

Panorama des techniques de codage/décodage conjoint et techniques de diversité adaptées à la transmission de flux vidéo et HTML sur lien IP sans fil point/multipoint

Catherine Lamy-Bergot^{*}, Anissa Mokraoui-Zergainoh^{†‡}, Thomas André[◇] et Béatrice Pesquet-Popescu[◇]

(^{*}) THALES Land & Joint Systems, EDS/SPM WaveForm Design group

160 boulevard de Valmy, B.P. 82, F-92704 Colombes Cedex

tél.: +33 1 46 13 27 90 - fax : +33 1 46 13 25 55

Email : `catherine.lamy@fr.thalesgroup.com`

([†]) Laboratoire de traitement et de transport de l'information, Institut Galilée, Université Paris 13

99, avenue Jean-Baptiste Clément, F-93430 Villetaneuse

([‡]) Laboratoire des signaux et systèmes - SUPELEC

3 rue Joliot-Curie, F-91192 Gif-sur-Yvette cedex

Email : `anissa.mokraoui@galilee.univ-paris13.fr`

([◇]) TELECOM Paris Tech

46 rue Barrault, F-75634 Paris Cedex 13

Email : `{thomas.andre, beatrice.pesquet}@telecom-paristech.fr`

Ce travail a été mené grâce à un soutien de l'agence nationale de la recherche via le projet RNRT DITEMOI (conventions n06 TCOM 003-*).

Résumé

L'engouement du grand public pour les applications multimédia sans fil ne cesse de croître depuis le développement d'Internet. Des contraintes d'hétérogénéité de canaux de transmission, de fiabilité, de qualité et de délai sont généralement exigées pour satisfaire les nouveaux besoins applicatifs entraînant ainsi des enjeux économiques importants. A l'heure actuelle, il reste encore un certain nombre de défis pratiques et théoriques lancés par les chercheurs de la communauté des communications numériques. C'est dans ce cadre que s'inscrit le panorama présenté ici.

Cet article présente d'une part un état de l'art sur les principales techniques de codage et de décodage conjoint développées dans la littérature pour des applications multimédia de type téléchargement et diffusion de contenu sur lien mobile IP. Sont tout d'abord rappelées des notions fondamentales des communications numériques à savoir le codage de source, le codage de canal ainsi que les théorèmes de Shannon et leurs principales limitations. Les techniques de codage décodage conjoint présentées dans cet article concernent essentiellement celles développées pour des schémas de codage de source faisant intervenir des codes à longueur variable (CLV) notamment les codes d'Huffman, arithmétiques et les codes entropiques universels de type Lempel-Ziv (LZ). Faisant face au problème de la transmission de données (Hypertext Markup Language (HTML) et vidéo) sur un lien sans fil, cet article présente d'autre part un panorama de techniques de diversités plus ou moins complexes en vue d'introduire le nouveau standard WiFi IEEE802.11n qui intègre un système à multiples antennes d'émission et de réception.

Mots clefs

Codage robuste, codage décodage source canal conjoint, codes entropiques, codes à longueur variable, codes arithmétiques, codes de Huffman, codes Lempel-Ziv, DEFLATE, H.264, Gzip, Zip, diversité, MIMO (multiple input multiple output), 802.11n.

I. Introduction

Le développement du réseau Internet, qui offre à ses utilisateurs des contenus toujours plus nombreux et toujours plus évolués a progressivement convaincu une frange très large de la population de l'intérêt de pouvoir conserver un lien avec le réseau. Le développement plus récent des systèmes sans fil et des terminaux qui les intègrent, en parallèle de l'augmentation des applications HTML (pour HyperText Markup Language) et multimédia ont amené un engouement toujours plus grand du grand public pour les applications multimédia sans fil. Cependant, des contraintes d'hétérogénéité de canaux de transmission, de fiabilité, de qualité et de délai sont généralement exigées pour satisfaire les nouveaux besoins applicatifs entraînant ainsi des enjeux économiques importants. A l'heure actuelle, il reste ainsi encore un certain nombre de défis pratiques et théoriques lancés par les chercheurs de la communauté des communications numériques. Cet article a pour objet de présenter un panorama des techniques existantes dans le domaine de la transmission de flux HTML et vidéo sur un canal avec erreurs ou pertes, en développant en particulier les principales techniques existantes dans la littérature dans le domaine du codage et du décodage conjoint pour des applications multimédia ou HTML, mais aussi pour les aspects diffusion et téléchargement sur lien mobile IP, et enfin les différentes techniques de diversité qui peuvent être considérées pour un tel lien, depuis la diversité d'antennes à la diversité de codage.

L'article est structuré comme suit. Tout d'abord sont rappelées dans la partie II des notions fondamentales de communications numériques. Le principe du codage de source, du codage de canal, de la paquetsation réseau et les théorèmes de Shannon et leurs principales limitations sont présentés, ainsi que différentes solutions et standards récents dans ces domaines. La partie III présente différentes techniques robustes de codage et/ou décodage de source ou source canal conjoint. L'amélioration de la robustesse du système par l'introduction de redondance au sein du codeur de source lui-même, par l'ajout de redondance dans le flux compressé via des techniques de codage source canal conjoint ou encore des techniques de codage décodage source canal conjoint sont considérées. Un intérêt particulier est porté sur les schémas de codage de source faisant intervenir des codes à longueur variable (CLV) comme les codes d'Huffman et arithmétiques ainsi que le cas des codes universels de type Lempel-Ziv (LZ) qui ne font pas partie de la classe des codes à longueur variable. La partie IV traite des différentes techniques de diversités qui peuvent être employées dans le cadre d'une transmission sur lien IP sans fil point/multipoint, ainsi que des structures de réseau et des stratégies de diffusion existantes pour une diffusion point/multipoint. Il est important de comprendre l'impact de la présence de la couche réseau qui dans le modèle traditionnel OSI (Open Systems Interconnection) est située entre la couche application et la couche physique, et qui les empêche donc de communiquer directement, sauf à utiliser des mécanismes dits trans-couches ou *cross-layer*. Les différents axes de collaboration source canal conjoint, tant au niveau codage que décodage, ainsi que la combinaison optimisée de l'emploi de diversité et d'introduction de robustesse pourront se heurter à la présence de cette couche réseau et donc nécessiter la mise en place de solutions *cross-layer*. Enfin, la partie V conclut en esquissant différents axes de recherche pour améliorer la diffusion de flux HTML et multimédia sur lien IP sans fil, notamment via l'utilisation conjointe de techniques de diversité et d'introduction de robustesse.

II. Notions fondamentales de communications numériques

Un système de communication classique est formé principalement de trois blocs de traitement importants : le codeur de source, le codeur de canal et le modulateur [148]. Le codeur de source a pour objectif de comprimer le signal numérique à transmettre via un canal physique de bande passante fixée. Cette opération, jugée primordiale pour des applications multimédia, consiste à extraire pratiquement toute l'information redondante contenue dans le signal. Ainsi, l'information récupérée à la sortie de ce codeur est décrite par un ensemble réduit de symboles appartenant à un alphabet fini. Ces données comprimées sont très vulnérables aux erreurs lors de leur transmission sur un canal bruité. La qualité du signal original reconstruit en dépend fortement. Il s'avère donc important de protéger le flux d'information à la sortie du codeur de source. A cet effet, il est prévu dans la chaîne de transmission classique un codeur de canal. Ce dernier introduit une certaine redondance afin de détecter et de protéger le flux d'information comprimé contre les erreurs lors de sa transmission.

Généralement, les informations à la sortie du codeur de canal sont modulées avant d'être transmises sur le canal. Le récepteur se charge ensuite de réaliser les trois traitements inverses, à savoir la démodulation, le décodage de canal et le décodage de source, comme illustré en figure 1.

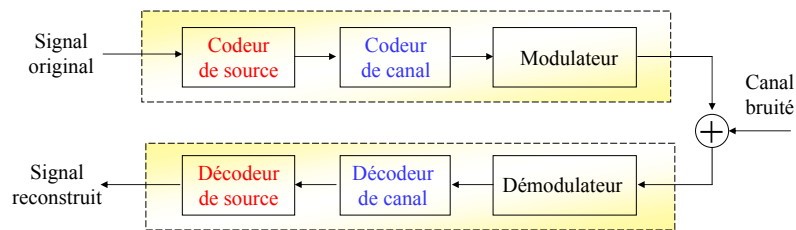


Fig. 1. Chaîne de communication classique.

A. Codage de source

Les applications multimédia actuelles manipulent des contenus hétérogènes : images, séquences vidéo et audio, pages Web ... Ces contenus sont caractérisés par une forte redondance d'information. Afin de pouvoir les transmettre sur un canal physique et de respecter les contraintes de temps réel, il est nécessaire de réduire cette redondance d'information [3][1].

On peut distinguer plusieurs types de redondance selon le contenu manipulé : spatiale, temporelle ou statistique [170]. La redondance spatiale est généralement liée à la forte corrélation des pixels dans une image. Cette redondance est réduite par le biais de transformations (en cosinus discrète, ou en ondelettes). La redondance temporelle est liée à la corrélation disponible entre les images successives d'une séquence vidéo. Cette redondance est généralement réduite par un codage prédictif, tandis que la redondance statistique est liée à la fréquence d'apparition des symboles quantifiés. Cette redondance est réduite par un codage entropique.

On distingue deux classes de codage de source. La première classe de codage correspond aux techniques de compression sans perte. Les codeurs associés à cette classe sont appelés codeurs entropiques. Au décodage, l'information est complètement reconstituée. Ce type de codage est utilisé lorsque les détails contenus dans le signal à comprimer sont importants pour l'application considérée et ne doivent pas être détériorés. Dans la suite de cet article, nous nous intéressons aux codeurs entropiques utilisant les statistiques de la source. Une attention particulière est portée sur (i) les codes à longueur variable, notamment les codes de Huffman et arithmétiques ; et (ii) les codes universels de type Lempel-Ziv qui ne font pas intervenir la redondance statistique.

La deuxième classe concerne les techniques de compression avec perte. Ce type de codage est utilisé pour atteindre des taux de compression élevés. Au décodage, une dégradation, selon le type d'application, est acceptée.

1) **Standards de normalisation** : Cette section présente rapidement les standards de normalisation sur lesquels s'appuient les codeurs de source utilisés dans le cadre d'applications de diffusion et de téléchargement de contenu à savoir le codeur de séquence vidéo H.264 [3] et le codeur entropique DEFLATE [43] de fichiers HTML [2]. Le flux DEFLATE, c'est-à-dire le train d'information comprimé, est ensuite encapsulé selon les spécifications du format Gzip [44] ou Zip [45]. Introduisons tout d'abord le codeur DEFLATE.

a) *Codeur entropique DEFLATE pour le téléchargement de fichiers HTML* : Le protocole de communication informatique client-serveur, appelé HTTP (HyperText Transfer Protocol), est développé pour le World Wide Web. Ce protocole intervient pour le transfert de pages Web entre le serveur HTTP et le navigateur Web. Une page Web comporte à la base un fichier HTML auxquels sont référencés un ensemble d'objets tels que des images au format JPEG ou Gif, Java applet ...

Le HTML est un langage normalisé par le consortium W3C [2]. La toute dernière version est le HTML 4.01. C'est un langage structuré utilisé pour créer des pages Web. Ce langage de balisage est représenté par une série d'éléments de texte codés pour mettre en évidence la structure et le format d'un document à afficher par le navigateur Web.

Le protocole HTTP impose aux clients et serveurs un ensemble de règles à suivre pour le téléchargement de pages Web. Plusieurs spécifications de ce protocole ont été proposées depuis 1990. La dernière version à l'heure actuelle, HTTP1.1, est décrite dans la référence [53]. Dans cette dernière version, une nouvelle fonctionnalité de codage entropique des fichiers HTML y est offerte. Les fichiers sont donc comprimés selon les spécifications de la norme DEFLATE [43]. Celle-ci combine deux algorithmes de codage entropique : (i) une variante de l'algorithme universel LZ-77 [213] que nous présenterons à la section II-A.3 ; et (ii) l'algorithme de Huffman [84] que nous décrirons à la section II-A.2.a. De plus, trois modes de fonctionnement y sont proposés. Le premier correspond uniquement à la segmentation d'un fichier de taille importante sur différents supports de stockage. Le deuxième mode spécifie que le codage de Huffman est réalisé par le biais de tables statiques prédéfinies dans DEFLATE. Ce mode permet d'obtenir une compression et décompression rapide. De plus les tables ne sont pas transmises dans le flux comprimé. Tandis que le troisième mode comprime le train Lempel-Ziv par le biais de tables de Huffman construites de manière dynamique. Le deuxième mode est plus rapide et moins complexe que le troisième mode.

Le codage de source proposé par Lempel et Ziv repose sur la construction adaptative d'un dictionnaire. Cet algorithme, contrairement aux autres algorithmes de la même famille, n'exige pas la lecture du flux d'information avant sa compression. L'idée de base est de considérer une partie du flux de données en entrée comme dictionnaire. Celui-ci est ensuite mis à jour au fur et à mesure du traitement de codage. Une partie des données à traiter est envoyée dans un tampon de longueur fixée par les spécifications données dans [43]. L'algorithme cherche la plus longue chaîne qui puisse correspondre (matching) à celle commençant par le premier symbole dans le tampon contenant les symboles à coder. Le résultat de cette recherche est représenté dans le cas d'un matching par le couple $\langle \text{longueur}, \text{pointeur} \rangle$ soit par le singleton $\langle \text{caractère} \rangle$ lorsqu'il n'y a pas de matching. Notons qu'un codage de Huffman, selon le mode de fonctionnement choisi, est ensuite appliqué au train de codes Lempel-Ziv ainsi obtenus.

b) *Codeur vidéo H.264 pour la diffusion de flux vidéo* : Le codec vidéo le plus récent actuellement utilisé est le codec H.264/MPEG-4 AVC. Initialement développé et standardisé par l'ITU-T, ce codec a également été repris par le groupe ISO MPEG et c'est l'équipe commune de travail ITU/ISO qui l'a finalisé [3][203].

Ce codec vidéo répond donc à la double dénomination H.264 (pour l'ITU) et à la partie 10 du standard MPEG-4 (pour l'ISO). Il a été développé pour plusieurs types d'applications notamment pour des applications de transmission des données multimédia via des canaux sans fil et de la diffusion de vidéos en haute définition sur les ondes radio. Ce codec vidéo est bien plus performant en terme de compression et de fonctionnalités que tous les autres codecs vidéo déjà existants comme illustré en figure 2, où il apparaît sous la dénomination H.26L qui fut sa première appellation pour l'ITU.

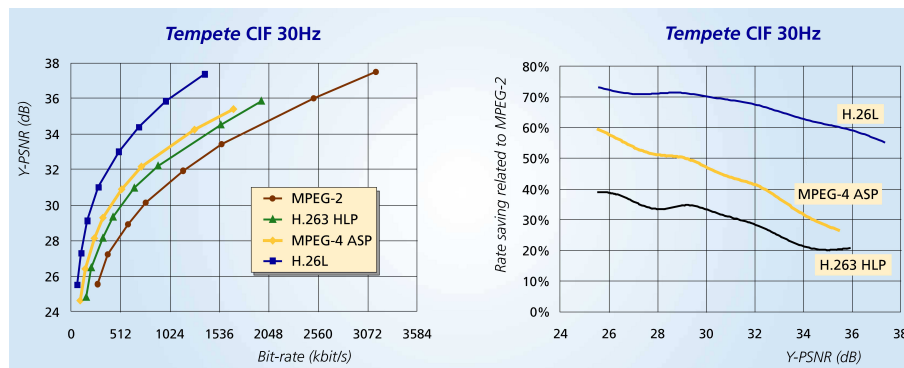


Fig. 2. Illustration de l'efficacité de compression du codeur H.264/AVC par rapport aux codeurs des générations précédentes (source : [166]).

