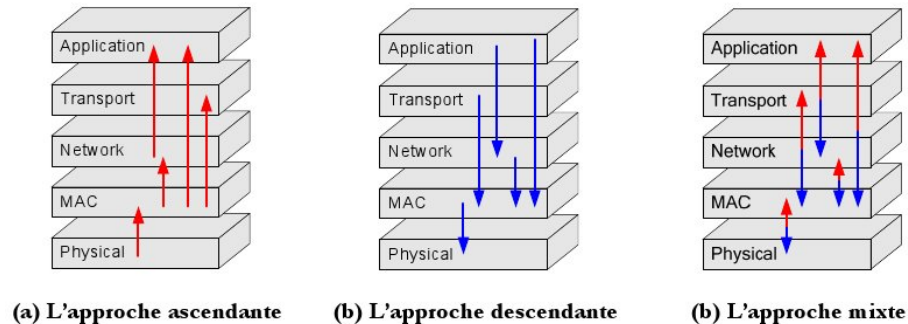


FIG. 3.8 – Les approches du cross-layer

Afin de simplifier la présentation de ces travaux, nous pouvons les classer en trois grandes approches identifiées dans [SM05][MVDS05]. Nous présentons, ci-dessous, une description de chacune de ces approches.

3.10.1.1 Les approches ascendantes

Les auteurs de [SS03] présentent les avancées majeures et les futures orientations de recherche dans le domaine des réseaux sans fil. La conception Cross-layer est présentée comme l'un des défis majeurs qui doit être relevé. L'article soulève le problème de contrôle de congestion du protocole TCP sur les réseaux sans fil et propose l'utilisation du mécanisme de notification explicite de congestion ECN¹⁰ afin de différencier les pertes causées par la congestion et les pertes causées par des interférences.

Le mécanisme ECN propose d'utiliser le bit ECN présent dans l'entête du protocole TCP. Ce bit est initialisé à '0' par l'émetteur et peut être fixé à '1' par un routeur s'il est congestionné. Quand le paquet arrive au récepteur, ce dernier informe l'émetteur sur l'état du bit ECN. Ainsi, l'émetteur pourra faire la distinction entre un état de congestion et un état d'interférence.

3.10.1.2 Les approches descendantes

L'ordonnancement optimal pour minimiser la congestion et la distorsion (CoDiO : Congestion-Distortion Optimized) est l'une des principales techniques dans les approches descendantes étudiées dans plusieurs travaux [E.S05][CM01][ESG05]. Le CoDiO se base sur des algorithmes d'ordonnancement qui décident des paquets vidéo à envoyer suivant leur importance et suivant le débit disponible. Ceci réduit la congestion et le délai de transmission dans le réseau.

Une autre forme du Cross-Layer consiste à insérer entre la source et la couche réseau responsable des demandes de renvoi, une couche FEC. Tous les paquets récupérés par le

¹⁰Explicit Congestion Notification

code correcteur ne font pas l'objet d'une demande de renvoi. Dans [JYC07], Jeong-Yong et al ont proposé une architecture inter-couches qui consiste à utiliser une FEC doublée dans les deux couches MAC et application comme solution proactive combinée avec une autre réactive par l'utilisation du protocole ARQ au niveau MAC dans un réseau IEEE 802.11e.

Une deuxième méthode est d'intégrer FEC et ARQ dans une même couche. Celle-ci consiste à envoyer un premier groupe de paquets avec une redondance faible, et de ne demander le renvoi de nouveaux paquets de redondance que si le nombre de paquets perdus dans le groupe est supérieur à la capacité de correction du code. Les résultats sont meilleurs qu'avec le système précédent mais c'est aussi une méthode plus lente, donc difficilement envisageable. Nous savons que toute demande de renvoi en multipoint est problématique.

Une troisième idée est de rationaliser les demandes de renvoi. Dans ce cas le récepteur indique le nombre de paquets non reçus. Ainsi un même paquet de redondance pourrait remédier à des pertes de paquets différents de plusieurs récepteurs. Nous voyons donc que même couplé avec des FEC, les méthodes basées sur de l'ARQ restent difficilement envisageables en temps réel.

3.10.1.3 Les approches mixtes

Les approches mixtes exploitent à la fois les interactions ascendantes et descendantes. Dans [MVDS05], les auteurs présentent une architecture Cross-layer pour analyser, sélectionner et adapter les différentes stratégies présentes sur les couches du modèle OSI. L'architecture proposée nécessite une optimisation conjointe du fonctionnement des protocoles existant à travers toutes les couches. La problématique du Cross-layer a été formalisée comme étant un problème d'optimisation avec l'objectif de sélectionner une stratégie commune à travers toutes les couches OSI.

Dans cette modélisation, uniquement les couches applications (APP), liaison de données (MAC) et physique (PHY) sont considérées. Chaque couche, PHY/MAC/APP possède respectivement un nombre défini de stratégies d'adaptation et de protection contre les erreurs. Les stratégies de la couche physique peuvent représenter les différentes combinaisons de modulation et de codage qui sont disponibles. Les stratégies MAC correspondent à des mécanismes ARQ, FEC, ordonnancement de trames, contrôle d'admission. Les stratégies applicatives correspondent à des mécanismes d'adaptation du codage vidéo, décomposition en paquets des images vidéo, lissage du trafic, classement du trafic suivant des priorités, ordonnancement de paquets et aussi les mécanismes FEC et ARQ appliqués au niveau paquets. La stratégie Cross-layer commune est définie par une combinaison optimale des stratégies des trois couches.

3.11 Codage conjoint source-canal (JSCC)

Dans les schémas de communications classiques, le message à transmettre subit d'abord une opération de compression, afin de lui enlever le maximum de redondance possible. Puis, une redondance structurée est ajoutée au message, afin de le rendre plus robuste à la transmission. En effet, transmettre un message à travers un canal introduit toujours une détérioration du message original. Le décodeur tente de retrouver le plus fidèlement

possible le message original, en s'aidant notamment de la redondance structurée.

En quelque sorte, la redondance présente dans le message est initialement enlevée du message, avant d'être ré-introduite sous une autre forme. Ces opérations ont été motivées par le théorème de séparation de Shannon. Ce théorème énonce que les opérations de compression et d'ajout de redondance peuvent être réalisées séparément sans perte d'optimalité. Cependant, ce théorème présente certaines limites.

Ces limites ont poussé les chercheurs à entreprendre des recherches sur le codage conjoint source canal, qui consiste à réaliser conjointement les opérations de compression et d'ajout de redondance. Ce type de codage présente une stratégie extrêmement répondue dans la littérature [JS08][XG08][XJ08][WNLJ06]. Cette approche est similaire à celle du cross-layer, la similarité entre ces deux paradigmes réside dans l'échange d'information entre couches, sauf que le concept du codage conjoint a ses spécificités qui seront montrées par la suite.

Le premier réflexe a été de régler les codeurs source de telle manière que des erreurs éventuelles de transmission aient un impact moins néfaste au décodage. Cette approche vise alors à rendre les codeurs source plus robustes, sans forcément prendre en compte les techniques de codage canal. Le schéma de cette approche est donnée par le premier schéma de la figure 3.9

Une autre stratégie consiste à allouer conjointement le débit de la source et du canal pour donner une meilleur performance et réduire le délai et la complexité du théorème de séparation de Shannon (schéma 2 de la figure 3.9). Cette stratégie peut être améliorée dans un contexte hiérarchique, où différents éléments transmis n'ont pas la même importance. Pour en bénéficier, il faut évaluer précisément l'impact d'une erreur commise sur tel ou tel élément transmis. A cet égard, les codeurs en sous-bandes, MDC et LC, sont des bases de travail bien adaptées.

D'autre part, les codeurs canal doivent prévoir des protections inégales des différents constituants du train binaire, UEP, pour que la contribution globale à la distorsion totale soit plus homogène. Cette stratégie hiérarchique a été prolongée dans un mécanisme de modulations codées [KM00]. Dans [Kar09], les auteurs ont utilisé une architecture JSCC¹¹ qui consiste à contrôler seulement le débit de canal pour réduire la complexité du traitement tout en améliorant la qualité perçue de la vidéo.

3.11.1 Equilibrage des rendements

Le codage conjoint a également des conséquences importantes au niveau système. En effet, il a été observé que la stratégie traditionnelle, consistant à comprimer au maximum la source pour un niveau de distorsion donné, n'améliore pas toujours les performances du système global. Ceci parce qu'elle rend plus sensible le train binaire aux erreurs de transmission; une erreur peut avoir des conséquences catastrophiques sur la compréhension du signal après décompression. Pour compenser cet effet, il faudrait alors mieux protéger le train binaire issu du codeur source contre les erreurs de transmission (FEC).

¹¹Joint Source Channel Coding

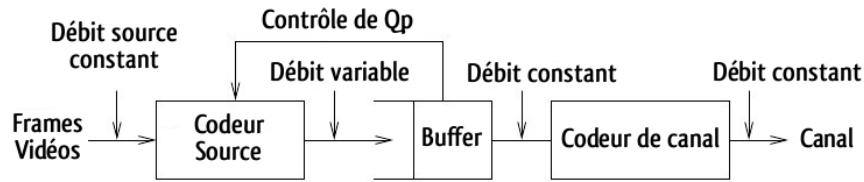


Schéma 1

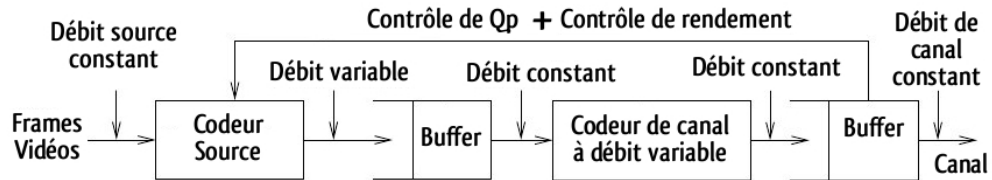


Schéma 2

FIG. 3.9 – Les architectures JSCC classique

En pratique, il existe toujours des contraintes de complexité matérielle et de délai qui limitent l'utilisation des codeurs canal très sophistiqués, de sorte que le bilan global, pour cette stratégie, n'est pas toujours intéressant.

Habituellement, on procède alors à un équilibrage a posteriori du rendement entre codeur source et canal afin de trouver un point de fonctionnement compatible avec l'application visée (qualité du signal reconstruit pour un canal donné). Ce mécanisme a été en particulier observé dans le contexte du visiophone.

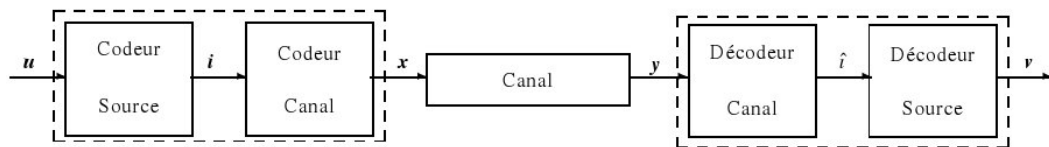


FIG. 3.10 – Le paradigme de SHANNON

Une meilleure stratégie consisterait à obtenir cet équilibrage entre codeur source et canal par une véritable conception conjointe. Cette stratégie consiste à trouver le meilleur couple (R_s, R_c) de telle sorte que :

$$(R_s, R_c) = \arg \min [D_{s+c}] \text{ avec } R_s + R_c \leq R \quad (3.17)$$

Où D_{s+c} représente la distorsion totale, R est la capacité du canal, R_s est le débit du codage de la source et R_c le débit du codage de canal. Notons que les deux paramètres R_s et R_c sont en relation, respectivement, avec le pas de quantification du codage source QP et le nombre de bits de contrôle ajouté par le codage du canal.

Dans [JS08], Janatana et al proposent la combinaison des trois techniques, FMO, FEC et interleaving. Cette proposition consiste à utiliser la FMO pour tout le flux, puisque la charge causé par son utilisation est relativement petite, et ajouter la FEC et l'interleaving

seulement si les conditions prédites du canal sont mauvaises. Cela veut dire que les débits alloués à la FEC sont diminués lorsque le canal n'est pas bruité.

Le modèle du canal adopté dans l'article de Janatana est le modèle Markovien présenté dans le chapitre précédent. Les paramètres du modèle sont calculés pour chaque image i , puis ils calculent la probabilité que le canal soit dans un bon état $P_g(i)$. Si cette probabilité est plus grande que celle de l'image précédente $P_g(i-1)$, le canal est prédit qu'il sera dans un bon état. Cette technique de codage conjoint utilisant une double protection est faite comme le montre les étapes de l'algorithme 3.1 :

Algorithme 3.1 Exemple de JSCC

1. Calculer le taux d'erreur des images précédentes à partir des information de Feedback.
 2. Traiter la matrice de transition et calculer la probabilité que l'état suivant du canal soit bon P_g , en mettant le canal dans un état "bon" comme une condition initiale.
 3. Traiter P_g et $P_g(i-1)$:
 - a. Si $P_g(i) \geq P_g(i-1)$, le canal est considéré bon
 - Utiliser FMO pour l'image i
 - b. Sinon $P_g(i) < P_g(i-1)$, le canal est considéré bruité
 - Utiliser FMO, FEC et Interleaving pour l'image i
 4. Répéter l'étape 1 jusqu'à ce que toutes les images sont traitées.
-

Dans [XG08] Xuejuan et al ont proposé un modèle de qualité de débit (R-Q) pour représenter les caractéristiques du codage H.264 en utilisant le PSNR comme métrique. Après la création d'un tel modèle, une optimisation d'allocation de débit de canal est faite en fonction de l'état de ce dernier. Cette optimisation est assurée en se basant sur une analyse de la distorsion totale sur le modèle R-Q proposé et les caractéristiques de corrections des turbo-codes. Un modèle similaire est proposé dans [WNLJ06] capable de contrôler les paramètres de codage, qui sont le débit de codage, paramètre de quantification et le nombre de paquets, en fonction des conditions du canal et le débit disponible dans le réseau pour optimiser la qualité de la vidéo reçue. Dans [XJ08] Xavier et al proposent un codage conjoint source canal basé sur l'utilisation des codes à longueur variable et les turbo-codes.