La norme propose quatre profils de codage de façon à ce que le codeur puisse s'adapter aux exigences des différentes applications envisagées [3]. Les principaux profils sont appelés « *Baseline* », « *Main* », « *Extended* » et « *High* ». Le profil « *Baseline* » est prévu essentiellement pour des applications de communications temps-réel tels que la vidéophonie et le DVB (Digital Video Broadcasting). Le profil « *Main* » est prévu pour le stockage des contenus multimédia et les vidéos Haute Définition (HD). Le profil « *Extended* » concerne plutôt les services multimédia faisant intervenir Internet et les transmissions sans fil, et propose un mode de partition de données

(DP) [201] intéressant sur un canal perturbé ainsi que pour des techniques de protection inégale aux erreurs [175]. Le profil « *High* » est particulièrement adapté aux besoins des studios d'enregistrement. Le schéma de la figure 3, dans un premier temps, présente de manière globale les quatre profils et montre le lien qui existe entre eux. Notons que les grandes lignes du codage sont communes à tous les profils.

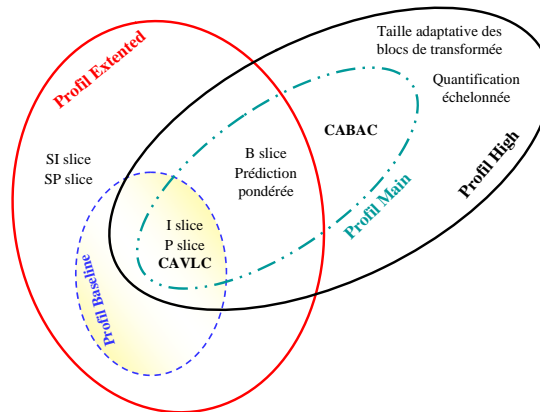


Fig. 3. Vue d'ensemble des différents profils proposés par H.264.

Contrairement à certains standards travaillant sur les images entières, le standard H.264 repose sur la notion de *slice* ou portion d'image, qui correspond à un groupe de macroblocs et est codé indépendamment des autres slices. Égal à tout ou partie seulement de l'image, le slice est indépendant des autres slices de l'image. Le découpage en plusieurs slices définit ainsi implicitement une partition de l'image, ce qui améliore sa robustesse aux erreurs et pertes. Si la notion de slice n'est pas neuve en elle-même, les auteurs du standard H.264 l'ont étendue. On trouve, selon le profil utilisé, plusieurs types de slice possibles : le slice de type I (Intra), le slice de type P (Predicted), le slice de type B (Bi-predictive), le slice de type SP (Switching P) et le slice de type SI (Switching I).

La norme H.264 propose deux types de prédictions : la prédiction Intra et la prédiction Inter. Pour la prédiction Inter, un slice est prédit par compensation de mouvement à partir de l'estimation des vecteurs de mouvement des macroblocs de l'image courante par rapport aux macroblocs de référence. Ces derniers sont situés dans des images précédemment codées. Pour la prédiction Intra, le codeur exploite uniquement les informations contenues dans le slice courant sans faire référence à celles précédemment codées.

Le principe de codage se déroule comme suit. Chaque image de la séquence vidéo est tout d'abord divisée en macroblocs. Pour chaque macrobloc d'un slice de type donné, une version prédite de ce macrobloc est soustraite à sa version originale. C'est cette différence qui est traitée au niveau du codeur présenté par le schéma bloc simplifié de la figure 4. Le traitement se déroule en trois étapes principales : (i) la transformation des macroblocs de différence au moyen d'une transformée entière qui est une approximation de la classique transformée en cosinus discrète (DCT) [203]; (ii) l'opération de quantification des coefficients DCT; et (iii) le codage entropique des symboles

quantifiés.

La norme H.264 propose trois différentes possibilités de réaliser le codage entropique des symboles quantifiés : (i) le codage VLC adaptatif avec contexte appelé CAVLC (Context-based Adaptive Variable Length Coding) ; (ii) le codage arithmétique adaptatif avec contexte, appelé CABAC (Context-Adaptive Binary Arithmetic Coding) ; et (iii) le codage Exponential-Golomb utilisé comme mode par défaut pour les éléments syntaxiques autres que les résidus. Le code Exp-Golomb adopte un format unique (préfixe, suffixe) étendu à la taille nécessaire pour l'élément syntaxique considéré. Les deux premiers types de codage font intervenir des codes à longueur variable.

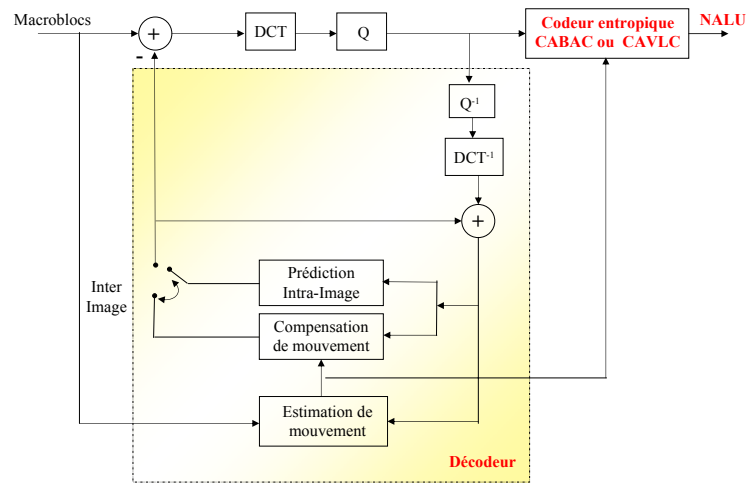


Fig. 4. Schéma bloc simplifié du codeur H.264.

Dans le premier mode, les coefficients de la transformée sont transmis au moyen d'un codage appelé codage à longueur variable avec adaptation au contexte CAVLC. Ce mode de codage est semblable au concept « run, length » employé dans des codecs précédents comme MPEG-2 et H.263, la différence principale est que les tables VLC des différents éléments varient en fonction des éléments précédents. Comme les tables VLC sont adaptées à chaque fois aux statistiques conditionnelles, la performance du codage entropique est améliorée par rapport à celle obtenue par les schémas employant une unique table VLC.

Dans le deuxième mode de codage avec adaptation de contexte (CABAC), le codage repose sur un codage arithmétique. Uniquement disponible dans certains profils de H.264/AVC, ce mode de codage entropique est légèrement plus efficace que le CAVLC mais présente l'inconvénient d'être plus sensible aux erreurs et d'être plus complexe à décoder.

2) **Les codes à longueur variable** : Notons A l'alphabet $\{a_1, \dots, a_{card(A)}\}$ correspondant à l'ensemble des symboles pouvant être émis par la source. Un code à longueur variable est une application de cet ensemble dans l'ensemble C formé par des mots de code correspondant à des séquences de 0 et 1 de longueur finie et variable.

Ce sont les probabilités d'occurrences des symboles appartenant à l'alphabet A qui entraînent la construction des mots de code.

Le modèle de probabilité le plus simple d'un code à longueur variable est celui d'une source sans mémoire. On suppose dans ce cas que chaque symbole est indépendant des autres symboles. Si l'hypothèse d'indépendance n'est pas satisfaite, on utilise dans ce cas le modèle de Markov d'ordre p faisant intervenir la probabilité conditionnelle du symbole courant en fonction de celles des p symboles précédents. Bien entendu, plus l'estimation du modèle de probabilité est précise, plus la compression sera efficace.

Rappelons ci-dessous la définition de l'entropie d'une source. Notons $P(a_i)$ la probabilité d'occurrence associée au symbole a_i émis par la source. L'entropie d'une source sans mémoire est donnée par la relation suivante [40] :

$$H(A) = - \sum_{i=1}^{card(A)} P(a_i) \log_2(P(a_i))$$

Dans le cas où la source est Markovienne, l'entropie de la source se ramène alors à :

$$H(A) = - \sum_{i=1}^{card(A)} \sum_{j=1}^{card(A)} P(a_j)P(a_i | a_j) \log_2(P(a_i | a_j))$$

Rappelons que le théorème de codage de source établi par Shannon affirme qu'il est possible de reconstituer les symboles de la source sans erreur à condition que la longueur moyenne du code C soit supérieure à l'entropie de la source.

Nous présentons, dans ce qui suit, uniquement les codes à longueur variable en étroite relation avec notre projet à savoir les codes d'Huffman, les codes arithmétiques. Les codes particuliers de type Lempel-Ziv y seront également considérés bien qu'il n'appartiennent pas aux codes à longueur variable.

a) *Les codes à longueur variable de type Huffman* : Les codes de Huffman sont pratiquement intégrés dans toutes les nouvelles normes de codage, notamment dans le codec vidéo H.264 et DEFLATE. Ce sont des codes uniquement décodables. En effet, les longueurs relatives des mots de code satisfont l'inégalité de Kraft [40]. A chaque symbole de la source est donc associé un unique mot de code.

L'algorithme de codage utilisé pour déterminer le code à employer est résumé ci-dessous [84] :

- 1- Estimer les probabilités des symboles de la source en fonction de leur fréquence d'apparition ;
2. Classer les symboles de la source dans l'ordre décroissant de leur probabilité ;
3. Générer un nouveau symbole de la source ayant une probabilité égale à la somme des deux symboles les moins probables ;
4. Attribuer les bits 0 et 1 aux deux symboles les moins probables respectivement ;
5. Répéter les étapes 3 et 4 jusqu'à arriver au symbole de la source le plus probable ;
6. Attribuer à chaque symbole de la source le mot de code binaire formé par les bits assignés à l'étape 4.

Il a été montré théoriquement [83][92][103][30], qu'une version « généralisée » de cet algorithme qui consiste à associer un mot de code à plusieurs symboles de la source permet d'améliorer les performances de cet algorithme.

En effet, la longueur moyenne du code est assez proche du seuil de l'entropie de la source. Cependant la réalisation pratique reste assez difficile, parfois même impossible puisque le nombre de bits requis n'est pas entier [185].

L'algorithme de codage décrit ci-dessus, permet à chaque symbole de la source d'attribuer un mot de code binaire unique. L'opération de décodage d'une telle séquence de symboles se fera simplement de manière inversée. En effet à partir de la table des mots de code de Huffman, construite par l'algorithme de codage, les symboles de la source seront décodés.

b) Les codes à longueur variable de type arithmétiques : L'idée directrice des codes arithmétiques est assez ancienne. En effet le codage arithmétique a été initialement évoqué dans le travail original de Shannon sur la théorie de l'information [169]. Cette idée a été exploitée en premier par Elias [52] et publiée par Abramson [5] au début des années 1960. Des mises en œuvre pratiques ont été proposées par Rissanen [161][162] et Pasco [140]. Par la suite, diverses améliorations ont été publiées. Il existe à présent de nombreuses versions d'algorithmes de codage arithmétique. Les travaux de Witten [205], [206] sont des références reconnues. Howard et Vitter ont également proposé des versions d'algorithmes à complexité réduite [82]. Ces algorithmes ont été adoptés par une majorité de standard. Citons par exemple le Q-Coder [141] qui est utilisé dans le standard de compression d'images binaires JBIG. Le MQ-Coder, variante de Q-Coder, est employé dans le standard de compression d'images fixes JPEG2000 [173][182]. Citons également le CABAC [125] qui a été adopté par le standard de compression vidéo H.264/AVC.

Le codage arithmétique se singularise des techniques plus classiques de compression par le fait qu'il travaille sur le message à transmettre tout entier, et non plus sur chaque symbole (ou groupe de symboles) de la source. Ainsi, au lieu d'attribuer à un mot de code un symbole (ou plusieurs symboles) de la source, on subdivise un intervalle de référence pour définir une valeur représentative du message entier. Cette valeur est alors émise sous sa représentation binaire et est décodée en réception par l'opération de sub-division d'intervalles inverse. L'intérêt de cette approche est que chaque symbole (ou groupe de symboles) de la source se retrouve donc codé par un nombre rationnel de bits. Ceci explique le succès des codes arithmétiques qui permettent grâce à cela d'approcher l'entropie de la source avec des complexités d'encodage et de décodage linéaires par rapport à la longueur de la séquence. De plus les probabilités des symboles peuvent être fournies, permettant ainsi d'introduire des modèles adaptatifs.

Considérons plus précisément le procédé de codage et de décodage. Le travail de codage consiste à déterminer un sous-intervalle inclus dans $[0, 1[$ qui sera représentatif du message à codé.

L'algorithme de codage procède comme suit. Notons A l'alphabet de la source. L'intervalle de départ, $[0, 1[$ est divisé en $\text{card}(A)$ sous-intervalles de longueur proportionnelle à la probabilité de chaque symbole. Pour coder le premier symbole, le codeur sélectionne l'intervalle correspondant et le divise en $\text{card}(A)$ sous-intervalles. Le sous-intervalle correspondant au deuxième symbole est alors sélectionné. Comme ce dernier sous-intervalle appartient à l'intervalle sélectionné à l'étape précédente, il représente non seulement le deuxième symbole mais aussi le premier. Les mêmes opérations de division et de sélection sont répétées récursivement jusqu'au codage du dernier symbole.

Avant de présenter l'algorithme de codage arithmétique dans sa version générale, définissons les variables

introduites par l'algorithme. Les variables « *low* », « *high* » et « *range* » correspondent respectivement à la borne inférieure, à la borne supérieure et à la largeur de l'intervalle courant. Le symbole *EOS* (End Of Sequence) est un symbole particulier correspondant au symbole ajouté à la fin de la séquence de symboles de la source à transmettre. L'algorithme de codage arithmétique est résumé ci-dessous [205][206] :

1. Initialiser : $low = 0, high = 1$;
2. Tant que *EOS* n'est pas codé :
 - 2.1. Lire le prochain symbole de source a_i ;
 - 2.2. Calculer $range = high - low$;
 - 2.3. Mettre à jour $high = low + range \times \sum_{j=i}^{card(A)} P(a_j)$;
 - 2.4. Mettre à jour $low = low + range \times \sum_{j=i+1}^{card(A)} P(a_j)$;
3. Transmettre l'intervalle $[low, high[= I_c$;
4. Arrêt.

Le décodage d'une telle séquence se fera simplement de manière inversée, les bornes de l'intervalle de travail évoluant pour le décodage de manière similaire à celle du codage à partir de la valeur I_c transmise, permettant de décoder au fur et à mesure les symboles de la source a_i .

3) **Les codes entropiques universels de type Lempel-Ziv** : Plusieurs algorithmes de type Lempel-Ziv ont été développés dans la littérature [18]. Les algorithmes particuliers de type LZ auxquels nous nous intéresserons dans la suite de cet article diffèrent des algorithmes présentés ci-dessus dans le sens où ils n'exigent aucune connaissance a priori des propriétés statistiques de la source qu'ils engendrent.

Il a été montré que les codes LZ se rapprochent de l'entropie de la source lorsque la taille du dictionnaire est élevée [207]. Toutes les différentes variantes d'algorithmes sont basées sur le même principe qui consiste à découper la source en plusieurs mots tous distincts.

Parmi ces algorithmes de codage Lempel-Ziv, nous nous intéressons dans la suite plus particulièrement à la première version de base appelée Lempel-Ziv-77 (LZ-77) [213]. En effet c'est cette version d'algorithme qui est à l'origine de celle adoptée par le protocole HTTP1.1 [53].

Lempel-Ziv-77 est une méthode de codage entropique développée initialement pour la compression de texte [18][213]. Les données du fichier à compresser sont analysées séquentiellement de gauche à droite.

Notons T le texte composé de n caractères consécutifs. Le i -ème caractère dans T est noté $T[i]$. $T[i, j]$ représente la phrase composée par l'ensemble des caractères $T[i]T[i+1]T[i+2]\dots T[j]$. Supposons que les $n = i - 1$ caractères ont déjà été analysés pour construire $h - 1$ phrases. Ces phrases constituent le dictionnaire noté $T[1, i - 1] = s_1s_2\dots s_{h-1}$, où s_k représente la k -ème phrase.

La méthode de codage repose sur la construction adaptative d'un dictionnaire. A cette étape, l'algorithme de codage cherche dans le dictionnaire (en l'occurrence dans $T[1, i - 1]$) la plus longue h -ème chaîne de caractères qui serait identique à celle disponible dans $T[i, i + l_h - 1]$ avec $l_h \leq L$ où L représente la taille de la fenêtre de

recherche prédéfinie.

La chaîne retenue est codée par un triplet (ou symbole) noté $\langle p_i, l_i, c_i \rangle$, où p_i représente le pointeur vers le dictionnaire indiquant le début de la chaîne à coder, l_i la longueur de la nouvelle chaîne à inclure dans le dictionnaire, et c_i le caractère suivant $T[i + l_i]$ à inclure dans le dictionnaire. Ainsi, le texte codé par l'algorithme Lempel-Ziv-77 est représenté par une suite de triplets (ou symboles) : $\langle p_0, l_0, c_0 \rangle, \langle p_1, l_1, c_1 \rangle, \dots, \langle p_i, l_i, c_i \rangle, \dots$. La figure 5 illustre deux étapes de codage consécutives de l'algorithme Lempel-Ziv-77.

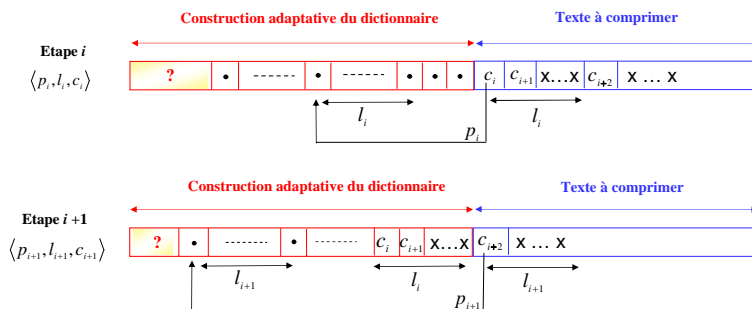


Fig. 5. Schéma de principe de l'algorithme Lempel-Ziv-77.

Dans la référence de normalisation de DEFLATE [43], la taille du dictionnaire ainsi que celle de la fenêtre de recherche sont respectivement fixées à 256 octets et 32 K-octets. Les mots p_i , l_i et c_i sont alors respectivement codés sur 15 bits ($p_i = p_i^{14} p_i^{13} \dots p_i^0$), 8 bits ($l_i = l_i^7 l_i^6 \dots l_i^0$) et 8 bits ($c_i = c_i^7 c_i^6 \dots c_i^0$). D'autre part le triplet est ramené au couple $\langle l_i, p_i \rangle$ dans le cas d'un matching, et sinon au singleton $\langle c_i \rangle$. Il a été montré, pour ce choix particulier de paramètres (i. e. taille du dictionnaire, taille de la fenêtre de recherche), que la longueur moyenne de ce code est aussi proche du seuil de l'entropie de la source.

Le décodage LZ-77 d'une séquence LZ se fera simplement de manière inversée. L'algorithme de décodage LZ s'appuie sur la lecture des symboles transmis $\langle p_i, l_i, c_i \rangle$ par le codeur LZ. Il est impératif, au moment du décodage, de respecter l'ordre chronologique dans lequel les codes LZ ont été générés.

A partir des symboles LZ lus, l'algorithme de décodage LZ construit de manière adaptative le dictionnaire associé au texte codé. A chaque étape, l'algorithme interprète le symbole lu $\langle p_i, l_i, c_i \rangle$ et déduit la chaîne de caractères $T[i, j]$ à insérer dans le dictionnaire. Au final, c'est-à-dire à la lecture du dernier symbole, le dictionnaire contiendra le texte complet décodé.

Dans la version classique de l'algorithme de décodage LZ, le dictionnaire à mettre à jour est supposé initialement vide c'est-à-dire qu'il ne contient aucun texte. Néanmoins, il est possible de l'initialiser par un texte connu, à condition que le codeur le précise et informe le décodeur du texte prédéfini. Cette initialisation a malheureusement

un coût de transmission. C'est d'ailleurs pour cette raison qu'on préfère travailler avec un dictionnaire vide.

B. Codage de canal

A l'heure actuelle, les besoins en communication multimédia se sont multipliés. Les canaux de transmission sont devenus hétérogènes. En effet, ils correspondent à des portions filaires et non filaires. L'Internet filaire est sujet à de nombreuses congestions qui se traduisent par des pertes de paquets de façon aléatoire. Pour pallier ces pertes, le protocole de transport TCP (Transmission Control Protocol) orienté connexion s'appuie sur un mécanisme d'acquittement. Il demande la retransmission des paquets perdus de façon à garantir la fiabilité de la communication. Cependant, ce mécanisme ne permet pas de respecter les contraintes de délai inhérentes par exemple aux applications de diffusion vidéo. Dans ce cas, un protocole comme UDP (User Datagram Protocol) lui sera préféré. En effet, celui-ci ne gère pas la retransmission des paquets et ne garantit donc pas la fiabilité de la connexion. Or, lorsque les données sont transmises sur des liens sans fil, ces données sont également victimes d'évanouissements ou de rafales d'erreurs. Face à un système sans garantie de connexion, il devient donc nécessaire de protéger les bits codés contre les erreurs, c'est le but du codage de canal.

Comme pour le codage de source, Shannon a établi un théorème concernant le codage de canal. Il a introduit la notion de capacité de canal. Celle-ci correspond à la limite fondamentale qui borne la quantité d'information transmissible sur un canal avec une probabilité d'erreurs arbitrairement petite. De plus Shannon apporte la preuve de l'existence d'un tel code. Cependant il n'explique pas la manière de le construire.

Depuis 1948, de nombreux travaux de recherche ont été menés afin de trouver des codes correcteurs d'erreurs de plus en plus proches de la limite de Shannon. Dans ce qui suit, nous listons dans l'ordre chronologique, les différents codes correcteurs proposés dans la littérature.

Citons tout d'abord les premiers codes de canal basés sur la détection de la parité [77], les codes de Golay [61], les codes convolutifs [51], les codes BCH (Bose Ray-Chaudhuri Hocquenghem) [23][24], les codes de Reed-Solomon, les codes cycliques [156], les codes LDPC (Low Density Parity Check) [59]. Bien que performants, ces derniers avaient été mis à l'écart pour des raisons de complexité de temps de calcul sur les calculateurs de l'époque. Des premiers travaux sur les codes concaténés ont ensuite été proposés dans la référence [57]. Les travaux de Viterbi ont permis de mettre en place un algorithme pour le décodage sur treillis en temps réel des codes convolutifs, appelé d'ailleurs jusqu'à présent algorithme de Viterbi [188].

Une nouvelle approche a été ensuite proposée par Bahl, Cocke, Jelinek et Raviv [12]. L'algorithme proposé calcule de manière optimale les probabilités a posteriori des bits transmis à partir de leurs probabilités a priori et du treillis du code convolutif. Cet algorithme, appelé BCJR, est alors dit à sorties souples. Les travaux de Berrou, Glavieux et Thitimajshima, publiés en 1993, ont révolutionné les codes correcteurs d'erreurs [22]. Ces auteurs ont proposé les turbo codes. Depuis, de nombreuses études théoriques se sont penchées sur les algorithmes basés sur les principes de concaténation avec des algorithmes de décodage à décision souple [19].

C. Paquetisation et adaptation au réseau

A l'heure actuelle, le problème de compatibilité et d'adaptation avec les protocoles et les architectures réseaux est une question incontournable dans le développement de tout standard de codage de source. Nous discutons dans ce qui suit de l'adaptabilité de DEFLATE et de H.264 aux réseaux de transmission.

Le codeur H.264 propose d'utiliser deux couches. La première couche, appelée VCL (Video Coding Layer), est directement liée à la représentation du flux vidéo efficacement codé [3]. La deuxième couche, appelée NAL (Network Adaptation Layer), est une couche d'adaptation au réseau. Celle-ci convertit la structure VCL en un format compatible avec le réseau de transmission. La couche NAL divise le train binaire du code en paquets appelés NALUs (NAL Units). Chaque slice est encapsulé dans une NALU séparée. Selon le mode de codage employé, le slice correspondra à l'image entière ou seulement à une de ses parties. Lorsque le CABAC est utilisé, le codeur arithmétique et l'état des contextes sont initialisés au début de chaque NALU. Ceci permet de coder et de transmettre les slices de manière indépendante. Ceci constitue un premier outil de lutte contre la propagation des erreurs de transmission. Néanmoins, vu le caractère adaptatif des modèles statistiques dans le CABAC, la réinitialisation fréquente du codeur se traduit en une perte d'efficacité de compression. Lorsque le CAVLC est employé, la compression est réalisée en codant chaque symbole du flux vidéo par un mot de code à longueur variable provenant d'une table de mots VLC donnée par le contexte, et donc éventuellement dépendante des précédents symboles codés ou des paramètres de compression. Dans ce mode, chaque mot de code est représenté par un nombre entier de bits et les contextes n'ont pas à être réinitialisés entre les NALUs.

Des techniques de masquage d'erreurs sont prévues au sein de son modèle de vérification JM [198] de H.264. Ces techniques permettent de traiter les cas de pertes de macroblocs. Cependant, le problème de la correction d'erreurs éparpillées dans le paquet n'est pas considéré. Lorsqu'un macrobloc est perdu, le décodeur doit reconstituer trois types d'informations indispensables à la continuité du décodage par des techniques de masquage à savoir (i) le vecteur de mouvement associé au macrobloc perdu ; (ii) les informations de texture ; et (iii) le mode de prédiction.

Ces techniques de masquage ne sont néanmoins pas capables de corriger des erreurs ou des pertes. Il est donc facile à comprendre qu'elles ne seront efficaces que pour des taux d'erreurs ou de pertes assez faibles. Au-delà, il s'avère nécessaire d'introduire un codage correcteur d'erreur suffisamment puissant pour ramener le taux d'erreur sur le canal à un taux d'erreur en entrée du décodeur de source suffisamment faible pour que le décodage de source s'effectue presque sans erreurs. Comme nous le verrons plus tard, de nombreux travaux dans la littérature portent donc sur la définition de codes correcteurs (codes de canal) adaptés aux trames vidéo H.264 afin de les rendre plus robustes, notamment dans des applications de diffusion pour lesquelles l'emploi de retransmission n'est pas toujours possible.

Dans les spécifications des standards Gzip [44] et Zlib [45], le problème de la paquetisation et de l'adaptation au réseau de transmission n'est pas considéré. Il est donc important de réfléchir à une stratégie compatible avec le réseau en question de façon à anticiper la lutte contre les erreurs de propagation.

D. Les principales limitations du théorème de séparation

Les systèmes de communications classiques actuels s'appuient sur des blocs de codage de source et de codage de canal optimisés séparément (voir figure 1). Le codeur de source, comme indiqué ci-dessus, réduit au maximum la redondance de la source à transmettre tandis que le codeur de canal ajoute de la redondance de manière contrôlée. Cette stratégie de séparation est directement liée aux théorèmes énoncés par Shannon. Shannon stipule qu'il est possible de transmettre des messages avec une erreur arbitrairement faible même sur un canal bruité, à condition que le débit d'information sur le canal soit inférieur à sa capacité. Shannon a montré, sous certaines hypothèses, que les systèmes de communications où chaque bloc est optimisé séparément peuvent atteindre des performances optimales au sens de la compressibilité de la source et de la capacité du canal.

Cependant, les hypothèses de ce théorème sont rarement vérifiées en pratique. En effet, le théorème de séparation n'est plus valable si une contrainte de délai existe. De même, il n'est pas applicable si le volume de données à transmettre est borné et est également petit. Généralement, les protocoles réseaux les plus répandus manipulent des paquets de taille relativement petite. De plus, le théorème suppose qu'il est possible de modéliser le canal de manière parfaite, ce qui n'est pas le cas puisque les caractéristiques du canal peuvent varier au cours du temps.

Notons que le théorème de séparation démontre qu'il existe au moins un schéma de codage mais ne garantit pas une complexité de calcul acceptable et de ce fait un schéma réalisable. De plus, en pratique les émetteurs et récepteurs peuvent ne pas posséder la même capacité de calcul (ex. stations de base d'opérateurs réseaux, ordinateurs, terminaux mobiles). En effet, pour des raisons de coût algorithmique, il est possible que le codeur n'exploite pas toute la redondance de la source. Il est donc difficile d'atteindre les limites de performance du système de communications telles que prévues par Shannon. Plusieurs travaux de recherche ont alors adopté de nouvelles stratégies visant à optimiser conjointement le codage de source et le codage de canal donnant lieu à des techniques de codage/décodage source canal conjoint.

L'objectif principal des travaux portant sur la diffusion de vidéo ou de flux HTML sur des liens sans fil vise à rendre robuste les trains compressés transmis sur des canaux bruités et hétérogènes. Rappelons que les codes à longueur variable sont largement utilisés dans les standards actuels de codage de séquence vidéo [3][204] ou de texte [44][45]. Cependant, ces codes à longueur variable sont extrêmement sensibles aux erreurs bits. En effet, une simple erreur bit induit plusieurs erreurs au niveau de la reconstruction des symboles. De plus, des problèmes de synchronisation surviennent et s'avèrent catastrophiques quant à la reconstruction du train binaire original. Actuellement des solutions de retransmission existent, qu'il s'agisse de solution au niveau de la couche liaison (MAC pour Medium Access Control) ou au niveau de la couche transport, comme le protocole TCP qui garantit la fiabilité de la transmission au moyen d'un mécanisme d'acquittement et de demande de retransmission des paquets perdus (ARQ pour Automatic Repeat Request). Cependant, les mécanismes de retransmission, en particulier à haut niveau comme TCP, ne permettent pas, lorsque la liaison est fortement perturbée, de garantir une qualité de service tout en respectant les contraintes de délai inhérentes aux applications temps réel de type diffusion vidéo (streaming) [76].

Une solution viable serait donc d'adjoindre à cette solution déjà disponible, des techniques robustes de codage et décodage source canal conjoint de façon à utiliser le moins possible les demandes de retransmission. Ainsi la robustesse du système sera renforcée tout en respectant les contraintes de temps réel liées aux applications multimédia.

III. Techniques robustes de codage et décodage source canal conjoint

Cette partie présente une liste non exhaustive des différentes techniques de codage robuste de source et de codage/décodage conjoint source canal dont l'objet est de répondre à la demande précédemment établie, c'est-à-dire celle de l'offre d'une bonne qualité de service tout en respectant les contraintes de délai inhérentes aux applications temps réel de type diffusion vidéo (streaming).

Les techniques de codage robuste de source qui sont présentées au paragraphe III-A peuvent être vues comme un moyen d'introduire de la redondance en menant le travail directement sur l'information source. Alors que les techniques de codage source canal conjoint, présentées au paragraphe III-B travaillent quant à elles dans le but d'introduire de la redondance à partir du flux codé. Enfin, le paragraphe III-C traite des techniques de décodage source canal conjoint robuste, qui ont pour vocation de corriger au mieux le flux reçu en tirant partie de toute redondance disponible à la réception.