3.11.2 Décodage conjoint source-canal

Dans de nombreux standards de compression, tel que JPEG, H264, une étape de compression à l'aide d'un code à longueur variable, VLC¹², des données est effectuée. Les trains binaires générés à l'aide d'un VLC sont particulièrement sensibles aux erreurs de transmission, principalement à cause de la perte de synchronisation qui peut résulter de certaines erreurs introduites par le canal.

Une solution classique consiste à faire appel à un codage canal puissant ou à insérer dans le train binaire des marques de synchronisation (présenté dans la section 3.2) . Cependant, ceci conduit à une augmentation significative du débit binaire sur le canal. Des techniques plus récentes ont exploité la structure particulière des VLC afin de développer des

¹²Variable Length Coding

algorithmes de décodage souple de trains binaires générés à l'aide d'un VLC . L'ensemble des successions de mots de code possibles est décrite à l'aide de treillis exploitant certaines informations supposés disponibles a priori au niveau du décodeur (nombre de mots de code, longueur du train binaire à décoder, etc.). Ce type de récepteur source-canal et appelé "turbo décodeur".

Les performances obtenues sont bien meilleures que celles atteintes par des décodeurs classiques. L'ajout d'information sur la source, par exemple la prise en compte de la syntaxe que doit respecter le train binaire généré par le codeur source, améliorent encore les performances en réception.

Enfin, le décodeur peut mettre en oeuvre un estimateur au sens du maximum de vraisemblance (MV) ou au sens du maximum a posteriori (MAP) du train binaire émis. Dans le second cas, les probabilités a priori des mots de code doivent être disponibles au décodeur ; elles peuvent également être estimées. Les méthodes de décodage a longueur variable peuvent être divisées en 3 classes, décodage séparé, décodage avec estimation parfaite de la source et décodage avec estimation paramétrique de la source. Ces trois classes sont introduites par les trois sections qui suivent.

3.11.2.1 Méthode de décodage séparé

Dans ce premier cas, décodage de canal et de source sont réalisés séquentiellement et indépendamment. De plus, le décodage de source est effectué par lecture dans des tables et correspond donc à ce que l'on appelle un "décodage dur". Dans ce cas, seule la connaissance de la table de codage VLC est nécessaire au codeur et au décodeur et aucune information additionnelle de statistique de source n'est requise au décodeur. Le décodage dur de la source représente le schéma classique dans les systèmes de communication.

L'inconvénient essentiel de ces méthodes est qu'elles ne tirent pas parti des connaissances a priori liées à la source pour réaliser un décodage souple source ou pour un décodage-canal aidé par la source. De nombreuses études sur la base des hypothèses des deux autre types ont pourtant montré que des gains significatifs pouvaient être obtenus grâce à une connaissance exacte ou à une estimation de la statistique de la source. On peut donc dire que, pour un schéma de transmission et un canal donnés, cette méthode fournit la borne inférieure en terme de performance.

3.11.2.2 Méthodes de décodage avec estimation parfaite de la source

La structure de l'arbre VLC ainsi que la statistique des valeurs discrètes de la source, ou symboles, qui lui sont associées sont supposées parfaitement et définitivement connues du décodeur. Cette statistique peut être utilisée par le décodeur de source, dans une forme dite de décodage souple.

Les méthodes de décodage qui supposent une connaissance parfaite de la statistique de source au décodeur ne peuvent s'appliquer que dans des cadres théoriques que l'on retrouve peu fréquemment en pratique. Pour un schéma de transmission et un canal donnés, le deuxième type de méthode fournit la borne supérieure en terme de performance.

3.11.2.3 Méthodes de décodage avec estimation paramétrique de la source

Dans un certain nombre de cas, il est possible d'envisager de modéliser des sources réelles par des modèles paramétriques. Par exemple l'utilisation d'un modèle de Markov d'ordre 1 pour représenter les vecteurs mouvement (VM) d'une séquence d'images animées [eTEF98]. Ces paramètres, estimés au codeur et supposés parfaitement transmis, peuvent être ensuite utilisés au décodeur pour tirer la statistique de source, c'est-à-dire des probabilités de transition des symboles VLC des VM.

Les méthodes de décodage avec estimation paramétrique de la source sont un premier pas vers une mise en oeuvre dans des applications pratiques. Toutefois, la méthode décrite dans l'article de A. H. Murad et al [eTEF98] présente plusieurs lacunes. Tout d'abord, une première critique d'ordre général est que l'algorithme de décodage MAP (maximum a posteriori) utilisé est extrêmement complexe car il se réalise au moyen d'un treillis de décodage qui correspond au produit de 3 treillis de décodage élémentaires. La section suivante présente le principe de l'algorithme de décodage MAP.

3.11.2.4 Le schéma de décodage MAP

On considère une source A générant des symboles appartenant à un alphabet $A = \{a_0, a_1, \dots, a_N\}$. A chaque symbole a_k est associée une probabilité d'occurrence P_k . Un codage à longueur variable associe à chaque symbole un mot de code b_k pour donner l'alphabet $B = \{b_0, b_1, \dots, b_N\}$.

Prenons une séquence d'information $W_i = (w_{i1}, w_{i2}, \dots, w_{iL})$, $w_{ik} \in A$, $k \in [1, L]$ codée en utilisant un codeur à longueur variable VLC dans une séquence $X_i = (x_{i1}, x_{i2}, \dots, x_{iL})$, $x_{ik} \in B$, $k \in [1, L]$, qui sera codé par un codeur convolutionnel dans la séquence $Y_i = (y_{i1}, y_{i2}, \dots, y_{iL})$, et soit la séquence $\tilde{Y} = (\tilde{y}_{i1}, \tilde{y}_{i2}, \dots, \tilde{y}_{iL})$ le flux reçu à la sortie du canal.

Mathématiquement, le problème du MAP [YW05] peut être exprimé par l'équation 3.18. Où \hat{Y}_i est le flux, après un codage convolutionnel, le plus probable parmi tous les flux possible $\{Y_i\}$.

$$\hat{Y}_i = \underset{Y_i}{\operatorname{arg\,max}} \frac{\operatorname{Pr}(\tilde{Y} \setminus Y_i) \operatorname{Pr}(Y_i)}{\operatorname{Pr}(\tilde{Y})} \quad (3.18)$$

Notons qu'il existe une correspondance entre Y_i et X_i , par conséquent le flux VLC le plus probable \hat{X}_i peut être calculé comme suit :

$$\hat{X}_i = \underset{X_i}{\operatorname{arg\,max}} \frac{\operatorname{Pr}(\tilde{Y} \setminus Y_i) \operatorname{Pr}(X_i)}{\operatorname{Pr}(\tilde{Y})} \quad (3.19)$$

Pour le cas du codage à longueur variable, la probabilité $\operatorname{Pr}(\tilde{Y})$ peut être calculer. Pour simplifier l'opération de décodage, Park et Miller [MP00] ont utilisé une approximation de $\operatorname{Pr}(\tilde{Y})$. Puisque la source est Markovienne, il est possible d'avoir :

$$\hat{X}_i = \underset{X_i}{\operatorname{arg\,max}} \left(\frac{Pr(\tilde{y}_l \setminus y_{il}) Pr(x_{il})}{2^{-V(\tilde{y}_k)}} \right) \times \prod_{k=2}^L \frac{Pr(\tilde{Y}_k \setminus Y_{ik}) Pr(x_{ik} \setminus x_{i(k-1)})}{2^{-V(\tilde{y}_k)}} \quad (3.20)$$

Où $v(\tilde{y}_k)$ est le nombre de bits utilisés pour coder \hat{y}_k , $Pr(x_{ij})$ est la probabilité que le mot de code x_{ij} soit transmis en premier. $Pr(x_{ik} \setminus x_{i(k-1)})$ est la probabilité que le mot de code x_{il} soit transmis immédiatement après le mot de code $x_{i(k-1)}$. L'équation 3.20 peut être dérivée de la même manière comme suit :

$$\hat{X}_i = \underset{X_i}{\operatorname{arg\,max}} [(\log Pr(\tilde{y}_l \setminus y_{il}) + \log Pr(x_{il}) + V(\tilde{y}_1) \log 2) + \sum_{k=2}^L (\log Pr(\tilde{Y}_k \setminus Y_{ik}) + \log Pr(x_{ik} \setminus x_{i(k-1)}) + V(\tilde{y}_k) \log 2)] \quad (3.21)$$

Cette formule est similaire à celle de la métrique du chemin de décodage convolusionnel. Ainsi, ce décodeur peut être utilisé comme étant un décodeur convolusionnel, il représente le célèbre décodeur de viterbi, qui est utilisé comme une métrique de chemin. Cette métrique est basée sur les trois termes $Pr(\tilde{Y}_k \setminus Y_{ik})$, $Pr(x_{ik} \setminus x_{i(k-1)})$ et $V(\tilde{y}_k)$. Le premier terme dépend strictement de la connaissance des paramètres du canal de transmission. Le deuxième terme dépend seulement des statistiques de la séquence de la source. Donc, la connaissance des statistiques du canal et de la source sont nécessaires pour implémenter cette métrique.

La technique de DCSC¹³ a fait l'objet de quelques articles, par exemple dans [MJ02], Son idée est d'améliorer le décodage du code canal par l'utilisation au niveau bit d'une information source a priori calculée à partir des probabilités symboles du VLC. Cela est possible en établissant une bijection entre l'arbre du VLC et le treillis du code correcteur d'erreur.

Les auteurs de [MJ02] ont utilisé un décodage dur du VLC, après le décodage a priori de la source, ce schéma de transmission est intitulé TC-DD¹⁴. Mais ce décodage du VLC peut aussi être souple. En effet, il est possible d'utiliser soit le treillis bit de Balakirsky [Bal97], cette solution est appelée TC-DSTB¹⁵; soit utiliser le treillis symbole présenté par Bauer et Hagenauer dans [eJH00], cette dernière approche est nommée TC-DSTS¹⁶.

Le principe de ces trois techniques est rappelé à la figure 3.11. Dans les deux derniers schémas de décodage a priori sur la source est utilisé une première fois au niveau bit dans le décodage du turbo-code, et une seconde fois sous forme binaire ou symbole, selon le treillis, lors du décodage souple de source.

Pour les méthodes utilisant un décodage dur du VLC ou un décodage souple avec un treillis bit, seul le nombre de bits par paquet est utilisé. Par contre les méthodes où le

¹³Décodage Conjoint Source Canal

¹⁴Turbo-Code Décodage Dur

¹⁵Turbo-Code Décodage Souple Treillis Binaire

¹⁶Turbo-Code Décodage Souple Treillis Symbole

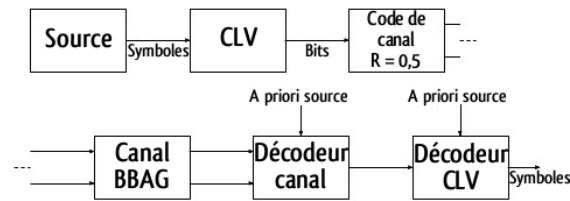


FIG. 3.11 – Turbo-décodage avec information à priori sur la source

VLC est représenté par un treillis symbole nécessitant le nombre de bits et le nombre de symboles par paquet.

3.12 Conclusion

Ce chapitre a présenté plusieurs type de FEC : les FEC classiques utilisant des codes correcteurs d'erreurs et les FEC perceptuelles ou FECp. Les FECp souffre de plusieurs limitations. Ainsi, ce que l'on récupère grâce aux données redondantes n'est jamais de qualité égale à l'original, contrairement aux FEC à codes correcteurs. Ces derniers induisent, contrairement aux FEC perceptuelles, un temps de latence à l'émetteur.

Si on oublie le temps de codage et de décodage (des algorithmes de codage très rapides existaient pour des codes de Reed-Solomon), il persiste tout de même un temps de latence incompressible puisque pour pouvoir commencer à coder, le codeur de canal devra attendre d'avoir reçu de la part du codeur de source suffisamment de données. Toutefois, les codes systématiques apportent une solution à ce problème. La source peut commencer à transmettre les données utiles et en conserver une copie en mémoire. Le codeur de canal ira chercher cette copie pour générer les données redondantes. Le même problème se posera à la réception puisque en cas de pertes le décodeur devra attendre de recevoir tous les données pour pouvoir commencer à décoder.

Dans la suite de ce document nous aurons donc à l'esprit d'essayer d'assurer une protection inégale du flux vidéo par le FEC utilisant des données déjà pré-empaquetées sous forme d'unités NAL .

Chapitre 4

Codage conjoint source canal utilisant la UEP et les codes RS

4.1 Introduction

Dans le premier chapitre de ce document nous avons présenté les mécanismes de résilience de la norme H.264/AVC ainsi que le masquage des erreurs dans un environnement mobile. Le troisième chapitre montre plusieurs techniques de FEC permettant la protection du flux vidéo et la correction des erreurs de transmission au niveau récepteur. La technique la plus adaptée au contexte des pertes est celle qui utilise les codes de type Reed-Solomon pour la génération de la redondance. Nous rappelons que cette méthode offre de hauts niveaux de correction d'erreurs grâce essentiellement à sa capacité à localiser les erreurs. De plus, des algorithmes de codage/décodage hautement optimisés permettent de l'envisager dans un contexte temps réel.