A. Techniques de codage robuste de source

1) **Principe du codage par descriptions multiples** : Parmi les techniques de codage robuste de source, nous présentons dans ce qui suit le codage par descriptions multiples. L'utilisation de descriptions multiples (ou MDC pour Multiple description coding) est une technique de codage de source typiquement adaptée à la transmission de flux multimédia sur des canaux non fiables (best effort, sans garantie de délai d'acheminement, sans garantie de qualité) comme les canaux IP sans fil. La technique MDC a pour objet de lutter contre les pertes de paquets liées aux pertes par congestion ou aux pertes suite aux erreurs de transmission. De la redondance est introduite au niveau source, afin (i) de garantir un niveau minimum de qualité à la réception ; et (ii) d'éviter un emploi trop fréquent de masquage d'erreur qui s'avère à la longue désagréable pour l'utilisateur. En pratique, les méthodes MDC sont plus adaptées aux effacements qu'aux erreurs bits, et sont donc naturellement plus utilisées dans les contextes de pertes.

Succinctement, la technique MDC repose sur la séparation de l'information originale en n flux corrélés qui seront transmis sur n canaux indépendants (par exemple des chemins différents dans le réseau à commutation de paquets Internet ou des slots différents sur le lien sans fil, ...). Les pertes étant statistiquement décorréélées pour les différentes descriptions, le récepteur est à même d'en recevoir au moins une qui lui garantit une reconstruction de qualité minimale. En cas d'erreur, la description n'est pas prise en compte et la reconstruction se fait au niveau du récepteur avec les descriptions reçues correctement. Ce procédé amène à une qualité réduite mais supérieure à celle obtenue par une technique de masquage. Une présentation générale des problématiques du codage par descriptions multiples (MDC) est effectuée dans les références [63][195]. L'objectif poursuivi est de produire plusieurs descriptions du

signal à transmettre. Ces descriptions doivent être, dans la mesure du possible, indépendantes et équilibrées, de sorte que le décodage de l'une d'elles suffise à obtenir une qualité suffisante (faible distorsion latérale), et de sorte que chaque description supplémentaire reçue permette d'améliorer cette qualité. Bien entendu, on souhaite également rester aussi proche que possible des performances d'un codeur à description unique dans le cas où toutes les descriptions sont reçues correctement (faible distorsion centrale).

Le principe du MDC implique l'ajout de redondance au signal à transmettre, par rapport à un codage à description unique. On peut distinguer plusieurs types de redondance [195]. En particulier, la redondance peut être due :

- à la transmission effective des mêmes données dans plusieurs descriptions ;
- à la nécessité de coder des informations supplémentaires ;
- au fait que coder séparément deux ensembles de données corrélées est moins efficace que de les coder conjointement.

2) **Construction des descriptions** : On peut distinguer trois grands types d'algorithmes de compression par descriptions multiples : (i) par sous-échantillonnage (dans le domaine spatial, temporel ou fréquentiel), (ii) par quantification, ou (iii) par transformée. Les méthodes "concurrentes" par codage correcteur d'erreurs (FEC pour *Forward Error Correction*) et par codage scalable sont évoquées plus loin. En fonction, notamment, du nombre de descriptions ou du taux de redondance désiré, l'une ou l'autre de ces méthodes s'avère plus performante.

L'un des problèmes à résoudre est celui du mismatch. Il s'agit de la situation où le décodeur, en l'absence de l'une des descriptions, est incapable de reproduire la situation prévue au codeur. Dans le cas d'un codeur prédictif, par exemple, cela se traduira par l'utilisation d'une référence approximative lors du décodage.

a) *Par sous-échantillonnage (splitting)* : La méthode du splitting est définie de façon générale dans [90] et appliquée à la vidéo dans [10]. Il en existe de nombreuses variantes. Cette méthode conduit à des descriptions équilibrées, et en nombre dyadique. Mais elle offre une mauvaise efficacité de codage dans le cas où l'on choisit de nombreuses descriptions. De ce fait, la plupart des applications en temporel se limitent à deux descriptions. Or, dès que le taux de perte augmente, il devient intéressant d'avoir de plus en plus de descriptions [138][58]. Cependant, notons par exemple que, dans [11], le principe du splitting simple est étendu à des descriptions déséquilibrées.

Il est intéressant de noter que l'emploi de différentes descriptions n'empêche pas d'estimer le taux de redondance. Ainsi dans [189], est proposé un schéma adapté à H.264 permettant un réglage précis du taux de redondance. Cette méthode tire partie de la corrélation du signal dans le domaine où il est sous-échantillonné. Ainsi, il reste possible d'estimer de façon acceptable une description manquante à partir des autres descriptions. On définit les descriptions par entrelacement temporel d'images [200], par entrelacement de pixels [190][58] ou de vecteurs de mouvement [98] ou par entrelacement de coefficients de transformée [36][37]. La redondance est alors déterminée par les statistiques des sources, à moins qu'un pré-filtrage ne permette de régler un compromis entre la redondance et la distorsion latérale.

Un des points importants de ce travail reste naturellement la question de l'optimalité de la partition en descriptions,

comme discuté dans [13]. Ceci est à rapprocher des choix de prédiction et de sous-échantillonnage qui varient grandement dans la littérature. Ainsi, les algorithmes développés dans [200][10][191][58] utilisent des prédicteurs multiples, indépendants pour chaque description, permettant ainsi de reconstruire la séquence correspondante sans mismatch en cas de perte de l'autre description. L'inconvénient de cette méthode provient de l'augmentation du débit due au fait que le codage indépendant des deux descriptions est forcément moins efficace que leur codage conjoint. Dans [58], on utilise un sous-échantillonnage spatial en quinconce au lieu d'un sous-échantillonnage temporel plus traditionnel. Avant le sous-échantillonnage, on effectue un pré-filtrage passe-bas dans le domaine DCT, ce qui permet de contrôler la quantité de redondance. Afin de limiter le mismatch induit par le filtrage, on transmet également l'erreur de prédiction entre le signal central décodé avec et sans la description courante. Ailleurs, comme dans [98], les vecteurs de mouvement et les coefficients DCT sont répartis par le biais d'une grille en quinconce. La redondance du codeur est alors très faible, ce qui conduit à une distorsion latérale importante. On peut également, comme dans [97], prédire chaque bloc à l'aide de blocs des deux images précédentes compensées en mouvement, mais on ignore le mismatch.

Certains auteurs se posent également le problème de la limitation de l'impact de l'erreur. Ainsi dans [191], chaque description reçoit une image sur deux. Le prédicteur central forme une superposition linéaire des deux dernières images reconstruites. Le codage optionnel de l'erreur de mismatch permet de contrôler de façon très flexible le niveau de redondance. Cela permet non seulement de limiter l'impact de l'erreur, mais aussi d'améliorer l'estimation de la description manquante. La stratégie adoptée est différente selon le type de réseau. On peut aller plus loin en tirant partie, comme dans [153], d'un outil de robustesse de H.264/AVC, les « *redundant pictures* ». Le schéma proposé permet d'atteindre de très bonnes performances.

D'autres approches comme par exemple celle proposée dans [160] sont, dans l'esprit, proches du splitting tout en introduisant de la redondance. Les vecteurs de mouvement et une grande partie des coefficients basse fréquence (BF) y sont dupliqués, alors que les autres données sont réparties alternativement dans les deux descriptions. Quoique simple, cette approche conduit à de bien meilleurs résultats que des méthodes plus complexes à base de transformée corrélante (voir paragraphe ci-après) pour de forts taux de redondance. Dans [38], la technique employée est identique, mais la répartition est légèrement différente. Chaque description reçoit alternativement ou bien la seule couche de base, ou bien la totalité des coefficients d'un bloc ou d'une image. Cela permet d'obtenir deux descriptions équilibrées. Dans [99], les coefficients DCT dupliqués peuvent être basse-fréquence ou haute-fréquence (HF). Dans tous ces algorithmes, le nombre de coefficients dupliqués peut être facilement ajusté en fonction des caractéristiques de la source et des canaux. Ceci est réalisé dans la référence [160] de façon à optimiser le compromis RRD (redondance - débit - distorsion). Cette idée est également reprise dans les références [38][39][129][100].

b) Par quantification : Les algorithmes de quantification par descriptions multiples, tels que ceux développés dans [56][151][186], associent à chaque valeur d'entrée un indice par description, ce qui est appelé quantification scalaire (QS), par opposition à la quantification vectorielle (QV), traitée au paragraphe III-B.2. La redondance introduite et la distorsion latérale sont contrôlées par l'attribution de ces indices à chaque classe de quantification. C'est en pratique un compromis débit/distorsion qui permettra de décider de ces attributions.

Par exemple, dans [187], on construit un codeur MDC sans mismatch reposant sur un algorithme de raffinement mutuel. Dans [110], on transmet la somme de l'erreur de prédiction et de l'erreur de mismatch. Le même quantificateur devant être employé pour les deux signaux d'erreur, ce schéma ne peut pas être utilisé pour contrôler séparément la distorsion centrale et la distorsion latérale.

c) *Par transformée* : Les principales transformées proposées dans les normes vidéo sont présentées ci-après.

– *Par transformée corrélante* :

Les transformées corrélantes pour le MDC (soit les multiple description correlating transforms ou MDCT) ont été introduites dans [192]. Au lieu de coder de façon indépendante deux variables décorréelées, on introduit une certaine quantité de redondance entre elles. Dans ce schéma, la quantification des données a lieu avant l'opération de transformation. Les coefficients de chaque description doivent être décorrélés afin de maximiser l'efficacité de codage. Au décodeur, les coefficients manquants peuvent être estimés à partir de ceux déduits de l'autre description.

Les premiers travaux publiés sur ce sujet font appel à la « piecewise correlating transform » (PCT), qui est une transformée linéaire non-orthogonale. La transformée optimale, qui minimise la distorsion latérale pour une redondance donnée, est paramétrée par une seule variable qui contrôle la quantité de redondance induite [193]. Une approche généralisée est proposée dans la référence [60].

Les performances de ces codeurs sont très bonnes lorsque le taux de redondance est faible. Une petite augmentation de la redondance entraîne une diminution très rapide de la distorsion latérale. Cependant, la transformée est inefficace pour de forts taux de redondance, et les performances s'en ressentent. C'est pourquoi le schéma hybride proposé dans [196] utilise une transformée corrélante pour les faibles redondances, et une description multiple par couches plus efficace pour les redondances plus élevées.

Dans les références [157] et [109], un codeur vidéo à deux descriptions est présenté. Celui-ci utilise trois boucles de prédiction à l'encodeur (une boucle centrale et deux boucles latérales), prenant en compte toutes les situations possibles au décodeur. La PCT est utilisée pour coder l'erreur de prédiction centrale dans deux descriptions. Les deux boucles latérales codent grossièrement l'erreur de mismatch qui serait obtenue en l'absence de l'une des descriptions. Dans [85], on n'envoie pas l'erreur de mismatch comme un signal additionnel dans chaque description. Une seule description contenant la somme de l'erreur de prédiction et de l'erreur de mismatch est envoyée.

Il reste cependant difficile de construire une bonne transformée corrélante pour une source de distribution arbitraire. De plus, il est généralement difficile d'étendre ce type d'approche à plus de deux descriptions.

– *Par transformée redondante* :

Il s'agit de l'une des approches les plus prometteuses pour générer des descriptions avec une quantité contrôlée de corrélation. Les exemples typiques de transformées redondantes font appel aux trames (frames) [65] et à la poursuite adaptative (ou MP pour matching pursuit) [119] pour générer les descriptions. La quantification intervient ici après la transformée.

L'utilisation des trames est limitée par le fait que tous les sous-ensembles de trames reçues ne permettent pas de reconstruire un signal de bonne qualité [65]. Pour cette raison, la plupart des travaux emploient les trames

dités harmoniques (m, n) avec $m \geq n$. Ces trames présentent la propriété intéressante de garantir une bonne reconstruction du signal malgré la perte de $m - n$ éléments de la trame. Par exemple, il est montré dans [64] et [31] que ce schéma obtient de meilleures performances que les schémas fondés sur la protection inégale aux erreurs (ou UEP pour Unequal Error Protection) comme introduit au paragraphe III-B.1.b. D'autres trames fondées sur la DCT sont utilisées dans [35]. Dans la référence [145] est présenté un schéma de décodage de trames d'ondelettes par descriptions multiples fondé sur des techniques d'optimisation convexe.

Dans les références [181] et [136], un schéma reposant sur le MP est présenté. Celui-ci permet de générer deux descriptions. Les atomes qui approximent au mieux le signal sont répétés dans les deux descriptions, alors que les atomes suivants sont répartis alternativement entre elles. La quantité de redondance est alors contrôlée par le nombre d'atomes partagés. On duplique donc l'information importante, sans exploiter la redondance inhérente à la transformée en ondelettes. Dans [181], cette information supplémentaire est également exploitée pour améliorer la qualité du décodeur central par maximum de vraisemblance.

Dans [136], on optimise la quantité d'information mismatch codée en fonction du taux de perte du canal. Dans [152], on regroupe les atomes similaires entre eux de sorte à obtenir une bonne représentation du signal, puis on les distribue parmi les descriptions. Cette approche tire partie de la redondance de la transformée et permet de réduire la distorsion centrale.

Une approche combinant la quantification scalaire et la poursuite adaptative pour générer des descriptions est présentée dans la référence [29].

Dans [184], les descriptions sont construites à l'aide d'une transformée en ondelettes temporelle redondante avec compensation de mouvement. Ce schéma permet de régler le taux de redondance en fonction du nombre de niveaux de la transformée. Il fait suite à de nombreuses études qui ont proposé l'extraction de descriptions multiples à partir des codeurs scalables basés ondelettes [73][142] ou multi-ondelettes [32], avec éventuellement, un décodage exploitant les codes turbo [72] ou d'autres techniques itératives basées sur l'optimisation convexe [145] pour la reconstruction optimale. Dans [144][146] est présentée une nouvelle approche du codage par descriptions multiples (MDC) basée sur une synthèse à l'aide d'un opérateur de trame quelconque. Cette méthode vise à exploiter le fait que la source admet une représentation creuse dans la trame considérée. L'idée s'inspire de la théorie du « Compressed Sensing » (CS) qui a pour but l'acquisition et la compression simultanée de données. Cette approche diffère de celle classiquement adoptée qui consiste à acquérir le signal sous sa forme complète, pour ensuite le quantifier et le compresser et ainsi éliminer une quantité considérable de composantes acquises, tout en essayant de conserver un maximum d'informations. Cette approche récente, développée notamment dans [27][47], permet de reconstruire de façon optimale, à l'aide de projections aléatoires, un signal ayant une représentation creuse dans une base. Le nombre de projections nécessaire est de l'ordre de grandeur du nombre de coefficients non nuls, et par conséquent beaucoup plus petit que la taille initiale du signal. L'article décrit comment construire un schéma à plus de deux descriptions pour la transmission d'images sur des réseaux à pertes. A partir d'une représentation fixée au niveau de la synthèse, les coefficients peuvent être codés de manière à minimiser le débit pour une distorsion maximale fixée et le problème est ensuite re-formulé sous la forme d'un problème d'optimisation convexe.

– *Par ajout de symbole interdit :*

Une autre technique, qui peut être également regroupée avec ces transformées redondantes, est celle qui consiste à introduire des symboles ou intervalles interdits dans les trains binaires des codes à longueur variable. Cette technique introduit une redondance dans la source elle-même, afin d’obtenir une meilleure robustesse en permettant la détection d’erreurs de transmission.

Dans le cadre de codage arithmétique, Boyd et al. [24] proposent de réserver une partie de l’intervalle unité $[0, 1[$ à un symbole, appelé symbole interdit (SI), ne faisant pas partie de l’alphabet de la source afin que le décodeur puisse détecter les erreurs de transmission. Ce symbole interdit n’est jamais codé par le codeur arithmétique. Cependant une probabilité non nulle lui est attribuée $P(a_{SI}) = \varepsilon$. L’intervalle initial $[0, 1[$ est donc ramené à $[0, 1 - \varepsilon[$. Il a été montré, dans la référence [33], que la procédure d’insertion d’un symbole interdit correspond à une redondance par bit codé de $\rho_{SI}(\varepsilon) = -\log(1 - \varepsilon)$ (bit/symbole).