Dans ce chapitre, nous allons proposer une technique de protection inégale d'un flux H.264 en utilisant les codes RS. Notre proposition est basée sur l'utilisation de la scalabilité de la vidéo, compressée en norme H.264, pour définir un algorithme d'affectation de rendement de codage aux différents paquets du flux vidéo. A l'aide d'un simulateur construit par nos soins, une évaluation des performances de notre approche sera présentée dans la section 4.3.

4.2 Diagramme en bloc du système proposé

Codage de source et codage de canal sont deux fonctions essentielles dans tous les systèmes de communication. Le codage de source élimine au maximum la redondance de la source tandis que le codage de canal ajoute de la redondance mais de façon contrôlée. La plupart des systèmes existants optimisent ces deux blocs indépendamment. Cette pratique trouve sa justification dans le théorème de Shannon. Cependant, à cause des limitations pratiques telles que la complexité du codeur/décodeur ou du délai. Ainsi, il peut être avantageux d'optimiser conjointement le codage/décodage de source et le codage/décodage de canal.

Cette optimisation conjointe existe dans la littérature sous l'appellation "codage conjoint source canal (JSCC)", son idée est d'allouer conjointement le débit de la source et du

canal et faire coopérer les deux couches physique et application. Le but de cette technique est de donner une meilleure performance et réduire le délai et la complexité du théorème de Shannon.

Dans l'architecture classique du système JSCC pour la transmission vidéo présenté par la figure 3.6 du chapitre précédent. Le retour de l'information est utilisé pour garder un débit constant par rapport à la sortie de la source en ajustant d'une façon dynamique le paramètre de quantification Q . Mais cette technique nécessite une puissance de calcul excessive, ce qui la rend inadptée pour une vidéo conférence par exemple. De plus, changer le paramètre de quantification lors de l'encodage vidéo peut dégrader la qualité visuelle.

La Figure 4.1 montre la méthode [Kar09] que nous avons utilisé pour coder le canal tout en gardant Q constant, cela nous permet de réduire la complexité du traitement tout en améliorant la qualité perçue de la vidéo. Un débit constant est obtenu à la sortie de l'encodeur de canal en utilisant un rendement de codage variable. L'algorithme d'affectation (Rendement - NALUs) est décrit ci-après.

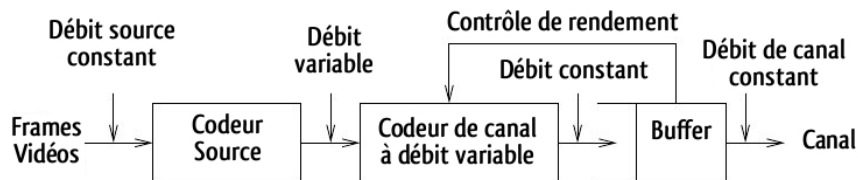


FIG. 4.1 – L'architecture du système *JSCC* utilisé

Dans ce chapitre, nous allons présenter un système de transmission numérique de la vidéo de notre conception. Notre algorithme, présenté ci-dessous, décrit une source binaire qui fournit des symboles groupés sous forme de slices. L'algorithme affecte un rendement à chaque partie de slice selon son type, puis fait un codage de canal. La transmission est simulée par l'ajout d'un bruit blanc gaussien au message codé.

L'algorithme simule également un décodeur qui effectue l'opération inverse faite par le codeur de canal pour tenter de corriger le flux. Finalement, l'algorithme quantifie la performance globale et nous comparons cette performance avec d'autres systèmes qui serviront de référence. Le diagramme en bloc de notre système est présenté sur la Figure 4.2.

Dans ce qui suit une explication détaillée de chaque bloc du système est présentée.

4.2.1 Le codage de source proposé

La première étape de notre travail consiste à compresser la séquence vidéo de test en utilisant une norme existante parmi celles introduites dans le premier chapitre. Un fichier de taille inférieur est obtenu. Le choix s'est posé sur H.264 car il représente le standard le plus récent et offre plusieurs outils de codage ainsi que des mécanismes de résilience contre la propagation des erreurs sur un lien sans fil.

La norme H.264 spécifie une couche vidéo (VCL) et une couche d'abstraction réseau (NAL) [E.G03]. Une image vidéo est représenté par une ou plusieurs slices qui sont codées

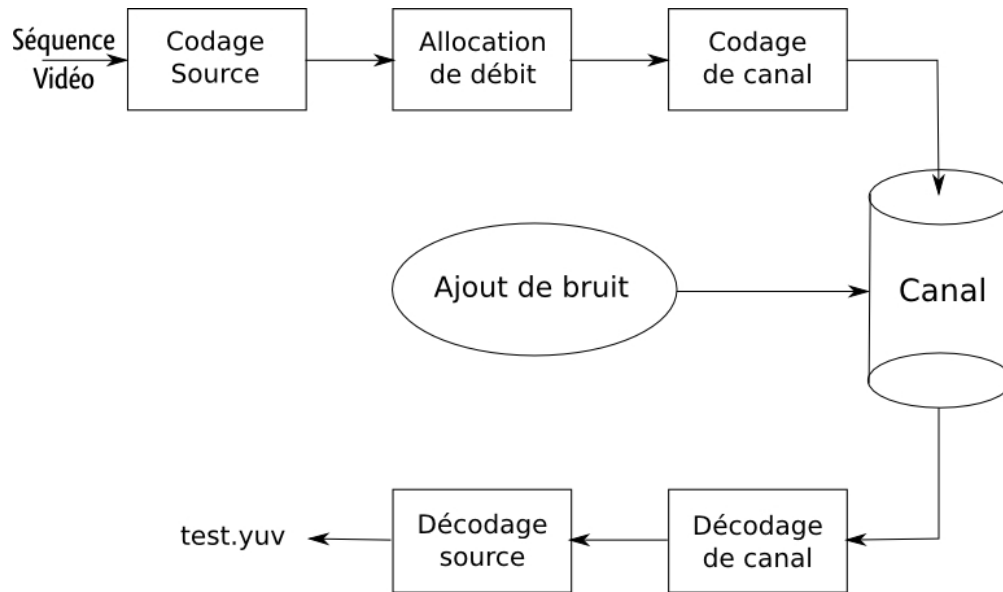


FIG. 4.2 – Diagramme en bloc du système proposé

d'une façon indépendante au niveau VCL, ces slices codées sont encapsulées dans des paquets appelés unités nal (NALUs) au niveau de la couche NAL

La norme définit deux types de NALUs : VCL et non VCL, les unités non VCL représentent les paramètres de décodage tel que l'ensemble de paramètres de séquences et l'ensemble de paramètre d'images. Une valeur numérique entière est associée à chaque NALU indiquant son type.

Afin d'améliorer les performances de l'algorithme, l'ensemble des paramètres de séquence est transmis à travers un canal dédié et sans erreurs car son intégrité est une contrainte critique dans la phase de décodage. Le reste des unités non VCL peuvent être transmis sans codage de canal car transportant des informations d'amélioration qui ne sont pas nécessaires pour le décodage de la source.

Les unités de type VCL transportent les données de la séquence vidéo elle même; ce qui est nécessaire pour le décodage. Pour cela un rendement de codage de canal est attribué à chaque NALUs par l'algorithme qui sera décrit par la suite.

Les outils de codage de ce type d'unité sont organisés en trois sous-ensembles appelés profiles (voir premier chapitre). Un profile peut être activé selon les besoins de l'application dans laquelle la vidéo sera utilisé. La subdivision de données (DP) [T.S04], par laquelle nous somme intéressés, est disponible dans le troisième profil. Pour cela nous avons activé ce dernier pour encoder nos séquences de tests.

Nous supposons que le profile étendu est utilisé et la subdivision de données activée. Grâce à cela, les slices sont codées en trois partitions A, B et C. La partition A contient des données qui sont obligatoires pour le décodage de la source, cela comprend les vecteurs de mouvement, les macroblobs et les paramètres de quantification. La partition B contient les coefficients des MBs I et la partition C les coefficient des MBs P et B.

Il est intéressant de noter que la partition B est considérée comme moins importante que la partition A et plus importante que la partition C. Pour améliorer les performances, la dissimulation des erreurs (EC) [S.K06b] au niveau décodeur peut être pris en compte dans l'allocation de débit de canal mais il faut prendre en considération que l'EC ne peut être accompli que si la partition A est reçue correctement. Notons que la dissimulation des partitions B et C est plus efficace pour de petites tailles de ces dernières, cependant, il est difficile de récupérer une longue partition [Kar09].

4.2.2 Allocation de débit de canal

L'allocation de débit est l'étape qui suit le codage de la source, elle permet d'allouer la bande passante aux différents paquets de la séquence vidéo sans dépasser la capacité du canal.

Soit $R_s(n)$ le débit de la source en *bit/frame*, généré par l'encodage de l'image F_n à l'instant n et soit N le nombre de slices de l'image F_n . Puisque le *DP* est activé, chaque slice est codée en 3 partition A, B et C. Chaque partition est encapsulée au niveau de la couche NAL dans une unité nal (NALU) de type 2, 3 ou 4. Les NALUs de type 2 sont utilisées pour les partitions A, les NALUs de type 3 et 4 pour les partitions B et C respectivement. Il en résulte que l'image F_n est codé en une séquence de $3N$ NALUs.

Soit R l'ensemble des débits de canal et R_c la capacité du canal en *bit/frame*. Dans le cas de l'utilisation de la FEC et la EEP pour les $3N$ NALUs avec une allocation du débit r à ces dernières, la contrainte d'adaptation de débit peut être satisfaite si :

$$\frac{R_s(n)}{r} \leq R_c \quad (4.1)$$

Cette inégalité ne prend pas en considération l'importance relatives des NALUs. Soit U_i ($1 \leq i \leq 3N$) la i^{eme} NALU et S_i sa taille, afin d'améliorer l'adaptativité de l'approche, nous utilisons la UEP à la place de la EEP et allouons un débit $r_i \in R$ à chaque U_i en fonction de son degré d'importance dans la phase de décodage source. La contrainte varie est devient :

$$\sum_{i=1}^{3N} \frac{S_i}{r_i} \leq R_c \quad (4.2)$$

Notre objectif est de réduire la distorsion du signal décodé en respectant la contrainte imposée.

4.2.2.1 Algorithme d'affectation de rendement

Nous sommes dans un schéma de classification obtenu par le mécanisme de Data-Partitioning tout en prenant en compte le type et la taille des partitions comme étant des critères supplémentaires de classification [Kar09].

Pour chaque image, le nombre total de NALUs est réparti en deux ensembles, le premier contient les NALUs de type 2 et le second les NALUs de type 3 et 4. La Figure 4.3 montre un schéma explicatif de l'algorithme pour une image de 9 unités NAL, une partie du débit binaire de canal est allouée au premier ensemble, le reste est alloué au second.

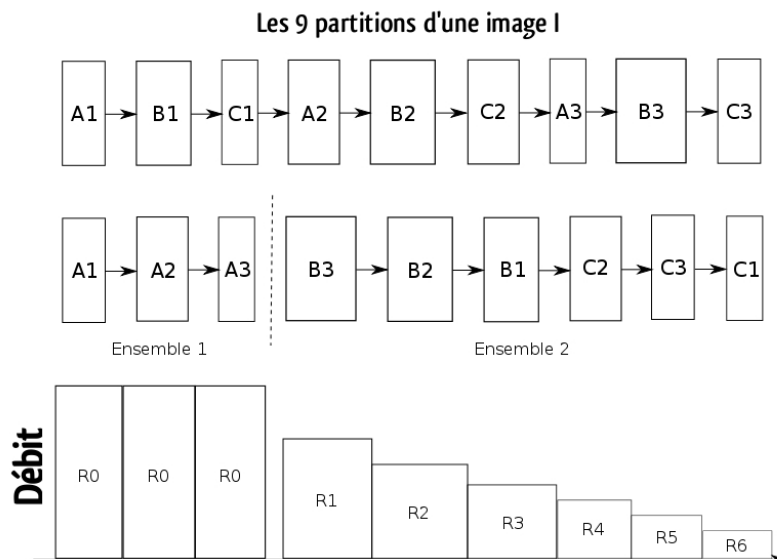


FIG. 4.3 – Classification des 9 NALUs d'une image

L'expérience montre que la taille des NALUs de type 2 ne varie pas d'une image à l'autre dans une séquence vidéo codée. Pour chaque image, le rapport entre la taille du bloc des NALUs de type 2 sur le total des blocs est relativement faible comme le montre la figure 4.4. Cette propriété nous donne la possibilité d'affecter le plus petit rendement à ce type de NALU pour assurer une meilleure protection à ces dernières tout en minimisant la consommation de la quantité de bits disponible.

Pour les NALUs du second ensemble, un algorithme d'allocation de débit est utilisé pour chaque trame. Ce dernier est basé sur les contraintes suivante :

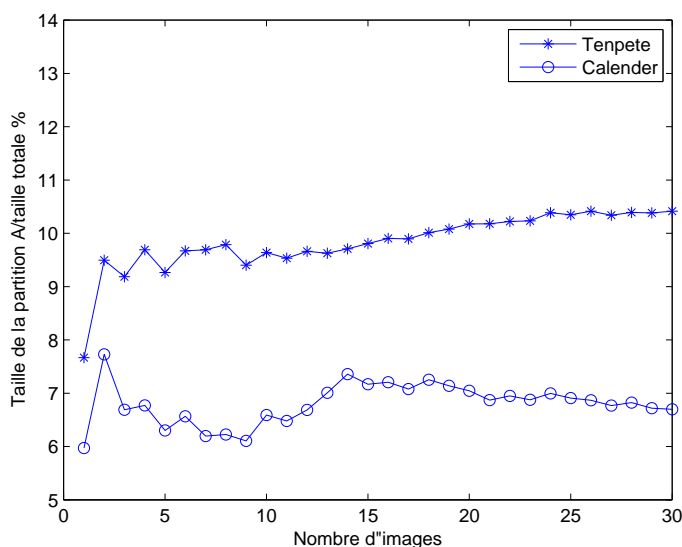


FIG. 4.4 – La taille du bloc de NALUs de type 2 par rapport à la somme des blocs

- Les NALUs de type 2 sont plus importantes que celles de type 3 et 4.
- Les NALUs de type 3 sont plus importantes que celles de type 4.
- Le degré de protection d'une NALU de type 3 ou 4 est directement proportionnel à sa taille.