Une autre forme de redondance consiste à insérer des marqueurs de synchronisation dans le train de symbole de source. Dans la référence [25], c’est le symbole interdit qui est inséré périodiquement pour jouer le rôle de synchronisation.

Dans la référence [41], les marqueurs de synchronisation sont placés à des positions connues du décodeur. Cette procédure se traduit par l’utilisation, à des intervalles réguliers, d’arbres de conversion binaire étendus où certains nœuds sont prolongés par un suffixe binaire de taille variable. Les transitions correspondantes à ces marqueurs de synchronisation sont déterministes. La fréquence d’insertion et la longueur des marqueurs de synchronisation sont réglées arbitrairement selon le niveau de redondance désiré.

B. Techniques robustes de codage source canal conjoint

L’idée principale des techniques de codage source canal conjoint est de modifier le codeur de manière à introduire de la redondance dans le train binaire codé [49]. Plusieurs formes de redondance ont été envisagées dans la littérature dans le but d’améliorer les performances de décodage.

L’inconvénient de ces techniques est qu’elles nécessitent pour la plupart l’introduction d’une redondance artificielle à l’aide d’un symbole interdit ou de marqueurs de synchronisation. C’est l’ajout de cette redondance qui est ensuite exploitée par le récepteur en vue de détecter les erreurs de transmission. Présentons dans ce qui suit les principales approches développées.

1) Introduction de redondance par codage correcteur d’erreur :

a) *Ajout de marqueurs :* La forme la plus simple de redondance consiste à introduire des marqueurs dans le flux codé. De part leur nature redondante, ils permettent la synchronisation dans un codeur vidéo H.263+. Le résultat obtenu dans la référence [106] est illustré par la figure 6. La séquence vidéo Foreman, au format CIF, est codée à une fréquence d’images de 30Hz, à un débit de 256Kbit/s. Le canal de transmission, à pertes de paquets, est simulé par des paramètres du modèle de Gilbert $p = 0.08$ et $q = 0.60$, correspondant à un taux moyen de pertes de 10%. La figure 6 montre deux images dans la séquence reconstruite au même instant, respectivement sans

utiliser de marqueur de synchronisation (image (a)), et en insérant un marqueur à chaque début de GOB (image (b)). L'image originale correspondante apparaît en (c). La figure 6 montre clairement l'intérêt des marqueurs de synchronisation. Ils permettent de limiter la propagation spatiale des pertes de paquets. En effet, en raison de la désynchronisation du décodeur, des macroblocs se trouvent reconstruits à des positions spatiales aberrantes (voir le résultat sur l'image (a)).

D'autres formes de redondance ont été envisagées afin d'améliorer les performances de décodage. Citons par exemple l'ajout de contraintes de terminaison (nombre de bits et/ou de symboles dans une séquence transmise), la transmission d'informations adjacentes ...



Fig. 6. Illustration de l'intérêt des marqueurs syntaxiques pour la robustesse (source : [106]).

b) Techniques de protection inégale aux erreurs ou pertes d'un code source : La séparation traditionnelle des opérations de codage de source et de codage de canal, instituée par l'application du théorème de séparation de Shannon et pour une plus grande facilité d'usage pratique, en particulier dans le cas d'une transmission sur un système reposant sur la pile protocolaire OSI, a montré ses limites. Les contraintes pratiques imposées par les applications de diffusion multimédia, en terme de limitation de complexité et de contraintes de délai ont amené la communauté scientifique à re-considérer des solutions de codage telle que le codage tandem, dont fait partie l'application de protection inégale aux erreurs. Ces techniques UEP ont pour objet de protéger plus efficacement les portions les plus sensibles aux erreurs du flux, et moins efficacement les portions les moins sensibles aux erreurs, afin d'allouer efficacement le débit utile. Citons par exemple les travaux développés dans la référence [105] pour

la protection du flux H.263 face à un canal avec erreurs. Ces approches sont naturellement adaptables au cas de canaux avec effacements, le type de codage correcteur utilisé variant en général en fonction du canal considéré.

Lorsque l'on considère la possibilité d'ajouter de la redondance à un flux de données, et que l'on souhaite adapter finement le taux de redondance au flux en question, la solution qui vient naturellement à l'esprit est d'insérer le module de codage correcteur générant la redondance au plus près du codeur de source, c'est-à-dire au plus près de celui qui connaît le rôle exact (et donc l'importance) de chacun des bits codés. C'est ainsi que les différentes parties du flux pourront être facilement protégées selon leur niveau de sensibilité et de leur importance dans la reconstruction finale au niveau du récepteur. Une telle technique correspond à l'insertion de protection UEP au niveau applicatif, à laquelle il est parfois fait référence sous la dénomination d'UEP-APPLI. On trouve des propositions de ce type par exemple dans [21].

On peut également souhaiter insérer la protection à d'autres niveaux de la transmission, plus particulièrement dans le cadre classique de la pile protocolaire OSI, par exemple comme données supplémentaires au niveau transport, comme la référence [199] tente de le normaliser ou naturellement au niveau de la couche physique considérée.

Dans tous les cas, lorsque l'on souhaite appliquer efficacement une protection différenciée, se pose la question de l'estimation, voire de la modélisation, de la sensibilité aux erreurs ou aux effacements et de la classification de l'importance des différentes portions du flux. Cette question a été abondamment traitée dans la littérature, par exemple dans [78][126][21], mais l'existence d'un modèle complet, précis, dynamique, temps réel et adapté au standard de compression H.264/AVC reste à établir. Vient ensuite le problème de la transmission de cette estimation de sensibilité et donc de la signalisation, problème qui dépend de la notion de *cross-layer design* ou design inter-couches. En dépit de remarques de prudence tout à fait justifiées [94], de nombreux chercheurs ont en effet proposé d'aller plus en avant dans la stratégie d'intégration des procédures de transmissions efficaces face aux erreurs sur liens de type IP sans fil. Ils ont montré qu'une approche de type cross-layer pouvait être un complément utile à une approche en couches [42][127]. En effet, il semble crucial de rappeler que l'approche cross-layer ne peut, de manière réaliste, remplacer totalement l'approche en couches à court ou long terme. En effet, seule l'approche en couches permet de garantir l'interconnectivité de nombreux services et protocoles les plus variés. En revanche, une approche cross-layer transparente [127] pour les couches protocolaires permet d'améliorer le fonctionnement sans mettre en péril l'ensemble du système.

2) **Quantification vectorielle optimisée canal (COVQ)** : De même qu'il est possible d'optimiser le codeur de canal (ou plus généralement un codeur correcteur d'erreur) en fonction de la source, on peut aussi optimiser le codeur de source en fonction du canal. Cette autre technique de codage conjoint source canal consiste à optimiser le codage de la source directement pour un canal bruité connu, afin de minimiser la distorsion globale du système. Une des caractéristiques cruciales pour la performance du système, surtout lorsque l'on utilise un quantificateur vectoriel (QV), est l'étiquetage binaire, c'est-à-dire l'assignation d'un mot de code source à un mot de code canal.

De façon générale, il est souhaitable que les mots de code proches en distance euclidienne correspondent à des étiquettes binaires proches en distance de Hamming. Ainsi, si un bit est erroné, la distorsion engendrée reste faible

car l'erreur de transmission se traduira par la reconstruction d'un centroïde « voisin » de l'initial. Il s'agit donc d'un problème de topologie. Les distances entre étiquettes du dictionnaire doivent « ressembler » aux distances entre centroïdes. Farvardin [54] a proposé une telle stratégie d'optimisation conjointe source canal pour un modèle de canal discret. En fait, le fonctionnement d'un algorithme comme l'algorithme Linde-Buzo-Gray (LBG), traditionnellement utilisé pour la recherche du dictionnaire de la QV, se prête bien à une modification en ce sens. Le LBG optimise itérativement les positions des centroïdes puis celles des frontières de telle manière à réduire la distorsion. D'autres algorithmes, reposant sur les cartes auto-organisatrices telles que les cartes de Kohonen optimisent la quantification en cherchant à réduire l'impact visuel, par exemple des contours de l'image considérée [102]. Maintenant, si l'on choisit une mesure de distorsion qui inclut celle introduite par le canal, le même algorithme effectuera l'optimisation conjointe. Cette méthode adaptée à un canal avec erreurs porte assez souvent le nom de « channel optimized vector quantization » (COVQ) dans la littérature.

Cette approche a été également utilisée dans le cas d'un système de codage conjoint source canal reposant sur des modulations hiérarchiques suivies d'un codage de source multi-résolution. On peut citer les travaux de Vetterli et al. [154] qui l'ont mis en place pour la première fois ou encore d'autres proposant l'emploi d'un quantificateur vectoriel structuré construit à partir de constellations issues de réseaux de points et d'une assignation linéaire d'étiquette minimisant simultanément la distorsion canal et la distorsion source [55].

3) **Codage source canal conjoint des codes universels Lempel-Ziv** : Dans cette thématique de codage source canal conjoint des codes Lempel-Ziv un seul travail a été développé (à notre connaissance) dans la référence [114]. Les auteurs de cette référence proposent une structuration des données Lempel-Ziv-77 combinée à des codes Reed Solomon de façon à ne pas introduire de redondance dans le flux des codes Lempel-Ziv.

Cette structuration vient de la constatation suivante. Lors de la construction du dictionnaire, l'algorithme de codage Lempel-Ziv cherche la plus longue sous-chaîne dans le texte qui puisse correspondre à la même sous-chaîne disponible dans le dictionnaire. Rappelons que la sous-chaîne retenue est référencée par un pointeur. Les auteurs signalent que plusieurs pointeurs peuvent adresser la même sous-chaîne. De ce fait, ils proposent d'exploiter la multiplicité des pointeurs dans le processus d'amélioration de la robustesse du flux codé. Le pointeur retenu est le pointeur qui contribue à la construction du syndrome de Reed-Solomon.

C. Techniques robustes de décodage source canal conjoint

Si l'objectif de tout codeur de source efficace est de réduire la redondance présente dans le flux d'informations à transmettre, on a vu qu'aucun codeur entropique standard ne savait parfaitement éliminer cette redondance. Ainsi, alors que chaque nouvelle génération de codeur de source, qu'il s'agisse de codeur de parole, de musique, d'image ou de vidéo, progresse toujours vers une meilleure efficacité de compression. Cependant, une part de redondance résiduelle est toujours présente, partiellement inhérente à la volonté de définir des standards généraux applicables à de nombreux types de contenus. C'est à cette redondance résiduelle que s'intéressent de nombreux travaux de décodage conjoint source canal de codes à longueur variable, afin de l'exploiter pour améliorer les performances

ou les services fournis dans le cadre d'une transmission multimédia.

L'idée principale adoptée par les approches de décodage source canal conjoint est d'améliorer les performances de décodage, sans aucune intervention au niveau du codage [48]. Ainsi, le récepteur corrige les erreurs en exploitant les redondances résiduelles, de tous types, laissées par un codeur de source sous optimal dans le cas d'une chaîne de communication séparée. Nous nous intéresserons dans ce qui suit en particulier aux méthodes de décodage souple des CLV de type Huffman et arithmétiques.

1) Principe du décodage souple des codes à longueur variable : De nombreux travaux ont été proposés pour le décodage souple de données codées à l'aide de codes à longueur variable. Les premiers travaux remontent à Massey [128]. De manière générale, les techniques de décodage souple des CLV reposent sur l'estimation statistique de la séquence émise par la source à partir des observations bruitées. Le décodage souple des CLV s'appuie sur (i) le choix de l'estimateur ; (ii) la définition de l'espace de recherche de l'estimateur ; et (iii) la méthode de recherche de la solution.

Un critère de vraisemblance, ou un critère de probabilité a posteriori est généralement utilisé pour optimiser l'estimateur. C'est le critère choisi qui définit la métrique. Celle-ci correspond à une mesure de pertinence des décisions du décodeur.

L'optimisation du critère est réalisée de manière exhaustive sur l'ensemble des décisions possibles du décodeur. Il est donc nécessaire de définir l'ensemble de recherche dans lequel sont exclus les mots de code où les contraintes liées au codage de source ne sont pas satisfaites. Ces contraintes sont déduites à partir de l'identification de la redondance résiduelle contenue dans les CLV, permettant ainsi de détecter et de corriger des erreurs de transmission au niveau du décodeur.

La suite de l'article est organisé comme suit. Nous identifierons tout d'abord les différents types de redondance exploitable dans les CLV. Cette étape est nécessaire pour améliorer les performances de décodage et réduire l'espace de recherche. Des résultats de simulation sont présentés où toutes les redondances sont prises en compte dans le processus de décodage. Nous présenterons ensuite les premiers résultats liés à l'estimation de la séquence source à partir des observations bruitées de la source.

a) Exploitation et classification de la redondance résiduelle des CLV : Pour réduire l'espace de recherche de l'estimateur, il est important d'identifier les différents types de redondance à exploiter [134][200]. Quatre classes différentes de redondance peuvent être distinguées, comme décrit ci-dessous.

– Redondance liée à l'orthographe

On peut exploiter cette forme de redondance lorsque les CLV sont incomplets (i.e. l'inégalité de Kraft est stricte) [165][210]. En effet, il existe un ensemble de mots de code qui ne peuvent être générés par le codeur de source (mots de code non valides).

Supposons que N corresponde au nombre de bits bruités reçus à la sortie d'un canal de transmission. La taille de l'espace de recherche pour le décodage, est égale à 2^N , est réduite à une taille inférieure puisque le code est incomplet. Ce type de redondance identifié au niveau des CLV correspond à une redondance liée à l'orthographe

des mots du CLV.

Prenons un exemple pour illustrer l'intérêt d'exploiter cette redondance. Soit un CLV incomplet de taille 3 défini comme suit $C = \{0, 11, 101\}$. Il est clair que la suite de bits 111011 ne peut pas être générée par le code C .

– *Redondance liée à la paquetsation pour la transmission*

Avant d'être transmises sur un canal de transmission de type IP, les données comprimées sont regroupées dans des paquets IP. La taille de ces paquets est variable (la donnée compressée l'étant généralement), mais l'en-tête est généralement de taille fixe (or champs d'extension). Plusieurs informations peuvent être déduites lors de l'analyse d'un paquet. Deux possibilités d'empaquetage des données à transmettre sont offertes. L'encapsulation des informations contenues dans un paquet est soit dictée par la norme (ex. H.264, NALU), soit n'est pas imposée par la norme (ex H.263). Dans ce dernier cas, l'empaquetage est réalisé de façon à en tirer profit pendant le décodage.

Citons par exemple les travaux proposés par la référence [107] concernant le codage des textures dans H.263+. Des blocs entiers de texture sont placés dans chaque paquet selon des règles bien précises.

Citons le codeur vidéo H.264 où la paquetsation fait partie de la norme. La paquetsation est réalisée de manière efficace (NALU). La fin d'un paquet IP correspond à la fin d'un bloc ou d'un slice. De plus, les paquets sont indépendants pour assurer une meilleure robustesse aux erreurs. Le nombre de bits, par exemple, à l'intérieur de chaque paquet peut être connu du décodeur par le biais de l'en-tête paquet.

– *Redondance liée à la sémantique du code*

Des travaux de recherche exploitant ce type de redondance, ont été développés dans des contextes particuliers tels que H.263 et H.264. En effet, des formes de redondance sémantique peuvent être identifiées.