En respectant les contraintes citées, l'allocation de débit pour les NALUs de type VCL est faite selon les étapes de l'algorithme 4.1.

Algorithme 4.1 Algorithme d'affectation de rendement

```
def Calculer_Vect(Image,Rc) :
```

```
  S3 = [ ]
```

```
  S4 = [ ]
```

```
  for i in Image :
```

```
    if (Type(i)==3) :
```

```
      Ordonner(S3,i)
```

```
    if (Type(i)==4) :
```

```
      Ordonner(S4,i)
```

```
  S = S3 + S4
```

```
  C += [ ]
```

```
  rate = (r31, ..., r3N, r41, ..., r4N) //r3i ≤ r3j et r4i ≤ r4j si i < j.
```

```
  for i in range(len(S)) :
```

```
    C += [S[i],rate[i]]
```

```
  D = 0
```

```
  for i in C :
```

```
    D += i[0]/i[1] //D =  $\sum_{i=1}^{3N} \frac{S_i}{r_i}$ 
```

```
  if D > Rc :
```

```
    i = 0
```

```
    while D ≥ Rc :
```

```
      r [2N - i] = r [2N - i] - 1/m
```

```
      i = (i + 1) mod 2N
```

```
      D =  $\sum_{i=1}^{3N} \frac{S_i}{r_i}$ 
```

```
  else :
```

```
    i = 1
```

```
    while D < Rc :
```

```
      r [i] = r [i] - 1/m
```

```
      i = (i + 1) mod 2N
```

```
//m est un entier qui sera décrit dans ce qui suit.
```

Nous nous plaçons donc dans le cadre d'une transmission nécessitant une protection inégale contre les erreurs où la source est subdivisée en plusieurs classes de sensibilité de protection qui doit être calculée avant le codage de canal.

4.2.3 Le codage de canal proposé

Dans le but de minimiser l'impact des erreurs de transmission sur la reconstruction de la vidéo, un codage de canal adaptatif à la structure de la source émise est nécessaire. Cette étape consiste à disposer d'une fonction de codage f_i , injective mais non surjective, qui transforme un message d'entrée m en $f_i(m)$: un message plus long qui sera transmis à la place de m .

Le canal peut brouiller le signal, c'est-à-dire que certains bits de $f_i(m)$ peuvent être changés. Avant de coder un paquet il faut d'abord le découper en blocs de même taille ; en ajoutant éventuellement des caractères de remplissage. On supposera que le canal n'efface pas de symbole, donc qu'un bit ne peut pas être supprimé.

Pour pouvoir garantir cette protection, nous avons pris en compte les quatre critères cités dans le chapitre précédent nécessaires à l'utilisation des codes RS comme codage de canal. Ils constituent une classe de codes intéressante car ils sont optimaux, c'est-à-dire qu'ils nécessitent le nombre minimum de redondance ($n - k$) pour obtenir une distance minimale donnée, ($n - k + 1$). De plus, il existe des implémentations de codage et de décodage rapides. Les deux opérations de codage et de décodage des codes RS ont été présentées en détail dans le chapitre précédent.

Les codes RS sont des codes détecteurs correcteurs d'erreur pouvant corriger les erreurs indépendantes, les symboles erronés contigus dans la mesure où le nombre d'erreurs ne dépasse pas la capacité de correction. Un code RS (n, k) est entièrement défini par le polynôme générateur $g(x)$ et le polynôme irréductible et primitif $p(x)$. Le corps de Galoi (2^d) contient 2^d éléments et α est une racine de $p(x)$. Le polynôme générateur $g(x)$ caractérise entièrement les propriétés de ce code en matière de détection et correction. La taille des symboles est de d . n est variable, il est calculé en fonction de type du paquet à coder, cela se fait par l'algorithme d'affectation de rendement déjà présenté où $m = k$ et le rapport k/n représente le rendement du code. Si l'on code des mots de longueur n , et la distance minimale est d , on appelle alors :

- Taux d'information (rendement), le rapport : k/n .
- Taux de correction, le rapport : d/n .
- On dira alors qu'un code RS est alors de type $RS(n, k, d)$ avec $d = (n - k)/2$.
Afin de décoder correctement une vidéo

Algorithme 4.2 Algorithme de codage de canal

```
def Codage_RS_Image(Image,Rate) :
    Image_Codée=[ ]
    i = 0
    For Nal in Image :
        Image_Codée+= [Coder_Nal(Nal,Rate[i])]
        i+ = 1

    retourner Image_Codée
```

4.2.4 Ajout de bruit

La phase de transmission suit la phase de codage. Cette étape est simulé par l'ajout d'un bruit au message codé en utilisant une distribution gaussienne. Le bruit blanc gaussien constitue un signal de test intéressant à partir duquel un critère de linéarité peut être défini : le NPR (Noise Power Ratio) E_b/N_0 ; E_b = énergie par bit, N_0 =densité spectrale du bruit. En pratique en téléphonie mobile, E_b/N_0 limite est considéré $\simeq 6dB$.

Avec les fonctions décrites ci-dessous, on peut générer un bruit gaussien, il nous faut une source gaussienne. Pour information, les fonctions random utilisé par python et même par C ou pascal sont pseudo-aléatoires (uniforme), ce n'est donc théoriquement pas du bruit blanc gaussien.

Algorithme 4.3 Algorithme de génération de bruit gaussien

```
def gngauss(moy, sigma) :
    PI = 3.14
    Noisy=Addnoise_flux(nals_Coder,0.002)
    u = random(1000)/1000.0
    r = sigma *  $\sqrt{(2.0 * \log(1.0/(1.0 - u)))}$ 
    u = random(1000)/1000.0
    gn = (moy + r * cos(2 * PI * u))

    retourner gn
```

Considérons le paquet vidéo comme étant un signal binaire de N points $X = \{x_1, x_2, \dots, x_N\}$ tel que $x_n = 0$ ou $x_n = 1$. Avant d'ajouter le bruit au signal binaire il faut d'abord le convertir en signal digital pour utiliser un décodage hard au niveau récepteur. Cela consiste à remplacer chaque bit "1" par un entier "+1" et chaque bit "0" par "-1". Le signal obtenu contient des "+1" et des "-1". Pour cela, un signal gaussien b de moyenne nulle est généré et ajouté au signal binaire X (voir algorithme 4.4).

Algorithme 4.4 Algorithme d'ajout de bruit

```
def Ajout_bruit(EbN0,Image) :
    moy = 0
    es = 1
    Image_bruitée=[ ]
    sn =  $10^{EbN0/10}$ 
    sigma =  $\sqrt{es/(2 * sn)}$ 

    for bit in Image :
        Image_bruitée+= [1 - 2 * bit + gngauss(Moy, sigma)]

    return Image_bruitée
```

La figure 4.5 représente la variation du TEB (Taux d'Erreur Binaire) selon la valeur du rapport E_b/N_0 en utilisant les deux fonctions déjà décrites. Notons que Le taux d'erreurs binaire (TEB) d'un message est le rapport du nombre de bits erronés par le nombre de bits du message.

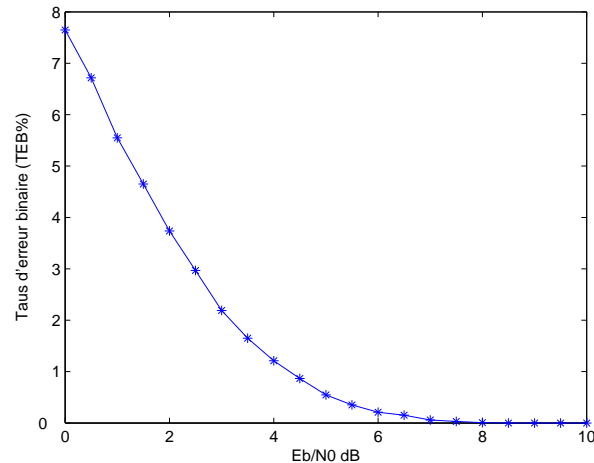


FIG. 4.5 – Variations du TEB selon la valeur du Eb/N0

4.2.5 Décodage de canal

A la réception de la séquence vidéo codée et bruitée, le décodeur doit être capable de récupérer au maximum les paquets perdus pour pouvoir décoder la séquence reçue.

Le message binaire issu de la source a été transformé en une suite d'entiers. Avant le décodage de canal, nous devons extraire le message binaire du numérique reçu, après l'ajout du bruit. Pour cela nous avons utilisé ce qu'on appelle un décodage hard pour régénérer les bits du message, son principe consiste à remplacer les entiers positifs par des bits à "1" et les entiers négatifs par des "0". Le décodage hard se fait au niveau de la couche physique du récepteur.

Le décodage de canal consiste à appliquer une fonction u ¹ sur le message binaire extrait $f_i(m)$, m_r , la notation devient alors $u(m_r)$. La fonction renvoie le message initial m ou dans certains cas, une réponse d'impossibilité de décodage si le nombre d'erreurs est trop important, c-à-d, si le nombre d'erreurs est supérieur à la capacité de correction du codec $d = 2t$. Dans ce cas, l'opération de décodage *RS* n'est pas exécutée et le paquet non corrigible sera rejeté. Par contre, si le nombre d'erreurs est acceptable, l'algorithme de décodage *RS* est activé et le message original est retrouvé.

L'algorithme 4.5 décrit le processus de décodage. Initialement, et après l'ajout des deux unités de paramètres de séquence et d'image reçus sans erreurs à travers le canal dédié, le codec essaye de décoder les trois premières unités NAL qui constituent la première image de type IDR en calculant le nombre d'erreurs dans le mot reçu (la fonction u). S'il n'arrive pas à faire le décodage, l'unité non corrigible sera rejetée.

La seconde partie de l'algorithme est une boucle qui sépare les slices du flux reçu et décode de la même manière les 9 unités de chaque image en utilisant le vecteur de rendement utilisé dans la phase de codage. Le résultat de ce processus est une suite d'unités NAL sans erreurs qui sera utilisée dans l'étape suivante, décompression H.264, pour avoir une vidéo décodée.

¹Le calcul du polynôme de syndrome

Algorithme 4.5 Algorithme de décodage de canal

```

def Décodage_RS_Images(Image, rate) :
    Image_décodée=[ ]
    i = 0
    For Nal in Image :
        Image_décodée+=[Décoder_Nal(Nal,Rate[i])]
        i+ = 1

    retourner Image_décodée

```

4.2.6 Le décodage de source

Après le décodage de canal et l'élimination des paquets non corrigible, le fichier résultat sera décompressé pour reconstruire la séquence vidéo. Cette étape nous permet de calculer le PSNR de la composante de luminance de la vidéo reçu et de comparer la qualité de cette dernière avec la séquence de référence. Le décodeur source permet alors l'interprétation des données et la reconstruction de la vidéo émise. Le résultat de cette étape est un fichier vidéo, le PSNR de chaque image de la séquence ainsi que le PSNR de toute la séquence vidéo.

4.3 Test de la méthode proposée

4.3.1 Environnement de simulation

Nous avons développé une application qui reprend les idées proposées précédemment. Cette application a été écrite en Python sous le système d'exploitation Linux Ubuntu 9.04. Le choix du langage est dû principalement aux énormes qualités et le succès indiscutable qui le qualifient.

Traditionnellement, ce genre de méthodes sont développés sous C ou C++ voir même Java. Mais il est indéniable que travailler sous ces langages demandent un temps et un effort intellectuel importants. Python base sa philosophie sur la simplicité d'écriture de code. Écrire un code sous Python apporte un gain de temps d'un facteur oscillant entre 2 et 10. Donc, l'étudiant, plutôt que de réfléchir à la manière d'implémenter une solution va plutôt travailler à améliorer celle-ci sans contraintes de programmation.

Nous avons opté pour le système libre Linux Ubuntu 9.04 pour s'essayer à l'univers de développement libre et tester les différentes alternatives qu'il propose par rapport à d'autre solution payante.

Python étant Orienté Objet, nous avons trouvé une grande facilité à développer notre application. Le tableau 4.1 résume l'environnement sur lequel a été effectuer la simulation et à partir duquel nous avons obtenus nos résultats lors de l'expérimentation.

L'application considérée a été modularisée au maximum comme l'attestent les algorithmes précédents. Chaque étape du Codage-décodage du système H.264 protégé par le système RS a été implémentée de manière indépendante. Le pseudo algorithme 4.6 résume

Ordinateur	HP Laptop 510
Processeur	Intel Inside Centrino Mobile Technology 2.0 GHz
RAM	1 Go DDR
Cache	1 Mo
DD	120 Go
OS	Linux Ubuntu 9.04

TAB. 4.1 – Environnement de simulation

les grandes étapes et l'allure générale du programme principal.