Citons par exemple les travaux développés dans la référence [107], où les auteurs s'intéressent aux CLV utilisés par la norme H.263+. La redondance liée au codage de la texture issue de la compensation en mouvement de deux images successives est prise en compte. Une transformation de type DCT est appliquée sur chaque bloc de taille 8x8 pixels de l'image. Les coefficients DCT de chaque bloc sont ensuite quantifiés puis lus selon un parcours en zigzag. Les 64 coefficients ainsi lus sont codés selon le triplet suivant : run (pour compter le nombre de zéros avant le coefficient non nul), level (la valeur du coefficient non nul), last (information pour indiquer si ce coefficient appartient au dernier bloc de l'image). Chaque bloc est donc terminé par un symbole particulier. Les mots de code associés à un bloc de texture H.263 vérifient alors certaines règles sémantiques qui sont exploitées pour améliorer les performances du codeur vis à vis des erreurs de transmission. Dans la référence [133], les auteurs estiment la redondance liée à la sémantique encore disponible dans le flux comprimé H.263.

Citons également les travaux proposés par la référence [114] concernant l'amélioration de la robustesse des codes Lempel-Ziv. Les auteurs ne se contentent pas de conserver le premier pointeur faisant référence à la plus longue sous chaîne dans le dictionnaire. Pour la même sous chaîne trouvée, ils exploitent la possibilité de trouver d'autres pointeurs. Si c'est le cas, ils sélectionnent celui qui leur permettra de reconstruire le syndrome de Reed-Solomon.

– *Redondance liée à la source Markovienne*

Une redondance supplémentaire peut être identifiée en considérant la source décrite par un processus de Markov. Cette information supplémentaire permet de construire un treillis à trois dimensions (une dimension correspond à l'état de la source de Markov, une dimension au nombre de bits déjà décodés et une dimension à la position dans le mot courant) pour réaliser le décodage [183]. Cette technique de décodage, comme les précédentes lorsqu'elle fournit des estimées souples, peut être combinée avec un décodeur canal de manière à obtenir un décodeur itératif.

– *Importance de la contribution de toutes les redondances : quelques résultats*

L'un des aspects du travail effectué pendant la thèse de C. M. Lee [108], a été de développer un décodeur de la texture générée par H263+ capable d'exploiter les trois sources de redondance mentionnées précédemment. De plus, l'auteur s'est fixé pour contrainte que ce décodeur soit le plus compatible possible avec la norme, de manière à ne pas avoir à rajouter de redondance artificielle. Ainsi, lorsqu'un paquet contenant des informations de texture bruitée doit être décodé, les seules informations supposées disponibles étant le nombre N_b de bits alloués à la texture dans tout le paquet et le nombre N_B de blocs de texture du paquet.

Il a alors été montré que pour le décodage complet de la texture contenue dans un paquet, l'exploitation simultanée de la redondance liée à l'orthographe, la grammaire et au conditionnement conduit à représenter la succession des mots de CLV d'un paquet de N_b bits contenant N_B blocs de texture par un treillis à quatre dimensions, inspiré de [183]. La complexité de ce treillis, évaluée dans [108], ne permet cependant pas d'envisager le décodage dans des temps réalistes.

Les auteurs de la référence [108] proposent alors de décomposer le problème en deux sous problèmes. Dans un premier temps, il s'agit de localiser les frontières de blocs de texture à l'intérieur d'un paquet. Ceci peut être réalisé à l'aide d'un treillis à deux dimensions, de complexité très réduite par rapport au treillis à quatre dimensions. En effet, il n'exploite que la redondance liée à l'orthographe des mots de CLV ainsi que la redondance liée au conditionnement. Dans un second temps, un treillis à trois dimensions pour le décodage de chacun des blocs de texture préalablement localisés est proposé. Cette formulation permet de mettre clairement en évidence les contraintes liées à la grammaire du codeur de source ainsi qu'à l'orthographe des mots de CLV, ce que ne permettait pas le décodage à base de listes multiples proposé par [134]. Le schéma en deux étapes est sous-optimal, car les blocs localisés lors de la première étape ne satisfont pas nécessairement la contrainte liée à la grammaire du codeur source, mais une évaluation de la complexité montre que le problème est alors bien plus facilement traitable.

Le décodage souple de la texture exploitant la redondance résiduelle a été testé sur des trains binaires générés par le codeur H.263+ [108]. Des paquets d'environ 1000 bits sont envoyés sur un canal gaussien. Par rapport à un décodeur classique, les performances de la solution proposée sont meilleures de 1.5 dB à 2 dB en terme de rapport signal à bruit (SNR pour Signal to Noise Ratio) pour des paquets de type INTRA et de 1 dB pour des paquets de type INTER. Comparée à une technique où les frontières de bloc seraient transmises, à partir de 9 dB de SNR, les performances de la technique proposée sont légèrement inférieures (de 0.1 dB à 0.2 dB) pour des paquets de type INTRA et de moins de 0.5 dB pour des paquets de type INTER, ce qui montre l'efficacité de la technique de la localisation des frontières de blocs. La figure 7 donne le taux d'erreur bloc pour différentes techniques de décodage de la texture : décodeur standard ; décodeur standard tenant compte de la contrainte ; décodeur (ML) souple avec

métrique de type maximum de vraisemblance, contraintes respectées ; décodeur (MAP) souple avec métrique de type maximum a posteriori, contraintes respectées et décodeur (MAP) souple avec métrique de type maximum a posteriori, contraintes respectées et frontières des blocs connues du décodeur.

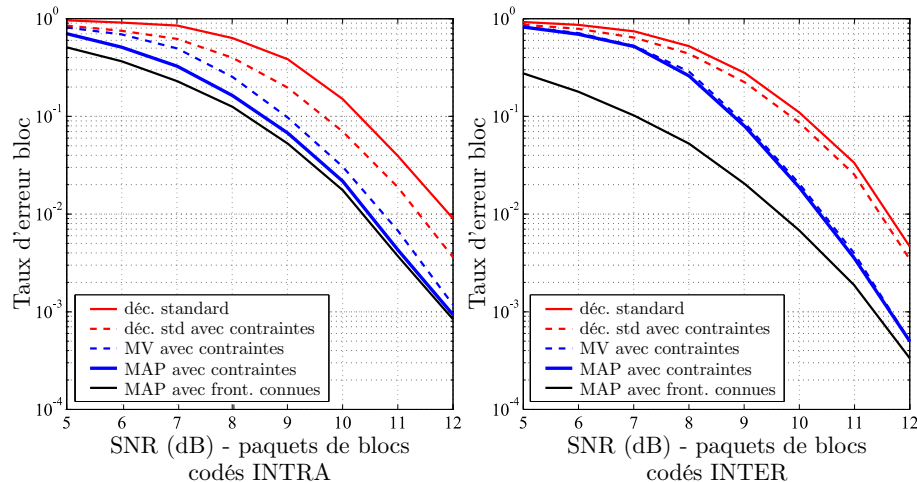


Fig. 7. Taux d'erreur bloc pour différentes techniques de décodage de la texture (source : [108]).

Le gain principal en terme de qualité visuelle est obtenu par la prise en compte des contraintes liées à la mise sous forme de paquets du train binaire et à celles liées à la grammaire (structure en blocs de la texture). Ce gain est illustré par la figure 8, où les valeurs de qualité sont exprimées en terme de Peak Signal to Noise Ratio (PSNR), mesure classique de qualité visuelle objective. La prise en compte des informations souples du canal permet d'améliorer encore les performances. Le décodeur ainsi obtenu est relativement compatible avec la norme H263+. En effet il ne nécessite pas la transmission d'informations annexes autres que celles déjà présentes et nécessaires au décodage en environnement non bruité.

Dans le cas du H.264/AVC, on peut par exemple réaliser un décodage tirant parti de la redondance liée à l'orthographe (détection de mots non valides) et de la redondance liée à la sémantique (détection de suite illogique de mots de code), comme proposé par exemple dans les travaux présentés dans [20][163].

b) Estimation et réduction de l'espace de recherche : Soit $\mathbf{y} = \mathbf{x}_1, \dots, \mathbf{x}_K$ une séquence de K symboles envoyés à l'entrée d'un codeur entropique. Celui-ci génère K codes de longueur variable notés $c = c_1, \dots, c_K$. Ces codes sont transmis à travers un canal bruité. Rappelons que l'objectif du décodeur est d'estimer la séquence émise à partir des observations bruitées $\mathbf{y} = \mathbf{y}_1, \dots, \mathbf{y}_N$, en optimisant un critère sur l'ensemble des séquences de code possibles ou l'ensemble des symboles de code possibles.

Introduisons les deux estimateurs couramment utilisés. Le premier estimateur employé est l'estimateur au sens du maximum a posteriori (MAP). Celui-ci cherche la séquence de plus grande probabilité d'avoir été émise sachant

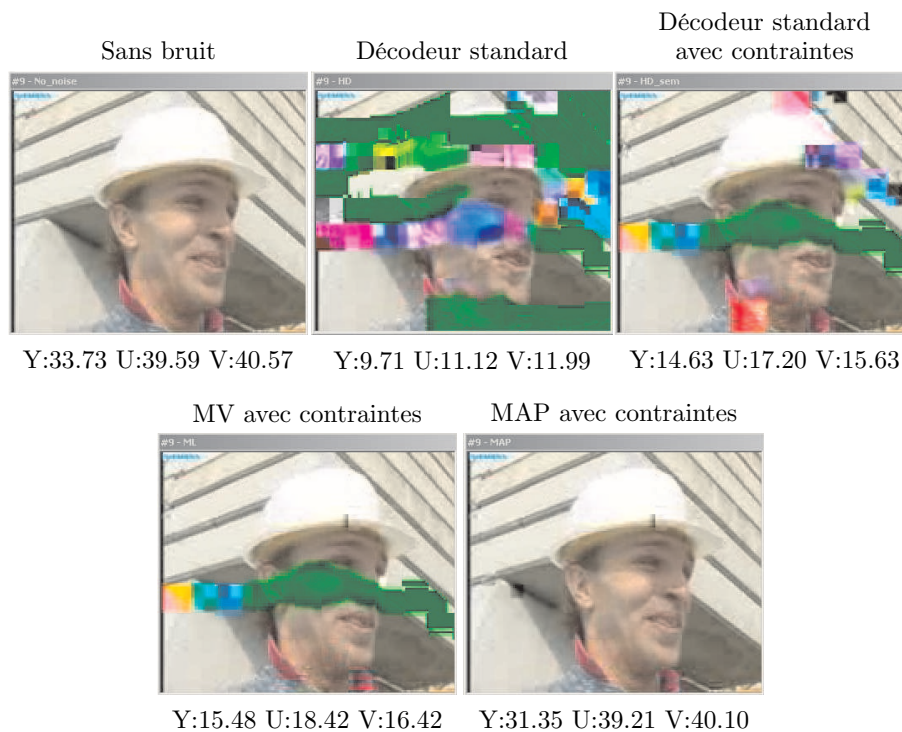


Fig. 8. Comparaison de la qualité de reconstruction de la première image de la séquence (source : [48]).

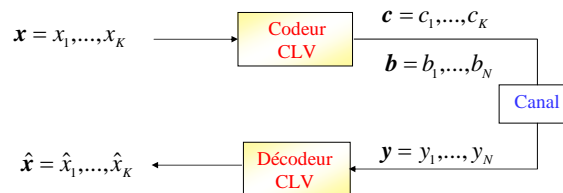


Fig. 9. Schéma classique d'un codeur à longueur variable.

les observations $y = y_1, \dots, y_N$. Il est donné par l'équation suivante :

$$\hat{x}_{1::K} = \arg \max_{x_1, \dots, x_K} \Pr(x_1, \dots, x_K | y_1, \dots, y_N) \quad (1)$$

Le deuxième estimateur concerne l'estimateur MAP symbole par symbole. Ce dernier cherche le symbole x_k de plus grande probabilité d'avoir été émis sachant les observations $y = y_1, \dots, y_N$. Il minimise le taux d'erreur symbole de la séquence. Il est donné par l'équation suivante :

$$\hat{x}_k = \arg \max_{x_k} \Pr(X = x_k | y_1, \dots, y_N) \quad (2)$$

c) *Recherche de la solution* : La solution recherchée est implémentée soit par des algorithmes en treillis qui sont en général complexes mais optimaux ; soit par des algorithmes séquentiels qui sont sous-optimaux et moins complexes [113]. Pour réduire l'espace de recherche, les algorithmes développés tiennent compte des contraintes imposées par la structure du CLV, du nombre de bits du train binaire codé (ou nombre de symboles composant le train binaire) lorsque cette information est disponible. L'ensemble des successions des mots de code satisfaisant ces contraintes sont représentés à l'aide d'un treillis de dimension plus ou moins élevée. Ce treillis peut être exploité par des techniques de décodage de canal standard telles que Viterbi [148][188], SOVA [75] ou BCJR [12] pour réaliser le décodage.

Dans la littérature plusieurs autres types de treillis ont été proposés [16][14][15][104][74]). Ainsi, la référence [14] a proposé un treillis exploitant la structure de l'arbre de Huffman associé au code à longueur variable. A chaque nœud du treillis correspond un nœud de l'arbre de Huffman. A titre d'illustration, un tel treillis construit à partir des 205 mots du CLV utilisé par H263+ pour représenter l'information de texture, nécessite de considérer à chaque instant environ 200 nœuds de chacun desquels partent une ou deux branches joignant les 200 nœuds de l'instant suivant. Un treillis plus compact est proposé par [93]. Dans cette structure, un nœud a représente la fin d'un mot (ou d'une succession de mots) du CLV dont la longueur (cumulée) est de a bits. Les branches reliant directement les nœuds a et $a+\ell$ représentent tous les mots de CLV de ℓ bits. Avec cette structure, la complexité provient du nombre de branches parallèles à gérer. Ainsi, pour les mots CLV utilisés par H263+ pour coder la texture, lorsque $\ell = 10$, 42 branches parallèles relient a et $a + 10$. Le travail proposé par la référence [131] concerne la simplification des tables de CLV de manière à alléger la structure des treillis utilisés pour le décodage. Schématiquement des mots de CLV de même longueur sont groupés en classes. Par exemple, lorsqu'on considère 16 mots de code à longueur variable de 8 bits de la norme H263+, $A_8 = \{00100000, 00100001, 00101110, 00101111\}$ on peut remarquer que tous partagent le même préfixe de 4 bits 0010, et que les autres bits prennent toutes les valeurs possibles. Ainsi, cet ensemble de mots de code à longueur variable peut être représenté par la classe unique 0010\$\$\$\$, où \$ représente soit 0 soit 1.

Avec cette simplification, 16 branches parallèles entre a et $a + 8$ dans un treillis tel que celui présenté dans [93] sont remplacées par une branche unique. Lorsque la probabilité a priori de chaque mot de CLV ainsi groupé est la même ou lorsqu'un décodage de type maximum de vraisemblance est réalisé, cette simplification n'a pas de conséquence sur les performances du décodeur. Lorsque des différences entre les probabilités a priori des mots regroupés sont significatives, un estimateur au sens du maximum a posteriori est sous-optimal avec une table de mots de code à longueur variable simplifiée. Dans tous les cas, des techniques de décodage inspirées des codes convolutifs poinçonnés [91] peuvent alors être mises en œuvre. L'optimalité de l'algorithme proposé est prouvée. Les mots d'un code à longueur variable sont regroupés en un minimum de classes. Les gains en terme de complexité pour différents décodeurs ont été évalués théoriquement. Les travaux cherchant à appliquer des techniques de décodage

de canal aux codes à longueur variable de type Huffman ont naturellement été transposés dans le contexte des codes de type arithmétique.