Algorithme 4.6 Algorithme général de l'application

```

Rc = 192000
Ra = 2
Seq_Image = Codage_h264(flux)
Seq_décodé=[ ]
//flux[0] : ensemble de paramètres de séquence
//flux[1] : ensemble de paramètres d'image
Image_codée = Codage_RS_Image(Image,Ra)
Image_bruitée = Ajout_bruit(EbN0,Image_codée)
Image_décodée = Décodage_RS_Image(Image_bruitée,Ra)
Seq_décodée += [Image_décodée]
for Image in Seq_image[5 : len(flux)] :
    Rate = Calculer_vect(Image,Rc)
    Image_codée = Codage_RS_Image(Image,Rate)
    Image_bruitée = Ajout_bruit(Eb/N0,Image_codée)
    Image_décodée = Décodage_RS_Image(Image_bruitée,Rate)
    Seq_décodée += [Image_décodée]

Flux_res = Décodage_h264(Seq_décodée)
retourner Flux_res

```

4.3.1.1 Description du CoDec RS utilisé

Après le téléchargement et l'installation du module Reedsolomon, nous devons l'importer sous python et créer l'objet CoDec pour pouvoir l'utiliser. L'objet crée contient les deux fonction de codage et de décodage, il a comme paramètres la taille du message source k et celle du mot de code résultat n . l'importation du module et la création de l'objet CoDec peuvent être lancé en utilisant les deux commandes suivantes :

```

>>>import reedsolomon
>>>C=reedsolomon.Codec(n, k)

```

La construction d'un mot de code comportant n symboles est effectuée à partir des k symboles du message source $U = 'u_1u_2..u'_k$, appelé généralement message d'information, et de r symboles de redondance où $r = n - k$. La méthode de codage consiste à laisser inchangés les k symboles d'information et à les reporter tels quels dans le mot de code en ajoutant les r symboles de redondance $'a_1a_2..a'_r$. Cela veut dire que ce code est de type systématique. Ces symboles dits de contrôle sont calculés à partir des symboles d'information du bloc auquel ils appartiennent, c'est à dire que les n symboles du mot de code sortant dépend seulement des k symboles en entrée correspondant, il est donc sans mémoire. Soit l'exemple suivant :

```
>>>import reedsolomon
>>>C=reedsolomon.Codec(8, 5)
>>>Encoded=C.encode('video')
>>>Encoded
'video\x96\n\xaf'
```

La suite codée contient le mot originale suivi par les 3 caractères de contrôle. Si le mot de code ne contient pas d'erreur, le resultat de décodage est donné comme suit :

```
>>>C.decode('video\x96\n\xaf')
('video', [ ])
```

Dans ce cas le message originale est récupéré avec succès, la deuxième partie du résultat présente les corrections faites par le Codec ; dans le cas de cet exemple la liste est vide car le nombre d'erreurs est nul. Si nous ajoutons des erreurs au message codé sans dépasser la capacité de correction, le message originale peut être reconstruit :

```
>>>C.decode('vidoo\x96\n\xaf')
('video', [3])
```

Donc de Codec a pu récupérer le caractère d'index 3 en utilisant les caractères de contrôle. Si le nombre d'erreurs dépasse $r/2$, le Codec va nous afficher un message d'erreur.

```
>>>C.decode('viooo\x96\n\xaf')
reedsolomon.UncorrectableError : Too many errors or erasures in input
```

Ce Codec exige une relation entre le nombre de bits par symbole et la taille maximale du mot de code. Par exemple pour un symbole de taille 8bits, n ne doit pas dépasser 255 symboles , donc la contrainte suivante doit être respectée :

$$n \leq 2^{symsize} - 1$$

Prenons maintenant une suite de 10000 paquets de taille 100 octets chacun et nous calculons le taux de perte des paquets (TPP) après codage, ajout de bruit et décodage RS de cette suite ; k est fixé à 50. La courbe (a) de la figure 4.6 montre la variation du TPP selon la valeur de N en utilisant un $Eb/N0 = 3$ pour générer le bruit. Pour le cas de la deuxième courbe, N est considéré fixe égale à 60 et la variation du TPP est calculée

selon la valeur du rapport E_b/N_0 .

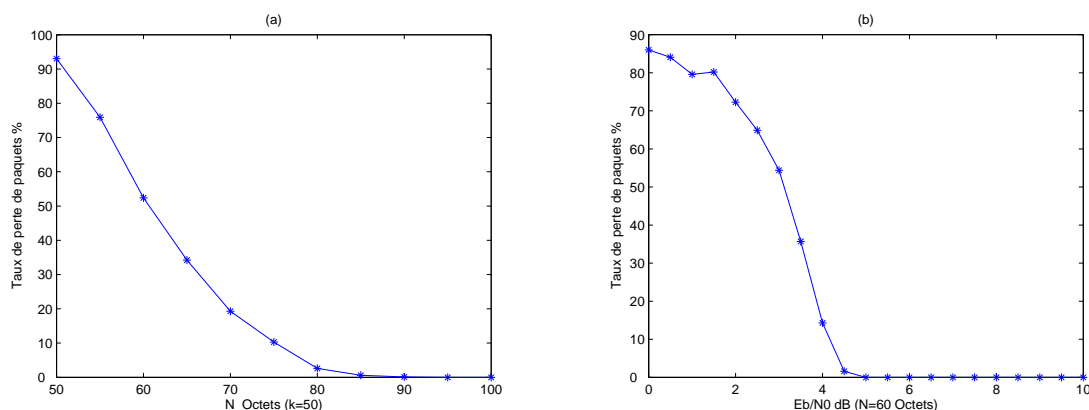


FIG. 4.6 – Variation du taux de perte de paquets utilisant les code RS

La vitesse de codage dépend du Codec choisi, le plus court sera le codec le plus rapide. Sur un ordinateur portable M 1,7 GHz pentium, le Codec peut coder plus de 11 Mo/s avec un Codec (10, 8), mais seulement 7 Mo/s avec un Codec (255, 223).

4.3.2 Résultats de simulation

Les résultats de simulation présentés dans cette section ont été obtenu pour deux séquences de test *src1_ref_625.yuv* et *src1_ref_625.yuv*² codées à l'aide du standard JM10.0 ; Les paramètres que nous avons utilisé pour coder ces deux séquences sont donnés par le tableau 4.2. Nous avons fixé le nombre de macrobloc (MB) par slice à 33 pour avoir des images codées en 3 Slices chacune, donc chaque image est codée en 99 MacroBloc, ce qui caractérise le format QCIF (Quarter Common Intermediate Format). Nous avons également choisi le profile étendu pour pouvoir utiliser la subdivision de données qui nous permet de coder chaque slice en 3 unités NAL (NALU), donc 9 NALUs par image.

Pramètre	Sa valeur
QP des images I et P	28
QP des images B	30
Nombre de MB/Slice	33
Nombre de Slice/Image	3
Codage entropique	CAVLC
Format de la vidéo	QCIF
Débit	15 Images/s
profile	Etendu

TAB. 4.2 – Les paramètres utilisés dans la phase de codage H.264

Après le codage de la séquence de test et l'obtention du fichier compressé test.264, on passe à l'étape du test de l'algorithme d'allocation de débit utilisé dans notre travail. L'exécution de ce dernier pour deux valeurs du Rc (Débit du canal en bits/image) sur la

²<http://www.vqeg.org>

24^{ème} image de la séquence src1 nous a donné les résultats présentés par le tableau 4.3. Notons que le vecteur de rendement est initialisé à [85/50, 80/50, 75/50, 70/50, 60/50, 65/50], et $m = 50$.

Type	Taille (octets)	Debit=160Kb/image	Debit=180Kb/image
A	531	50/70	50/70
B	2354	50/61	50/70
C	3062	50/52	50/57
A	569	50/70	50/70
B	1422	50/56	50/65
C	3847	50/52	50/53
A	591	50/70	50/70
B	1465	50/59	50/68
C	3937	50/54	50/63
Debit(Kb/image)		159976	179952

TAB. 4.3 – Le vecteur de rendement de la 24^{ème} image de la séquence src1

Selon les résultats obtenus, on peut dire que l'allocation de débit aux unités NAL, selon les contraintes exigées dans l'algorithme, est faite de façon efficace où les petit rendement sont affectés aux unités de haute priorité. Le vecteur moyen de rendement des partition B et C de toute la séquence vidéo ainsi que le rendement moyen de la EEP et donné par la figure 4.7.

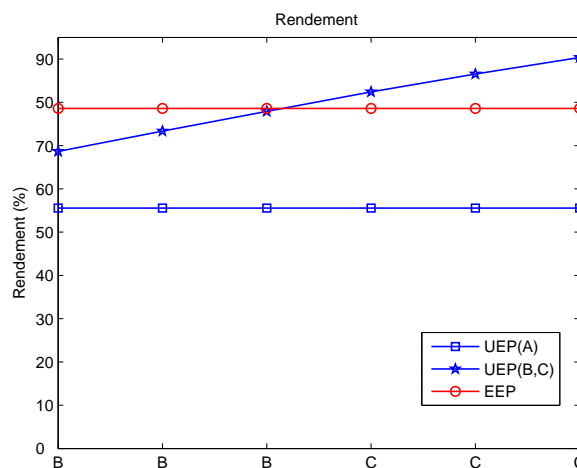


FIG. 4.7 – Le vecteur moyen de rendement

La variation du taux de perte des trois types de paquet après correction UEP est donnée par la figure 4.8.

4.3.2.1 Utilisation de la protection égale

Pour pouvoir comparer les performances de notre méthodes avec d'autres méthodes de codage, nous avons implémenté une technique de protection égale (EEP), en utilisant les codes RS, basée sur l'utilisation d'un débit constant pour toutes les images de la séquence.

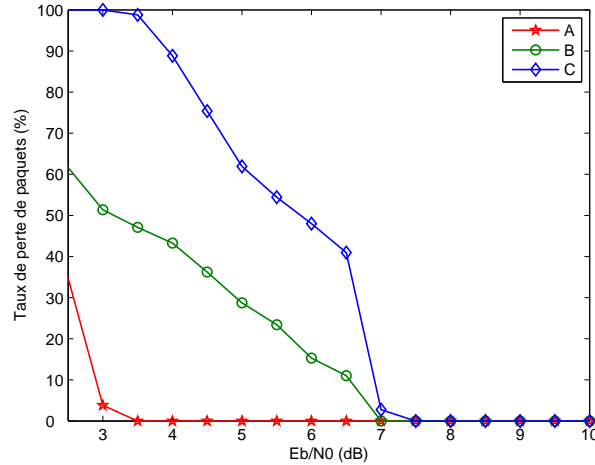


FIG. 4.8 – Variation du TPP des trois partitions

Ce dernier peut être calculé en fonction du nombre moyen de bits par image et le débit de canal R_c comme le montre l'équation 4.3.

$$r_{moy} = \frac{\frac{1}{n} \sum_n R_s(n)}{R_c} \quad (4.3)$$

Où $R_s(n)$ représente le nombre de bits de l'image n . Les deux algorithmes 4.7 et 4.8, montrent les étapes de codage et de décodage d'une séquence vidéo utilisant la protection égale.

Algorithme 4.7 Codage EEP

```

def Codage_EEP(Flux,Rc) :
    r_moy=Calculer_rate(Flux,Rc)
    Flux_Codé=[]
    Nbr_Images=int(len(Flux)/9.0)
    i = 0
    Tq i < Nbr_Images faire
        Flux_Codé+=Codage_RS(Flux[i],r_moy)
        i+= 1

    retourner Flux_Codé
  
```

Dans nos simulations, et en utilisant le nombre moyen de bits par image calculé pour les deux séquences et la valeur du R_c qui a été fixé à $192Kb/image$, le débit moyen correspondant $r_{moy} \simeq 0.7$.

4.3.2.2 Performance de la Méthode Proposée en taux d'erreur binaire

Après le test de l'efficacité de l'algorithme d'allocation de débit utilisé, nous allons quantifier les performances de notre méthode en calculant le TEB en sortie du décodeur de canal en utilisant les deux technique de protection. C'est l'un des paramètres les plus importants de cette étude, il nous permet de quantifier l'efficacité du codage.

Algorithme 4.8 Décodage EEP

```

def Décodage_EEP(Flux,Rc) :
    r_moy=Calculer_rate(Flux,Rc)
    Flux_décodé=[ ]
    Nbr_Images=int(len(Flux)/9.0)
    i = 0

    Tq i< Nbr_Images faire
        Flux_décodé+=Décodage_RS(Flux[i],r_moy)
        i+ = 1

    retourner Flux_décodé

```

Pour comparer les deux correcteurs nous avons donc pris une séquence vidéo, appliqué un codage UEP et un autre EEP sur cette dernière, ajouté le bruit aux deux séquences résultats et analysé le taux d'erreur binaire dans les deux cas ; avant et après le décodage de canal. La figure 4.9 représente l'évolution du TEB en fonction du rapport signal sur bruit avec et sans correction d'erreur. Dans cette figure, à gauche, indique le taux d'erreur binaire du système en utilisant la technique EEP et à droite celui obtenu avec un correcteur UEP .