2) **Décodage conjoint source canal de CLV** : Les normes multimédia les plus récentes ont su intégrer des codeurs entropiques de plus en plus performants, qu'il s'agisse de codes à longueur variable spécifiques comme le CAVLC de H.264 AVC/MPEG-4 Part 10 ou de différentes configurations de codeur arithmétique comme pour JPEG2000 et le mode CABAC de H.264 AVC/MPEG-4 Part 10. L'intérêt de ces codeurs entropiques est leurs bonnes performances en terme de compression, mais il n'en reste pas moins qu'il n'y a pas d'association directe entre symboles de la source et succession de bits générée par le codeur, ce qui rend le flux binaire particulièrement vulnérable aux erreurs de transmission.

En ce qui concerne le décodage conjoint de codes à longueur variables traditionnels intégrés dans les standards vidéo, différents travaux ont montré qu'il était possible d'exploiter la redondance existante dans le flux binaire sans ajouter de redondance supplémentaire ou de marqueur quelconque. L'intérêt de ces approches est donc double. Elles peuvent d'une part permettre de gagner en qualité finale, et d'autre part sont applicables sans aucune modification au niveau du codeur vidéo et de l'émission. On trouve par exemple ces travaux appliqués au standard MPEG-4 Part 2 dans [143] ou à H.264 AVC/MPEG-4 Part 10 dans [20].

En ce qui concerne les codes arithmétiques, une majorité des travaux de décodage source canal conjoint ont exploité un symbole interdit [33] ou des marqueurs de synchronisation [70] pour la détection d'erreurs de transmission au niveau du récepteur. Plusieurs techniques ont alors été développées pour corriger ces erreurs. Ces techniques sont pour la plupart basées sur les approches utilisées en codage de canal comme les algorithmes de décodage séquentiel [7] ou encore l'ARQ (Automatic Repeat Request) [33][67][80].

Un schéma de décodage de source canal conjoint intégrant de l'ARQ est présenté dans la référence [33]. Un symbole interdit est introduit dans le train de symbole de source. La méthode proposée permet la détection d'erreurs au fur et à mesure de la réception. Cette stratégie diffère de celle adoptée par les techniques classiques avec ARQ qui exigent la réception de tout le paquet avant de détecter une erreur éventuelle. De plus, dans le cas où le décodeur arithmétique détecte une erreur en décodant un SI, on peut choisir de ne pas retransmettre tout le paquet, mais seulement les n derniers bits pour une probabilité d'erreur préalablement fixée.

L'auteur de la référence [164] montre qu'en jouant sur l'emplacement du SI dans l'espace de probabilités, et en modifiant les modèles de probabilités des symboles de la source, le codeur arithmétique n'est autre qu'une généralisation des codes convolutifs et des codes blocs.

Pettijohn et al [80] exploitent la redondance introduite par un SI par le biais d'un décodeur séquentiel prenant à la fois des décisions dures et souples sur les observations bruitées. Un arbre binaire de décodage est généré. Les nœuds sont construits uniquement si l'observation se trouve à l'intérieur d'une zone de doute (appelée également null zone). En-dehors de cette zone, un 0 est assignée aux observations négatives, et un 1 aux observations positives. L'arbre de décodage est exploré en profondeur. Le survivant correspond au chemin maximisant la vraisemblance. Si le décodeur détecte un SI, la branche courante est élaguée, et l'exploration de l'arbre de décodage reprend à

partir du dernier nœud. Si le décodeur continue à décoder des SI, la zone de doute est élargie pour faire apparaître plus de nœuds sur l'arbre de décodage.

L'approche de Pettijohn et al. décrite ci-dessus a été couplée à un décodage dur de manière itérative. La méthode est présentée par la référence [208]. Citons également l'approche qui consiste à appliquer une modulation TCM (pour modulation codée en treillis) à la sortie du codeur arithmétique de manière à disposer d'une structure de treillis pour le décodage [41]. Cette méthode repose également sur un SI pour la détection des erreurs.

Les auteurs des références [66][69][67] proposent un décodage séquentiel. Celui-ci repose sur un symbole interdit qui permet la détection d'erreurs. Le décodage séquentiel s'appuie sur des piles contenant à chaque instant un nombre limité de trains binaires. Ces derniers peuvent correspondre au code émis par le codeur et sont ordonnés selon une métrique MAP. Les algorithmes généralement utilisés sont le Stack Algorithm (SA) et le M-Algorithm (MA) [7][113]. Cette technique a été appliquée au décodage de vidéos codées par JPEG2000. Plus de détails sont fournis dans la référence [68].

Une autre approche consiste à considérer des codes quasi-arithmétiques [71] afin de limiter le nombre de symboles possibles et donc la complexité dans le cadre d'un décodage MAP. L'un des intérêts d'une telle approche est qu'elle permet de réaliser des décodages itératifs entre un code quasi-arithmétique et un code correcteur concaténés.

Dongsheng et al. proposent une machine à états finis pour représenter le codage arithmétique binaire à valeurs entières utilisant un SI [46]. Un treillis à trois dimensions (nombre de bits, les états, profondeur) est construit. La dimension associée à la profondeur, décrit les états qui diffèrent par leurs paramètres internes non observables au niveau des entrées et des sorties du codeur. Ce treillis est exploité à la réception par le biais d'un décodeur Viterbi à sorties dures.

La référence [125] présente des travaux sur le CABAC dans un contexte de transmission de données multimédia (vidéo) sur un canal de type IP suivi d'un canal radio mobile. La transmission se fait par paquets qui peuvent arriver au destinataire entachés d'un bruit introduit par la partie radio-mobile du canal.

La référence [86] propose un schéma où le décodage est réalisé par un décodage séquentiel au moyen d'une pile ordonnée [212]. Ce schéma de décodage classique se révèle bien adapté à la gestion des contextes du CABAC. L'une des contributions du travail proposé par [86] est d'utiliser efficacement le modèle du canal de transmission de manière à bâtir des tests d'hypothèses permettant d'ajuster de manière objective la complexité du décodage en fonction des performances souhaitées.

Des résultats intéressants ont été proposés dans le cadre du CABAC dans les références [86][88][87][89][163]. L'objectif a été d'exploiter la redondance existante au niveau du codeur arithmétique et au niveau du conditionnement sous forme de paquets des données codées, sans ajouter de redondance supplémentaire. Pour identifier ces deux sources de redondance, il s'agit d'examiner de plus près le fonctionnement du CABAC. Ce codeur arithmétique permet de coder des données binaires selon des modèles de probabilités adaptatifs dépendant des contextes. Les quantités générées par la source ne sont pas toujours binaires, une binarisation des données est donc nécessaire. Différents schémas de binarisation peuvent être envisagés. Certains schémas de binarisation sont simplement des codes à longueur variable incomplets. De ce fait, la binarisation introduit une redondance qui peut être exploitée

pour réaliser le décodage. Le choix du contexte est essentiellement lié au type de données à coder. L'adaptation des probabilités est obtenue au fur et à mesure du codage. Ainsi, le CABAC repose sur différentes composantes : la binarisation, les contextes, l'adaptation des modèles de probabilités de chaque contexte et le codeur arithmétique binaire proprement dit.

A notre connaissance, il n'existe pas de techniques de décodage source canal conjoint pour les codes universels de type Lempel-Ziv.

IV. Techniques de diversité adaptées à la transmission IP sans fil point/multipoint

Afin de garantir une certaine qualité de service, illustrée notamment par un taux d'erreur maximal, il est souhaitable de minimiser la durée et la fréquence des évanouissements du signal arrivant sur l'antenne du récepteur. Ainsi, si à un instant donné, le récepteur reçoit L copies indépendantes du signal original émis, la probabilité pour que tous les L trajets soient affaiblis simultanément est égale à p^L , où p est la probabilité pour qu'un seul chemin soit affaibli. Employant des répétitions ou une redondance d'information, la diversité s'appuie sur la méthode avec laquelle ces L copies indépendantes et sans corrélation sont fournies au récepteur.

Il existe essentiellement trois types de diversité, plus ou moins bien adaptés selon le modèle du canal de propagation à compenser. A savoir :

- La diversité temporelle, où le signal à transmettre est répété à des intervalles réguliers, séparés par un espace de temps au moins égal au temps de cohérence du canal.
- La diversité fréquentielle, pour laquelle la même information est transmise sur différentes porteuses, séparées par une bande de fréquence au moins égale à la bande de cohérence du canal.
- La diversité spatiale, avec plusieurs antennes de réception et/ou d'émission disposées en des endroits différents, d'où la réception de signaux différents, si possible non corrélés.

A ces principes de diversité, s'ajoutent la diversité de polarisation et la diversité d'angle, plus spécifiques de la géométrie et de la disposition des antennes. La diversité d'antenne ou spatiale semble bien plus avantageuse que les deux premières en terme de bande passante ou de débit et de latence. Cependant un tel choix induit une complexité accrue du système d'antenne. Côté réception, la difficulté se situe principalement dans la recombinaison des L signaux délivrés par les L antennes, pour former un signal composite exploitable par le récepteur.

La diversité spatiale sera associée soit à une combinaison des signaux après détection, soit à une combinaison avant détection, plus économique. Dans tous les cas le bénéfice en terme de gain que procure la diversité spatiale est conséquent. Des bornes théoriques ont été établies, comme dans [81] qui permettent de déterminer des compromis entre la distorsion totale et le nombre d'antennes utilisées ainsi que de prendre éventuellement en compte la valeur du délai introduit par un processus d'ARQ.

La première méthode, beaucoup plus lourde tant en matériel qu'en traitement du signal, nécessite autant de récepteurs que d'antennes. Nous présentons ces méthodes de combinaison, de la plus simple à la plus compliquée [62].

- La combinaison passive : une ou plusieurs antennes redondantes sont placées à une distance minimale de $\lambda/2$ de la première, avec λ la longueur d'onde de la porteuse). Ce dispositif élève le niveau du signal à l'entrée du récepteur, sachant que la probabilité que les antennes soient simultanément sujettes à l'évanouissement du canal est réduite. Cependant, les signaux reçus sur ces antennes peuvent être en opposition de phase et leur combinaison destructive.
- La combinaison sélective : similaire à la précédente, mais une mesure du niveau RF permet la sélection d'une des antennes soit en fonction d'un seuil prédéterminé, soit de celle qui dispose du plus fort rapport signal sur bruit.
- La combinaison à égalité de gain : dans cette configuration, chacune des L antennes, dispose de son récepteur. La phase φ de chaque branche est estimée pour la mise en phase des signaux reçus et pour obtenir une combinaison linéaire optimale. La même pondération est appliquée sur les amplitudes des signaux de chaque branche.
- La combinaison à rapport maximal (*Maximum Ratio Combining*) : cette configuration est identique à la précédente mais une estimation de l'atténuation A_i de chaque branche est réalisée. Chaque branche est pondérée en fonction du niveau des signaux reçus de façon à favoriser les signaux de fortes amplitudes dans leur contribution au signal composite.

Symétriquement, la diversité s'applique aussi côté émission, comme dans le cas du standard WiFi IEEE802.11n [4], et ce grâce à la transmission simultanée de plusieurs flux de données à travers autant d'antennes, et à l'utilisation d'algorithmes spécifiques de démultiplexage et de codage des données. En fonction des traitements réalisés de part et d'autre du lien de communication, côté émission et/ou réception, la configuration de diversité est définie comme du type MISO (*multiple inputs, single output*), SIMO (*single input, multiple outputs*), ou encore MIMO (*multiple inputs, multiple outputs*). Véritables systèmes d'antennes intelligentes, les dispositifs MIMO exploitent autant les bénéfices de la diversité de réception que le gain de multiplexage spatial à l'émission.

La transmission de données vidéo dans un environnement IP sans fil est actuellement en plein essor mais s'avère souvent difficile à modéliser du fait de la grande variabilité des liens considérés. Elle mélange en effet d'une part (i) une transmission sur une partie filaire reposant sur le protocole de niveau 3 IP et propice aux erreurs de paquets liées à des pertes par congestion, et d'autre part (ii) une partie sans fil (canal de type UMTS ou WLAN), propice aux erreurs liées à des évanouissements et à du bruit sur le canal. Lorsque s'ajoute au problème le déploiement de techniques de diffusion vers multipoint telles que celles rendues populaires par les technologies xDSL (pour transmission sur réseau de paires torsadées) on s'aperçoit de la difficulté d'optimiser les transmissions puisqu'il s'agit :

- d'une part de décliner des protocoles bien souvent conçus pour des solutions filaires uniquement, donc conçus pour travailler à des taux d'erreur et pour des débits utiles bien différents de ceux fournis par les canaux sans fil ou hybrides fil/sans fil,
- d'autre part de réaliser une optimisation pour un modèle de canal de transmission connu presque uniquement par parties (filaire d'un côté, sans fil de l'autre), avec des contraintes différentes pour chacune des dites parties.

La solution classique est souvent de se régler sur un mode de fonctionnement imitant le filaire, en abusant des retransmissions (ARQ) et/ou en surprotégeant les données à émettre. De manière générale, on utilise donc une forte partie de la bande passante utile pour émettre des bits « peu utiles ». Le problème est donc de savoir comment protéger de façon efficace, en tenant compte des différentes couches du modèle OSI, des différences entre les parties de la chaîne de transmission, des particularités du flux vidéo que l'on souhaite transmettre et des limitations ou avantages propres aux techniques multipoints. En effet, la structure d'un flux vidéo (trames, niveaux de raffinements différents pour une vidéo scalable, ...) permet de définir des niveaux d'importance différents mais aussi des points de resynchronisation dans le flux.

A. Structures de réseau

La plupart des réseaux actuels se rangent en deux principales catégories : les réseaux centralisés, dont la topologie est figée autour d'un serveur central, et les réseaux mobiles ad-hoc. Ces deux types de réseaux ne sont pas sujets aux mêmes défaillances.

1) **Réseau centralisé** : La topologie d'un réseau à architecture centralisée est relativement stable et facilement prévisible. Typiquement, le réseau de distribution comporte un serveur qui héberge la totalité du contenu. Les inconvénients d'une telle structure sont nombreux. En particulier, la bande passante est globalement sous-exploitée, tout en étant sur-exploitée sur le lien du serveur, et la robustesse du réseau repose sur la fiabilité et la puissance du serveur. Il s'agit cependant d'une structure courante sur Internet.

Internet est l'exemple type du réseau à pertes de paquets. L'une des causes des pertes de paquets sur Internet est la congestion. La stratégie la plus courante consiste à réémettre les paquets perdus, ce qui n'est pas toujours possible, ni même souhaitable, en raison des délais supplémentaires induits et de la surcharge du réseau que cela occasionne.

D'autre part, la charge du réseau dépend fortement du jour et de l'heure. C'est une des raisons principales pour laquelle le codage de type FEC sans adaptation fonctionne assez mal. Enfin, en cas de congestion, les routeurs ne tiennent pas compte d'une éventuelle priorité des paquets avant de les jeter. Ces conditions placent le codage par MDC, avec des descriptions équilibrées, en position avantageuse.

2) **Réseau mobile ad-hoc** : Il s'agit de réseaux auto-configurés d'utilisateurs mobiles connectés par liens sans fil. La topologie de ces réseaux est très difficile à prévoir puisque les utilisateurs se réorganisent de façon aléatoire. De plus, ces réseaux sont très propices aux erreurs. Les liens peuvent en effet tomber pour cause d'interférences, d'atténuation du canal, de mobilité des nœuds ou du fait de l'insuffisance de l'infrastructure.

Ces réseaux constituent un défi pour les applications de diffusion (type streaming) de contenu multimédia. En effet, la livraison en continu des données implique la connexion continue entre le client et le ou les serveur(s). De plus, on souhaite comme dans le cas des réseaux classiques que la qualité de la vidéo décodée se dégrade

continûment avec les conditions de la connexion réseau. Cependant, l'infrastructure de ces réseaux permettant la diversité des chemins, semble bien adaptée à l'utilisation de schémas de codage par descriptions multiples.