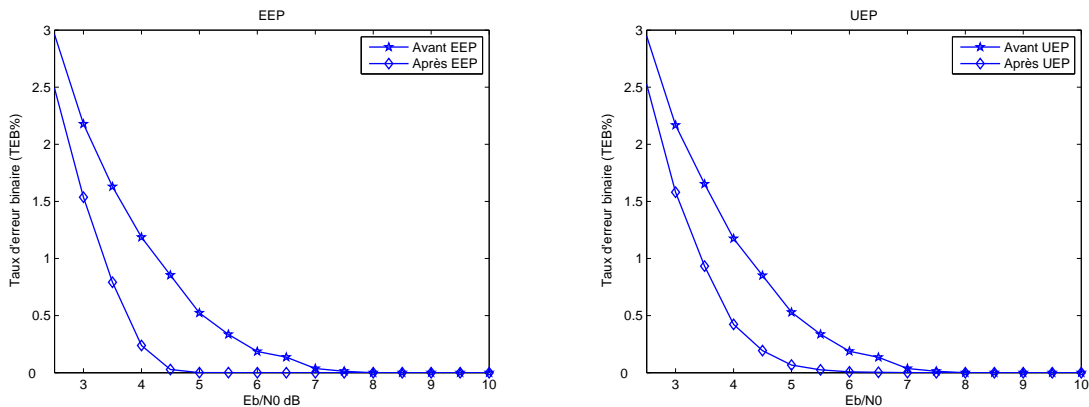


FIG. 4.9 – TEB avant et après la EEP et la UEP

Nous observons dans les deux cas que le TEB est amélioré en passant par la correction ; comme le montre la figure 4.9. En effet les erreurs commises avant le décodage de canal ne peuvent être totalement corrigées. La figure 4.10 montre que les taux obtenus dans les deux cas sont quasiment identiques, les deux correcteurs ont presque les mêmes performances avec un léger avantage à l'algorithme UEP.

Cependant il faut noter que, de point de vue complexité, le correcteur EEP possède un avantage par rapport à la UEP. Il demande moins de calcul à l'utilisation, et ses performances sont légèrement meilleures que le correcteur UEP vis-à-vis du taux d'erreur binaire tout en étant plus simple à implémenter. Par contre, son défaut majeur est qu'il ne prend pas en considération la sémantique du flux vidéo et il ne peut compenser efficacement la perte des paquets de haute priorité. Dans ce cas, le correcteur UEP peut être plus utile

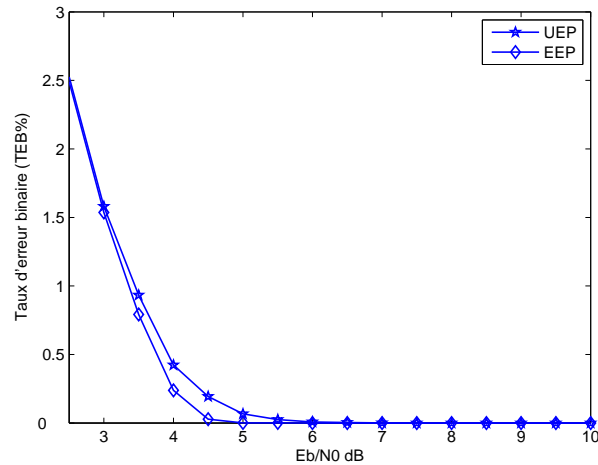


FIG. 4.10 – Comparaison du TEB de la UEP et EEP

car il offre plus d'efficacité vis-à-vis du PSNR et de la qualité visuelle, ceci fait l'objet de la suite de ce chapitre.

4.3.2.3 Performance de la Méthode Proposée en taux de perte de paquets

Intéressons nous maintenant à l'évaluation du taux de perte de paquets des trois blocs de partition. Ce dernier représente le rapport du nombre de paquets perdus par le nombre de paquet du bloc. Notons qu'un paquet est considéré comme étant perdu s'il contient au moins un bit erroné. Ce paramètre a un effet immédiat sur la qualité de transmissions multimédia (voix et vidéo) et un effet indirecte sur les applications de transfert de données qui utilisent typiquement le protocole TCP.

Les figures 4.11, 4.12, 4.13 montrent la variation du taux de perte de paquets selon le rapport E_b/N_0 des trois partitions A, B et C respectivement utilisant les deux méthode de protection UEP et EEP. 100 réalisations sont effectuées pour chaque valeur du rapport E_b/N_0 , une moyenne des taux de perte de paquet sera calculée.

Dans tous les cas observés, le taux de perte de paquets diminue quand E_b/N_0 croît. Ainsi, de manière asymptotique, nous pouvons voir que la FEC utilisée par la UEP a des capacités de protection inégale des trois types d'unités NAL.

Dans la figure 4.11, la UEP offre une meilleure protection par rapport à la deuxième technique quelque soit la valeur du rapport E_b/N_0 , ce qui est tout à fait le contraire de ce qui montre la figure 4.13 des partitions C. Les partitions B sont mieux protégées par la UEP entre 0 et 4.5dB (figure 4.12), haut delà de 4.5dB, la protection offerte par la UEP se dégrade et devient de moindre qualité par rapport à la EEP.

L'allure des deux tracés de la figure 4.12 trouve son interprétation dans le fait que la méthode UEP protège bien les grands paquets et mal les petits paquets. Cela a pour effet une meilleur récupération des paquets lorsque le canal est trop bruité avec la méthode UEP. Cependant, lorsque le canal devient moins bruité, la méthode EEP arrive par son rendement moyen à récupérer les grands paquets. D'un autre coté, les petits paquets sont

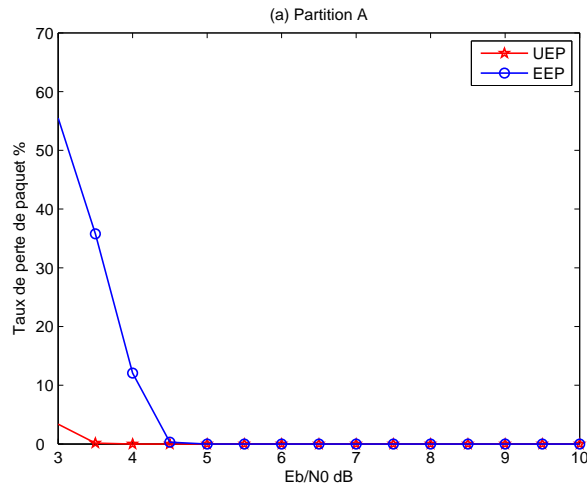


FIG. 4.11 – Le taux de perte de paquets A pour les deux méthode UEP et EEP

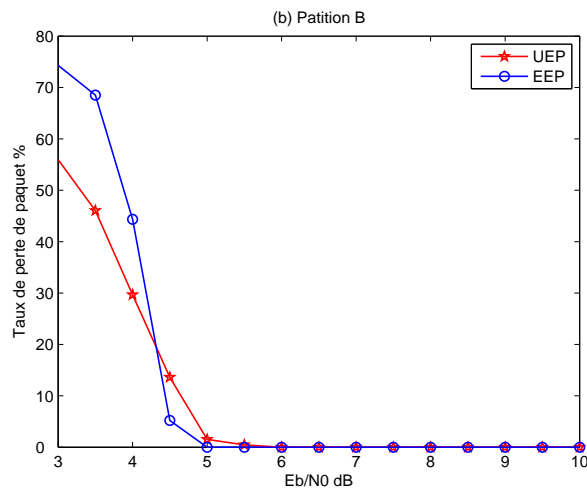


FIG. 4.12 – Le taux de perte de paquets B pour les deux méthode UEP et EEP

mieux protégés par la EEP ce qui fait qu'ils sont tous récupérés. Vu l'aspect inégale de la protection UEP, les petits paquets subissent une négligence qui entraîne leurs pertes. Cela explique pourquoi la EEP est plus efficace pour des canaux moins bruités.

Maintenant nous supposant que la dissimulation des erreurs (EC Error Concealment) est utilisée par le décodeur source pour récupérer les partitions B et C perdues de chaque slice. Cette récupération ne peut être possible si et seulement si la partition A de cette dernière est reçue sans erreurs. Pour cela nous pouvons conclure que la technique UEP proposée donne de meilleurs résultats par rapport à EEP car elle récupère la majorité des partitions A dans la phase de décodage FEC grâce à une forte protection de ces dernières.

Pour pouvoir quantifier l'efficacité de l'utilisation de l'EC, nous avons calculé "le taux de perte effectif" [Kar09] des partitions B et C. Ce dernier peut être calculé en fonction du taux de perte des partitions A et B ou C puisque il existe une relation directe entre ces deux ensembles où pour chaque slice "La réception de la partitions A \Rightarrow La récupération des partitions B et C". Le taux de perte effectif des paquets de type B et C après l'application

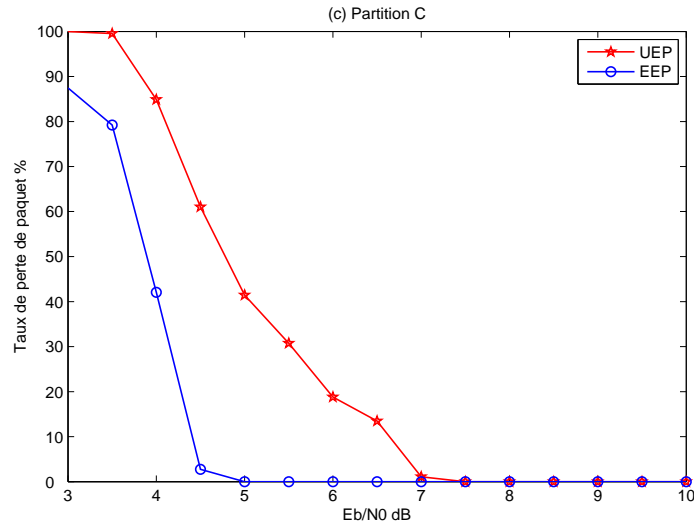


FIG. 4.13 – Le taux de perte de paquets C pour les deux méthode UEP et EEP

de l'EC peut être calculé comme suit :

$$TPP_{eff}(B/C) = TPP(A) \times TPP(B/C) \quad (4.4)$$

Où $TPP(A)$, $TPP(B/C)$ représentent le taux de perte de paquet des parties A, B ou C respectivement. La figure 4.14 montre la variation de ce coefficient, taux de perte effectif, par rapport à E_b/N_0 des deux types de partitions B et C. Il est clair que la UEP apporte une amélioration vis-à-vis le taux de perte de paquets par rapport à la UEP.

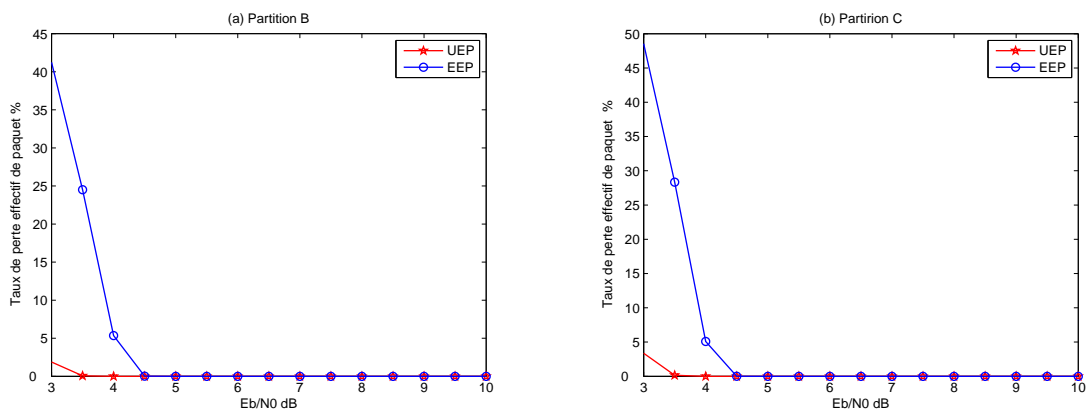


FIG. 4.14 – Le taux de perte effectif des partitions B et C

En conclusion, un taux de perte global peut être calculé pour toute la séquence en faisant une moyenne des trois TPP $TPP(A)$, $TPP_{eff}(B)$ et $TPP_{eff}(C)$ comme suit :

$$TPP_G = (TPP(A) + TPP_{eff}(B) + TPP_{eff}(C)) / 3 \quad (4.5)$$

La variation du TPP_G des deux techniques est donnée par la figure 4.15.

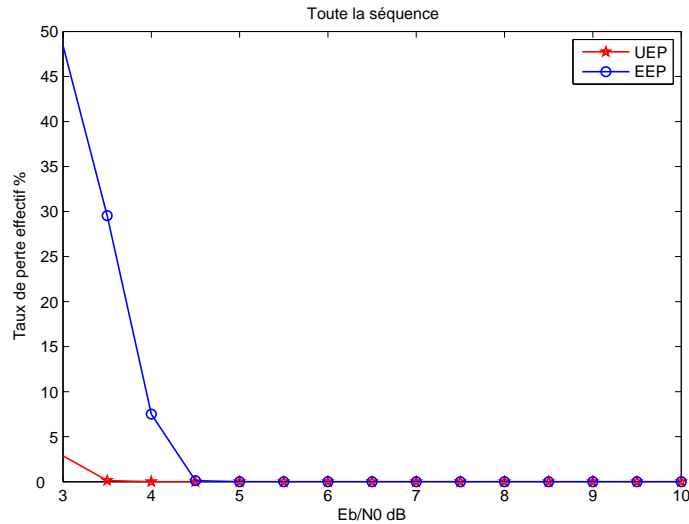


FIG. 4.15 – Variation du TPP global après l'EC pour la UEP et la EEP

4.3.2.4 PSNR et qualité visuelle

Pour voir l'impact de cette technique d'adaptation sur la qualité vidéo, nous avons calculé le PSNR des séquences vidéo reçues (après compression, codage de canal, transmission, décodage de canal et décompression) dans les deux scénarios (UEP et EEP) par rapport à la séquence originale. Les résultats sont donnés par la figure 4.16 où la variation du PSNR est donnée par rapport à la valeur du rapport E_b/N_0 . Il est intéressant de noter que le PSNR est une mesure de distorsion qui consiste à quantifier les performances des codeurs en mesurant la qualité de reconstruction de la vidéo décompressée par rapport à la vidéo originale, les valeurs typiques du PSNR pour des vidéos de bonne qualité varient entre 30 et 35db.

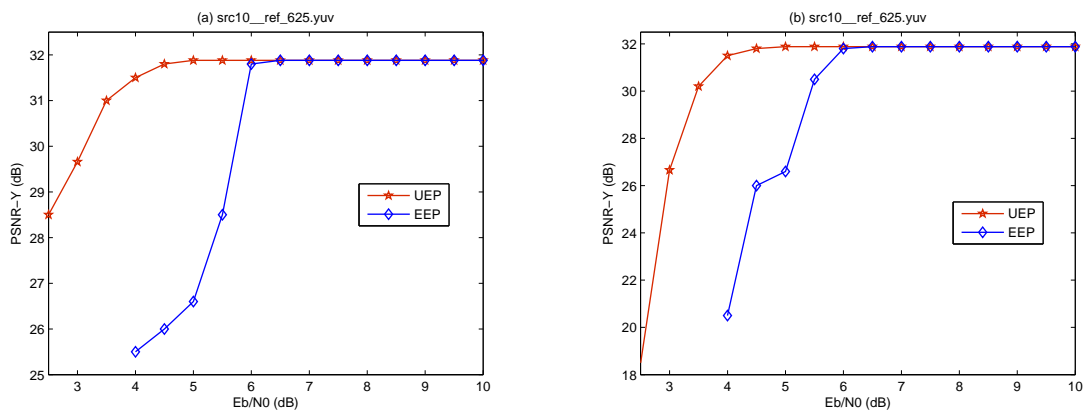


FIG. 4.16 – La variation du PSNR

Dans le premier scénario (UEP), il est clair que ce dernier permet le décodage H.264 d'un maximum de paquets perdus par rapport au second scénario puisque la FEC utilisée a permis de récupérer la majorité des partitions A ce qui rend la dissimulation des partitions B et C dans la phase de décodage H.264 plus efficace.

Pour confirmer les résultats obtenus, nous avons comparé la qualité visuelle des vidéos décodées utilisant les deux techniques UEP et EEP. La visualisation de la vidéo décodée montre une meilleure qualité pour la UEP.

4.4 Conclusion

Dans ce chapitre, nous avons proposé une technique de protection inégale d'un flux vidéo H.264. Notre proposition s'appuie sur un algorithme d'allocation de débit de canal selon la sémantique de la source dans le cadre d'un codage conjoint source canal. Cette allocation utilise l'outil de subdivision de données de la norme H.264 et prend en compte le type d'un paquet et sa taille comme contraintes de différenciation. La protection des différents paquets de la séquence vidéo est assurée par un codage de canal au niveau applicatif en utilisant les code Reed-Solomon. Cette protection est adapté aux données inégalement protégées. Notre technique a montré son efficacité par rapport à la protection égale vis-à-vis le PSNR et la qualité visuelle.

Conclusion Générale

Dans ce mémoire, nous nous sommes intéressés à la transmission par paquets de la vidéo en temps réel sur les réseaux sans fil qui sont caractérisés par des taux d'erreurs très élevés par rapport aux réseaux filaires. Pour cela, plusieurs approches sont proposées dans la littérature pour minimiser la dégradation de la qualité vidéo perçue par l'utilisateur.

Dans la première partie de ce mémoire, H.264/AVC, la dernière norme de codage vidéo qui a été adoptée par la communauté internationale, a été d'abord considérée. Une description détaillée des étapes de compression dans cette norme a été présentée. Ceci nous a permis d'introduire de nombreux concepts et outils de codage, permettant ainsi de garder ce mémoire assez consistant pour aborder sa suite.

La deuxième partie a été consacrée à l'étude des canaux de transmission. Les modèles du canal Gaussien et du canal de Rayleigh ont été considérés. Pour chaque modèle, le taux d'erreur correspondant est calculé en fonction de l'énergie par bit consacrée à l'émission. Une description succincte des fonctions qui se retrouvent dans un système de communications vidéo a été aussi présentée dans cette partie, tout en considérant l'aspect protocolaire.

Dans la troisième partie, un état de l'art des différentes techniques de transmission vidéo en temps réel sur les réseaux sans fil a été présenté. Les différents codes correcteurs d'erreurs ont été étudiés et des critères de sélection des bons codes à utiliser dans divers contextes ont été aussi exposés.

Dans la quatrième partie du mémoire une méthode robuste de transmission en temps réel de la vidéo sur les réseaux sans fil à commutation de paquets a été proposée. L'idée directrice est d'utiliser la scalabilité de la vidéo, compressée en norme H.264, pour définir un algorithme d'affectation de rendement de codage aux différents paquets du flux vidéo, et d'utiliser les codes Reed-Solomon comme codage de canal au niveau de la couche application. Ceci résulte alors en un schéma de codage conjoint source canal, basé sur une protection inégale des paquets vidéo à transmettre. L'évaluation des performances de cette technique, par rapport à une protection égale, a été ensuite effectuée. La validation de la technique proposée a été faite par simulation, montrant son efficacité.

Une perspective intéressante pour améliorer le présent travail est d'envisager une architecture inter-couche qui permettra de réduire le taux d'erreurs sur les paquets. Plus précisément, une FEC doublée au niveau des deux couches MAC et application, comme solution proactive, et une solution réactive basée sur le protocole ARQ peuvent être combinées harmonieusement. Le premier niveau de protection sera appliqué sur des paquets et le second sur des symboles. Si le nombre d'erreurs dépasse la capacité de correction

de ces deux niveaux de correction, une demande de retransmission (ARQ) est faite par une des deux couches. Donc les codes correcteurs dans cette solution auront pour rôle de minimiser la charge sur le protocole ARQ.

Appendices

Annexe A

Champs de Galoi

A.1 Groupe

Un groupe G est un ensemble d'éléments muni d'une opération binaire ".". Chaque paire d'éléments, a et b , a un seul élément c défini par l'opération binaire . dans G , $c = a.b$. Un groupe doit respecter les contraintes suivantes :

- Le groupe doit être fermé, pour n'importe quel élément a et b dans G , l'opération binaire $c = a.b$ doit toujours retourner un élément dans le même groupe G .
- La loi associative doit être applicable; $a(bc) = (ab)c$ pour tout a, b, c compris dans G .
- Il doit exister un élément e Pour tout élément a dans G tel que $a.e = e.a = a$. L'élément e est appelé l'élément neutre de G .
- Pour tout élément a dans G , il doit exister son élément inverse a' tel que $a.a' = e$. Un groupe est dit abélien si la loi de commutation est valable, $ab = ba$ pour tous les éléments de G .

A.2 Anneau

Un anneau R est un ensemble d'éléments muni de deux opérations binaires, addition et multiplication. Un anneau R est un groupe abélien selon l'opérateur d'addition. La multiplication doit respecter les contraintes suivantes :

- La multiplication doit être associative $a(bc) = (ab)c$, pour tout a,b,c dans R .
- La multiplication doit être distributive par rapport à l'addition, $a(b+c) = ab+ac$ et $(a+b)c = ac+bc$ pour tout a,b,c dans R . Un anneau est dit commutatif si sa multiplication est commutative, $ab = ba$ pour tous a,b dans R .

Un anneau est appelé domaine d'intégrité s'il possède un élément unité pour la multiplication et et qu'il n'admet pas de division par zéro.

- Il y a un élément unité pour la multiplication tel que $1a = a1 = a$ pour tout a dans R .
- Pour tout a,b en R , $ab = 0$, il y a alors la possibilité que $a = 0$ ou $b = 0$.

A.3 Corps ou champs

Un corps ou champ C est un domaine d'intégrité dans lequel tous les éléments non nuls sont inversibles. Pour un a non nul dans C , il existe un élément a^{-1} dans C tel que $aa^{-1} = 1$. Un corps possède toutes les propriétés définies précédemment.

A.4 Champs de Galoi

Les champs de Galois font partie d'une branche particulière des mathématiques qui modélise les fonctions du monde numérique. Ils sont très utilisés dans la cryptographie ainsi que pour la reconstruction des données comme on le verra dans les sections suivantes.

La dénomination "champ de Galois" provient du mathématicien français Galois qui en a découvert les propriétés fondamentales. Il y a deux types de champs, les champs finis et les champs infinis. Les champs de Galois finis sont des ensembles d'éléments fermés sur eux-mêmes. L'addition et la multiplication de deux éléments du champ donnent toujours un élément du champ fini.

A.5 Les opérations dans le champs de Galoi

Pour effectuer une addition ou ne soustraction sur un corps de Galoi, il faut voir que l'on additionne en fait deux vecteurs, et donc que l'on effectue l'addition ou la soustraction composante par composante.

Les opération d'addition et de soustraction revient à faire un ou exclusif entre deux nombres, par contre, la multiplication nécessite un élément primitif qui génère le groupe.

A.5.1 Polynômes primitifs

Ce polynôme permet de construire le champ de Galois souhaité. Tous les éléments non nuls du champ peuvent être construits en utilisant l'élément α comme racine du polynôme primitif. Chaque m a peut être plusieurs polynômes primitifs $p(x)$, mais dans le tableau 3.1, on mentionne seulement les polynômes ayant le moins d'éléments. Les polynômes primitifs pour les principaux champs de Galois sont donnés par le tableau A.1.

A.5.2 Construction d'un $GF(2^q)$

On veut construire tous les éléments du $GF(2^4)$ à partir du polynôme primitif :

$$p(x) = x^4 + x + 1$$

Comme mentionné précédemment, l'élément α est la racine du polynôme primitif, on peut en déduire :

$$p(x) = x^4 + x + 1$$

$$p(\alpha) = \alpha^4 + \alpha + 1$$

m	$P(X)$	m	$P(X)$
3	$1 + X + X^3$	14	$1 + X + X^6 + X^{10} + X^{14}$
4	$1 + X + X^4$	15	$1 + X + X^{15}$
5	$1 + X^2 + X^5$	16	$1 + X + X^3 + X^{12} + X^{16}$
6	$1 + X + X^6$	17	$1 + X^3 + X^{17}$
7	$1 + X^3 + X^7$	18	$1 + X^7 + X^{18}$
8	$1 + X^2 + X^3 + X^4 + X^8$	19	$1 + X + X^2 + X^5 + X^{19}$
9	$1 + X^4 + X^9$	20	$1 + X^3 + X^{20}$
10	$1 + X^3 + X^{10}$	21	$1 + X^2 + X^{21}$
11	$1 + X^2 + X^{11}$	22	$1 + X + X^{22}$
12	$1 + X + X^4 + X^6 + X^{12}$	23	$1 + X^5 + X^{23}$
13	$1 + X + X^3 + X^4 + X^{13}$	24	$1 + X + X^2 + X^7 + X^{24}$

TAB. A.1 – Polynômes primitifs dans $GF(2^m)^4$

$$0 = \alpha^4 + \alpha + 1$$

$$\alpha^4 = \alpha + 1$$

Maintenant, il suffit de multiplier l'élément α par α à chaque étape et réduire par rapport à $\alpha^4 = \alpha + 1$ pour obtenir le champ complet. On aura besoin de 13 multiplications pour compléter le champ.

Les éléments d'un "champ de Galois" de $GF(2^4)$ sont représentés dans le tableau A.2.

Eléments	Formes polynomiales	Formes binaires	Formes décimales
0	0	0000	0
1	1	0001	1
α	α	0010	2
α^2	α^2	0100	4
α^3	α^3	1000	8
α^4	$\alpha + 1$	0011	3
α^5	$\alpha^2 + \alpha$	0110	6
α^6	$\alpha^3 + \alpha^2$	1100	12
α^7	$\alpha^3 + \alpha + 1$	1011	11
α^8	$\alpha^2 + 1$	0101	5
α^9	$\alpha^3 + \alpha$	1010	10
α^{10}	$\alpha^2 + \alpha + 1$	0111	7
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1110	14
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	15
α^{13}	$\alpha^3 + \alpha^2 + 1$	1101	13
α^{14}	$\alpha^3 + 1$	1001	9

TAB. A.2 – Les éléments de $GF(2^4)$

Bibliographie

- [A.95] Glavieux A. Introduction a la theorie de l'information et au codage de canal. *Ecole Nationale Supérieure des telecommunications de bretagne, Departement Signal et Communications*, Novembre 1995.
- [AEM99] Richard E. Ladner Alexander E. Mohr, Eve A. Riskin. Generalized multiple description coding through unequal loss protection. *International Conference on Image Processing*, 1999.
- [AKL03] Thomas Wiegand Ajay K. Luthra, Gary J. Sullivan. Introduction to the special issue on the h.264/avc video coding standard. *IEEE transactions on circuits and systems for video technology, VOL. 13,NO. 7*, July 2003.
- [Bal97] V. B. Balakirsky. Joint source-channel coding for variablelength codes. *ISIT, p. 419, Ulm, Germany, juin/juillet 1997*.
- [Ben07] Abderrahim Benslimane. *Multimedia Multicast on the Internet*. ISTE Ltd, 2007.
- [BP02] A. Banchs and X. Pérez. Providing throughput guarantees in iee 802.11 wireless lan. *IEEE WCNC 02, vol.1*, 2002.
- [CB93] Punya Thitimajshima Claude Berrou, Alain Glavieux. Near shannon limit error-correcting coding and decoding. *Proceeding of the IEEE International Conference on Communications*, May 1993.
- [CJS99] R.L. Chunhung and L. Jain-Shing. Qos routing in ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, August 1999.
- [CM01] P. A. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media microsoft. *Research techrep MSR-TR-35*, Feb 2001.
- [CN99] S. Chen and K. Nahrstedt. Distributed quality-of-service routing in ad hoc networks. *IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad Hoc Networks, 17(8)*, august 1999.
- [Eff98] Michelle Effros. Optimal modeling for complex system design. *IEEE SIGNAL PROCESSING MAGAZINE*, November 1998.
- [E.G03] Lain E.G.Richardson. *H.264 and MPEG-4 video compression, Video Coding for Next-generation Multimedia*. WILEY, 2003.
- [eJH00] R. Bauer et J. Hagenauer. Iterative source/channel decoding using reversible variable length codes. *DCC , pp. 93102, Snowbird, UT*, Mars 2000.
- [EK04] F.Zhai R.Berry Thrasyvoulos N.Pappas E.Soyak, Y.Eisenberg and Aggelos K.Kasaggelos. Channel modeling and its effect on the end-to-end distortion in wireless vidéo communication. *international Conference on Image processing (ICIP)*, 2004.

- [E.S05] Xiaoqing Zhu A.Goldsmith B.Girod E.Setton, Taesang Yoo. Cross-layer design of ad hoc networks for real-time video streaming. *IEEE Communications Magazine, Volume 12, Issue 4, Page(s) : 59 - 65*, Aug 2005.
- [ESG05] X. Zhu E. Setton and B. Girod. Congestion-optimize scheduling of video over wireless ad hoc networks. *IEEE International Symposium on Volume 4, Page(s) : 3531-3534*, May 2005.
- [eTEF98] A. H. Murad et T. E. Fuja. Proceedings of the inter.conf. on image processing (icip), 4-7. October 1998.
- [eXC99] I.S Reed et X. Chen. Error-control coding for data networks. *chap. 5. Kluwer Academic Publishers, 1999*.
- [GH03] Y. Ge and J. Hou. An analytical model for service differentiation in ieee 802.11. *IEEE ICC '03, vol. 2*, May 2003.
- [GJS05] Thomas Wiegand Gary J. Sullivan. Video compression from concepts to the h.264/avc standard. *Proceedings of the IEEE, Vol. 93, NO. 1*, January 2005.
- [G.W09] G.Wilson. Understanding digital video compression. *ERICSSON REVIEW*, 2009.
- [H.C06] G.Shen Z.Xiong S.Li H.Cai, B.Zeng. Error resilient unequal error protection of fine granularity scalable video bitstreams. *EURASIP Journal on Applied Signal Processing*, 2006.
- [HH06] Young Yong Kim Hojin Ha, Changhoon Yim. Packet loss resilience using unequal forward error correction assignment for video transmission over communication networks. *ELSEVIER Computer communications 30 pages 3676-3689*, August 2006.
- [HJ03] Z.J. Haas and J.Deng. On optimizing the backoff interval for random access schemes. *in IEEE Transactions on Communications, vol. 51, no. 12, pp. 2081-2090*, December 2003.
- [HS98] R. Lanphier H. Schulzrinne, A. Rao. Real time streaming protocol (rtsp). *IETF RFC 2326*, April 1998.
- [HS03] R. Frederick V. Jacobson H. Schulzrinne, S. Casner. A transport protocol for real-time applications. *IETF*, July 2003.
- [IA01] C. Castelluccia I. Aad. Differentiation mechanisms for ieee 802.11. *in Proc. of IEEE INFOCOM, vol. 1, pp.209-218, Anchorage, Alaska*, April 2001.
- [IR58] G. Solomon I.S. Reed. Polynomial codes over certain finite fields. *Rapport technique, M.I.T. Lincoln Laboratory Group Report 47*, December 1958.
- [JS08] Rhandley D. Cajote J.Panyavaraporn and S.Aramvith. Joint explicit fmo, fec coding, and adaptive interleaving depth for h.264 wireless video transmission. *IEEE*, March 2008.
- [JYC07] Jitae Shin Jeong-Yong Choi. Cross-layer error-control with low-overhead arq for h.264 video transmission over wireless lans. *Computer Communications 30 ,14761486*, January 2007.
- [Kar09] Azni.Mohamed Frédéric.Guilloud Karim.Mokrani, Ramesh.Pyndiah. Joint source and channel coding for real time h.264 coded video transmission using mobile hand-held devices. *IETE Journal of Research, Vol 55, Issue 4, JUL-AUG 2009*.

- [KM00] S. Mirabbasi K. Martin. Hierarchical qam : A spectrally efficient dc-free modulation scheme. *IEEE Communications Magazine*, November 2000.
- [LHS07] P. J. Cherriman L. Hanzo and J. Streit. *Video Compression and Communications From Basics to H.261, H.263, H.264, MPEG4 for DVB and HSDPA-Style Adaptive Turbo-Transceivers, Second Edition*. IEEE PRESS, 2007.
- [Liu97] Zark et Gupta Liu, Ma. Error control schemes for networks : an overview. *Mobile Networks and Applications 2 (MONET 2)*, pages 167-182, 1997.
- [MBV01] A. T. Campell M. Barry and A. Veres. Distributed control algorithms for service differentiation in wireless packet networks. *Proc. IEEE INFOCOM*, 2001.
- [MJ02] P. Siohan et L. Guivarch M. Jeanne, J. C. Carlach. Source and joint source-channel decoding of variable length codes. *ICC ,vol. 2, New York, USA*, Avril 2002.
- [MM04] T. Turetli C. Barakat M. Malli, Q Ni. Adaptive fair channel allocation for qos enhancement in iee 802.11 wireless lan. *in Proc. of IEEE ICC, vol. 6, pp. 347-375, Paris, France*, July 2004.
- [MP00] D. J. Miller M. Park. Joint source-channel decoding for variable length encoded data by exact and approximate map sequence estimation. *IEEE Transactions on Communications, vol. 48, pages 1-6*, Janvier 2000.
- [MVDS05] CA Davis M. Van Der Schaar, N Sai Shankar. Cross-layer wireless multimedia transmission : challenges, principles, and new paradigms. *IEEE Wireless Communications Magazine, vol. 12, no. 4, pp. 50-58*, August 2005.
- [NS94] Carl-Erik W. Sundberg Nambirajan Seshadri. List viterbi decoding algorithms with applications. *IEEE Transactions on Communications, VOL 42, NO 21314, FEBRUARYIMARCW*, April 1994.
- [NSM06] Nasser Yazdani Nasser Sedaghati-Mokhtari, Mahdi Nazm Bojnordi. Cross-layer design :a new paradigm. *Proc In International Symposium on Communications and Information Technologies (ISCIT), On page(s) : 183-188, Bangkok*, Sept 2006.
- [NTS06] N.V Boulgouris N. Thomos, S. Argyropoulos and M.G Strintzis. Robust transmission of h.264/avc stream using adaptative group slicing and unequal error protection. *EURASIP Journal on Applied Signal Processing, Article ID 51502, Pages 1-13*, 2006.
- [PB94] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. *SIGCOMM Comput. Commun. Rev. Volume 24 , Issue 4, Pages : 234 - 244 ACM New York, NY, USA*, September 1994.
- [P.F97] T.Ottosson A.Svensson P.Frenger, P.Orten. Rate matching in multi-channel systems using rspc-codes. *in Proc. IEEE Vehicular Technology Conference*, 1997.
- [PK03] O. Koufopavlou P. Kitsos, G. Theodoridis. An efficient reconfigurable multiplier architecture for galois field $gf(2^m)$. *Microelectronics Journal 34 volume 3, issue 10, Elsevier*, Octobre 2003.

- [PSB99] R. Sivakumar P. Sinha and V. Bharghavan. Cedar : a core extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad Hoc Networks*,17(8) :1454-1465, august 1999.
- [PYN98] RAMESH MAHENDRA PYNDIAH. Near-optimum decoding of product codes : Block turbo codes. *IEEE transactions on communications ISSN 0090-6778 CODEN IECMBTs*, 1998.
- [QW03] M. A. Abu-Rghe Q. Wang. Cross-layer signalling for next-generation wireless systems. *Proc In IEEE Wireless Communication and Networking , New Orleans, Mars* 2003.
- [RK01] P.A. Humblet R. Knopp. Information capacity and power control in single-cell multiuser communications. *IEEE Transactions on Vehicular Technology*, 2001.
- [R.W06] N.Nikaein C.Bonnet CrossTalk R.Winter, J.H.Schiller. cross-layer decision support based on global knowledge. *IEEE Communications Magazine, ISSN : 0163-6804, pages 93- 99, Volume : 44, Issue : 1*, Jan 2006.
- [S.K06a] E.Steinbach M.Sgroi W.Kellerer S.Khan, Y.Peng. Application-driven cross-layer optimization for video streaming over wireless networks. *IEEE Communications Magazine, ISSN : 0163-6804, pages 122- 130, Volume : 44, Issue : 1*, Jan 2006.
- [S.K06b] Mrinal K. Mandal-S.Panchanathan S.Kumar, L.Xu1. Error resiliency schemes in h.264/avc standard. *Elsevier J. of Visual Communication and Image Representation Vol. 17(2)*, April 2006.
- [SkK05] K.R. Rao Soon-kak Kwon, A. Tamhankar. Overview of h.264/mpeg-4 part 10. *Elsevier Inc*, 2005.
- [SM05] V. Srivastava and M. Motani. Cross-layer design : a survey and the road ahead. *IEEE Communications Magazine, vol.43, no.12, pp.112-119*, December 2005.
- [SS03] P. Karlsson S. Shakkotai, T. Rappaport. Cross-layer design for wireless networks. *IEEE Communications Magazine, vol.41, no.10, pp. 74-80*, October 2003.
- [SYCL96] M. Ammar S. Y. Cheung and X. Li. On the use of destination set grouping to improve fairness in multicast video distribution. *in Proc. IEEE INFO-COM'96,San Francisco, CA, pp.553-560*, Mar 1996.
- [TE00] Caspar Horne Touradj Ebrahimi. Mpeg-4 natural video coding - an overview. *Elsevier Science B. V*, 2000.
- [T.S03] Thomas Wiegand T.Stockhammer, Miska M. Hannuksela. H.264/avc in wireless environments. *IEEE transactions on circuits and systems for video technology*, July 2003.
- [T.S04] M.Bystrom T.Stockhammer. H.264/avc data partitioning for mobile video communication. *ICIP'04*, 2004.
- [Tud95] P.N. Tudor. Mpeg-2 video compression. *Electronics and Communication Engineering Journal*, December 1995.
- [VL02] S. Valaee and B. Li. Distributed call admission control in wireless ad hoc networks. *Proc. IEEE VTC*, 2002.

- [V.T06] S.Iyer V.T.Raisinghani. Cross-layer feedback architecture for mobile device protocol stacks. *IEEE Communications Magazine, Volume 44, Issue 1, Page(s) : 85 - 92*, Jan 2006.
- [WNLJ06] Tung-Lin Liu Wen-Nung Lie, Zhi-Wei Gao and Ping-Chang Jui. Joint source-channel video coding based on the optimization of end-to-end distortions. *Springer-Verlag Berlin Heidelberg*, 2006.
- [XG08] Suyu Wang Lansun Shen Xuejuan Gao, Li Zhuo. A h.264 based joint source channel coding scheme over wireless channels. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing IEEE*, 2008.
- [XJ08] Luc Vandendorpe Xavier Jaspas. Joint source-channel codes based on irregular turbo codes and variable length codes” ,*ieee. transactions on communications, vol. 56, no. 11*, November 2008.
- [YW05] Songyu Yu Yue Wang. Joint source-channel decoding for h.264 coded video stream. *IEEE Transactions on Consumer Electronics, Vol. 51, No. 4*, November 2005.
- [ZC03] Ashfaq Khokhar Zhihui Chen. Improved mac protocols for dcf and pcf modes over fading channels in wireless lans. *IEEE*, 2003.

Résumé

Ce mémoire s'inscrit dans la problématique de la transmission vidéo par paquets en temps réel sur les réseaux sans fil. Un mécanisme capable d'assurer une qualité de la vidéo reçue dans un environnement contraignant, caractérisé par un taux de perte de paquets important et des délais de transmission strictes, est proposé. Le mécanisme proposé est basé sur un codage correcteur d'erreurs, utilisé en amont, associé à une protection inégale contre les erreurs de transmission. Le schéma de codage est particulièrement conçu pour une vidéo codée, compatible avec le standard H.264/AVC. Il s'appuie sur un algorithme d'optimisation débit-distorsion du débit de canal aidé par la source. Cette optimisation utilise l'outil de subdivision de données en partitions d'importances différentes et prend en compte le type et la taille de la partition codée. Un codage de canal au niveau de la couche application, utilisant les codes Reed-Solomon, est adapté aux données qui sont inégalement protégées. En s'appuyant sur les résultats de simulation qui ont été obtenus, la technique ainsi développée a montré son efficacité par rapport à la technique classique de protection égale.

Mots clés

Vidéo temps réel, H.264/AVC, La subdivision de données, Réseaux sans fil à commutation de paquets, Protection inégale, Reed-Solomon, Codage conjoint source canal, Perte de paquets.

Abstract

This thesis focus on the problem linked to the transmission of real time video over packet wireless network. Our objective is to define mechanism able to insuring a quality of video in spite to the problems of packet losses and transmission delays characterizing this type of network. The proposed mechanism is based FEC compatible with the H.264 standard. This mecanism relies on a rate-distorsion algorithm controlling the channel rates under a global rate constraint given by the network. This optimization takes into account the type of packet and his lenght ; the tool of data partionning is used in our proposition. This mecanism leading to an unequal error protection of differents units of coded sequence. A Reed-Solomon channel coding in application layer is adapted to unequal protected data. The experimentation results indicate the efficacy of our proposition compared to the equal error protection.

Key words

Real time vidéo, H.264/AVC, Data partitionning, Packet wireless network, Unequal error protection, Reed-Solomon, Joint source channel coding, Packet losses.