Dans [50], on présente l'étude d'un MDC combiné avec un routage multi-chemins s'appuyant sur les protocoles IEEE 802.11 et Dynamic Source Routing. On y compare les performances du routage multi-chemins avec MDC à celles du routage simple-chemin. Dans les scénarios mobiles, le routage simple-chemin est plus performant. Le MDC permet de soulager certaines parties du réseau en cas de congestion. Il présente alors de meilleures performances lorsque les chemins sont choisis sous certaines conditions (chemins disjoints dans lesquels le chemin alternatif n'est pas affecté par la congestion).

Dans [123], on propose de combiner le MDC avec le transport multi-chemins, et on montre que la diversité des chemins améliore la résistance aux erreurs de transmission sur un réseau ad-hoc. Dans [120][121], on propose un schéma MDC multicast pour la vidéo qui utilise des arbres multiples pour améliorer la robustesse. Enfin, dans [122], on étudie la sélection conjointe du routage et des serveurs pour le MDC de vidéos sur réseau ad-hoc.

B. Stratégies de distribution

1) **Les réseaux de diffusion de contenu (CDN)** : Les réseaux de diffusion de contenu (CDN pour content delivery networks) sont également des réseaux à pertes de paquets [112]. Il s'agit d'un ensemble de serveurs interconnectés et déployés dans divers endroits. Chaque serveur héberge une copie du contenu à diffuser, ce qui permet de contourner les problèmes de surcharge des serveurs ou de congestion du réseau. A titre d'exemple, on peut citer : Akamai (www.akamai.com), Peer 1 Rapid Edge (peer1.com), Amazon S3 (<http://aws.amazon.com/s3>).

L'architecture de ces CDN présente de nombreux avantages par rapport aux réseaux classiques [150] :

- ils permettent de contourner les problèmes de congestion et de surcharge des serveurs ;
- ils permettent de minimiser la latence, en choisissant le serveur le plus proche de chaque client ;
- en cas de défaillance d'un lien ou d'un serveur, un autre lien ou serveur peut prendre le relais (path diversity).

Les inconvénients de cette approche sont le coût d'un espace de sauvegarde très important, la nécessité d'une fiabilité élevée et les gros besoins en bande passante.

Dans le cas d'une transmission MDC, les problèmes ouverts sont les suivants :

- comment bien répartir les descriptions sur les serveurs [8] ?
- pour un client donné, comment choisir les meilleurs serveurs, c'est-à-dire les plus proches du client, ou ceux qui permettent de diversifier les chemins au maximum, tout en hébergeant des descriptions complémentaires ?

Dans [8][9], on montre qu'un schéma MDC est plus performant qu'un schéma sans descriptions multiples dans les réseaux CDN existants, même sans optimiser la position des serveurs.

Un système de diffusion (streaming) vidéo dans le cas de serveurs multiples est présenté dans [118], fondé sur un schéma de codage MDC-FEC.

Dans [116][26], on utilise les codes Tornado pour accélérer les téléchargements à partir de plusieurs sites miroirs. Dans [135] est présenté un algorithme d'allocation de débit qui minimise le nombre de paquets perdus, dans l'hypothèse où ils sont indépendants et identiquement distribués. Dans [136] est introduit un algorithme d'allocation

amélioré, qui permet non seulement de déterminer le débit de chaque serveur et qui également de définir une partition en paquets permettant de s'assurer qu'aucun paquet n'est envoyé par plus d'un miroir.

Dans [17] on considère un algorithme de programmation d'envoi des paquets qui optimise le compromis débit-distorsion du point de vue du client. Cet algorithme détermine quel serveur envoie quels paquets. La solution est fournie pour deux serveurs, et dans le cas où les descriptions multiples sont obtenues en répétant simplement un flux à description simple.

2) **Réseaux de pairs (P2P)** : Les réseaux P2P (pour Peer to peer) sont des systèmes décentralisés dont les nœuds ont tous des rôles similaires. L'idée est d'utiliser des ordinateurs ordinaires pour stocker des vidéos compressées, dans un contexte de diffusion P2P. Chaque ordinateur de ce type doit garder une description (au sens du codage par descriptions multiples) de la vidéo originale. Le client va ainsi demander à plusieurs pairs leurs descriptions, qui seront décodées et combinées pour le rendu final de la vidéo.

Cette solution présente plusieurs avantages :

- elle est très robuste aux défaillances des serveurs (qui sont des pairs), car une seule description a un impact limité sur la qualité finale de reconstruction et la même description pourrait être stockée par plusieurs pairs ;
- elle réduit la charge de chaque nœud, car une vidéo est répartie entre les différents pairs ;
- elle permet l'utilisation d'une bande passante montante réduite pour chaque nœud, chaque description ayant un débit limité par rapport à celui de la vidéo entière ;
- elle permet une sécurité plus grande du point de vue du fournisseur de services, car aucun utilisateur n'a directement accès à la vidéo complète.

Le codage vidéo MDC est intéressant dans ce cadre, puisqu'il permet à chaque client participant de stocker non pas la totalité, mais une partie seulement du flux compressé.

Dans [6], on propose un système de diffusion adaptatif sur P2P, avec un schéma MDC qui adapte à la volée le nombre de descriptions de base et de raffinement, ainsi que le débit et le taux de redondance de chaque description. Le schéma proposé inclut également une solution de contrôle de flux pour la diffusion vidéo. Le problème de la congestion des réseaux P2P est également un point d'étude important, traité par exemple dans la thèse de E. Setton [168].

Une intéressante comparaison entre la diffusion vidéo par descriptions multiples sur réseaux P2P et sur réseaux CDN est effectuée dans [95]. On y montre que le taux de perte de paquets, le nombre de trames non décodables (à cause d'un délai trop important) et le temps de réponse sont nettement inférieurs dans le cas des réseaux P2P. Les descriptions utilisées sont obtenues par la simple séparation en trames paires et impaires, mais il n'y a pas de processus de correction en cas de perte d'une description (simplement, attente de la prochaine trame décodable).

3) **Solutions hybrides CDN/P2P** : Il existe également des solutions qui combinent le P2P et l'approche par distribution de contenus. Dans CoopNet [139], le téléchargement par P2P est activé lorsque le serveur est surchargé. Celui-ci redirige alors les nouveaux clients vers ceux ayant déjà téléchargé le contenu.

Le problème de la diffusion (streaming) à la demande est traité dans [79]. Le scénario envisagé est celui d'un client obtenant du contenu simultanément à partir de plusieurs pairs. L'approche proposée permet d'attribuer différents segments à différents pairs, et de définir la quantité de codage de canal à assigner à chaque chemin, mais ne dit rien sur la façon dont les différentes parties du contenu sont créées.

Dans [209], le scénario décrit une transmission sans perte de plusieurs descriptions générées à l'aide de codes correcteurs d'erreurs FEC, chacune stockée sur un pair. Les auteurs montrent que le nombre optimal de descriptions s'accroît avec le temps de remplacement d'un pair (consécutif à sa déconnexion).

Dans [176], on forme N descriptions streamées par différentes applications multicast d'un réseau P2P. Chaque pair reçoit $n < N$ descriptions en fonction de sa bande passante. L'optimisation de la qualité de la vidéo reçue est réalisée en tenant compte de la bande passante de chaque client.

Une autre architecture de diffusion (streaming) vidéo sur réseau P2P s'appuyant sur un codage MDC est proposée en [147].

C. Vers une architecture adaptée à la diversité

Un des objectifs des travaux de recherche actuels visant à permettre la diffusion de flux HTML ou multimédia dans un contexte d'utilisateurs multiples est tout d'abord de définir les gains potentiels des architectures multipoints comparées à des solutions point-à-point, mais aussi l'intérêt de solutions scalables (vidéo SVC (pour Scalable Video Coding) ou MDC) et leur adaptabilité dans un contexte multipoints. Dans ce sens, toute approche innovante devra se comparer à des solutions déjà très prometteuses comme celle proposée par Schierl et al. [167] où les aspects de diversité de source, de codage efficace et de distribution pair-à-pair ont été considérés.

Dans la littérature, certains auteurs ont comparé les performances des solutions MDC aux solutions que l'on peut voir comme concurrentes. En voici un aperçu.

1) Le MDC face au codage scalable : Le MDC et le codage scalable, ou par couches (Layered Coding, LC), sont deux techniques qui génèrent plusieurs représentations de la source. Dans le cas du codage scalable, une couche de base (de très grande importance) est produite, et d'autres couches d'amélioration permettent de s'adapter au mieux aux capacités du client en terme de résolution spatiale, de fréquence et de débit. Si la couche de base est corrompue ou perdue, les couches d'amélioration sont inexploitable, même si elles sont reçues correctement. Il est donc primordial que la couche de base soit bien reçue. En pratique, on utilise des stratégies d'ARQ [124] et/ou des codes correcteurs d'erreurs [60].

Les descriptions d'un schéma MDC sont, au contraire, indépendantes, complémentaires et d'importance égale dans la plupart des cas. Un schéma MDC, par ailleurs, ne nécessite pas la prise en charge de transmission prioritaire.

De nombreuses comparaisons entre MDC et codage scalable ont été menées. Il apparaît que les résultats dépendent beaucoup des implémentations, des caractéristiques du canal, des mécanismes de transmission et éventuellement de retransmission . . .

Dans [159], on compare un codeur scalable protégé par FEC avec un schéma MDC sur un canal binaire symétrique et sur un canal à effacement aléatoire. Il apparaît que le MDC ne surpasse le codeur scalable que lorsque les conditions de transmission sont réellement mauvaises. Dans [158], on compare un codeur MDC semblable à [157] et le codec H.263 sur un réseau sans fil EGPRS (Enhanced General Packet Radio Services). Il est observé que le codeur scalable n'est pas moins bon que le codeur MDC. Les auteurs montrent également que dans le cas de deux canaux sans fil, le simple fait de répartir un flux mono-couche entre les deux canaux permet d'obtenir de meilleures performances que le MDC et que le codeur scalable ! Cependant, les auteurs se sont autorisés un délai de décodage de 3 secondes, ce qui rend possible la retransmission de la couche de base.

Dans [137], on montre que le codage scalable peut se montrer plus performant que le codage MDC si l'on utilise une allocation soigneuse des paquets entre les deux chemins.

Dans [172], on compare un codeur JPEG MD par transformée polyphase et un codeur JPEG progressif. Dans le cas où la retransmission de la couche de base est impossible, on montre que le MDC est toujours meilleur. Si les deux flux (MDC et scalable) sont transmis via UDP, le schéma MDC est meilleur dès que le taux de perte de paquets dépasse 2%. Dans le cas où la couche de base peut être retransmise via TCP de façon fiable, le MDC n'est avantageux que si le temps mis par les paquets à aller puis revenir de l'émetteur au récepteur (RTT pour Round-Trip delay Time) est long. Des conclusions identiques ont été tirées dans [194] où l'on compare le codage vidéo MD compensé en mouvement [191] et le codage scalable (en débit) H.263+ sur un réseau sans fil à sauts multiples. De même dans [111], où l'on compare un codage MPEG-2 dans un schéma MDC par quantification et un codeur scalable 3D-SPIHT [96].

Dans [211], on montre que le MDC est meilleur lorsque le taux de perte de paquets est plus important et les délais plus longs, alors que le codage scalable est plus adapté à de faibles taux de pertes et des délais raisonnables. On constate également que le MDC combiné à de l'ARQ donne de meilleures performances. Dans [28], on compare deux schémas MDC par splitting et un schéma scalable, et on montre que le schéma scalable donne de meilleures performances lorsqu'on peut optimiser le calendrier d'envoi des paquets au sens débit-distorsion.

Il est donc difficile de conclure quant à la supériorité de l'un des schémas. En général, le MDC a l'avantage lorsque les contraintes de délais sont importantes, ou lorsqu'il n'y a pas de feedback, ou lorsque le RTT est trop long. Si le réseau et les applications supportent la transmission prioritaire ou le contrôle d'erreurs, alors le codage scalable peut donner de meilleurs résultats.

2) **Le MDC face aux codes correcteurs d'erreurs** : Au lieu de faire en sorte que le codeur produise directement plusieurs descriptions, il est également possible d'utiliser des codes correcteurs d'erreurs sur différentes partitions d'un flux scalable [149][132].

Les approches par FEC ajoutent de la redondance inter-paquets (codes Tornado, fontaine, Reed-Solomon) et peuvent également entrelacer les paquets pour limiter l'impact des pertes en rafale. Le gros avantage de cette technique est la possibilité de généraliser facilement le problème à un nombre arbitraire de descriptions à partir de n'importe quel flux vidéo scalable. Cependant, une augmentation du nombre de descriptions provoque une

augmentation du délai de décodage. De plus, le taux de redondance doit être fixé à l'avance. Selon le taux d'erreurs, la redondance peut être insuffisante ou au contraire superflue. L'utilisation des codes FEC est donc délicate dans un environnement variable tel que celui des réseaux sans fil.

Il existe de nombreuses variantes combinant FEC, ARQ, UEP et MDC, chaque approche ayant ses avantages et ses inconvénients [187][202][197][180][179]. Parmi les codes les plus prometteurs récemment introduits dans la littérature, on trouve la grande famille des codes fontaines dont la particularité est d'être sans rendement (rateless) c'est à dire pour lesquels le taux de codage est variable selon le besoin de l'application ou de l'utilisateur. Font partie de cette famille des codes fontaine [117], dont notamment les codes LT (Luby Transform) [115] et les codes Raptor [171] et les codes en-ligne [130] qui sont des codes Raptor non systématiques.

L'UEP est une solution bien adaptée aux sources encodées de façon progressive (scalable). La question est alors de définir quelles portions du flux doivent être protégées pour un débit donné [155], [101].

La combinaison du codage scalable avec l'UEP est étudiée dans [34][178]. Dans [174], on étudie la génération de codes scalables à descriptions multiples pour des applications de multicast et de broadcast dans des réseaux à pertes de paquets. Dans [177] on construit des descriptions multiples scalables asymétriques à l'aide de codes Reed-Solomon.

La simplicité des FEC ainsi que leurs bonnes performances a conduit à leur utilisation massive, dans le cas où le taux de perte est connu de façon précise. Sinon, le MDC reprend l'avantage.

Dans [65][64][31], on utilise un schéma MDC à base de trames harmoniques appliqué à des coefficients DCT ou d'ondelettes. Les auteurs concluent à la supériorité du MDC, qui obtient des distorsions moins grandes pour un nombre différent de descriptions reçues. De plus, le schéma MDC fondé sur les trames s'avère plus robuste aux conditions fluctuantes du réseau car il ne souffre pas de l'effet de « chute brutale » (*cliff-effect*), qui peut être très important dans certains scénarios très réalistes.

V. Conclusions et perspectives

Le domaine des techniques existantes permettant d'améliorer la qualité d'une transmission de données HTML et vidéo sur un lien de type IP sans fil est particulièrement riche. On y trouve des approches diverses et souvent complémentaires, qui ont pour objet soit d'améliorer la transmission du signal par ajout de diversité (chemin, antennes, ...) sur un canal propice aux erreurs, soit d'améliorer la reconstruction en réception du signal dégradé par la transmission. Différents axes restent néanmoins à approfondir, parmi lesquels on peut citer la gestion d'effacements et non plus seulement d'erreurs par le décodage souple (HTML ou vidéo), l'ajout de techniques de retransmissions y compris pour des flux de type multicast, ou l'amélioration de techniques MIMO.

Il reste également à combiner et à comparer ces différentes techniques présentées, et ce en particulier dans des cas plus pratiques et des canaux moins « théoriques ». Explorer différentes solutions de diversité et de codage conjoint ainsi que leur combinaison en conditions réelles, afin d'offrir une plus grande adaptabilité des applications et services de téléchargement et diffusion dans des conditions d'utilisation, est l'un des défis à relever pour faire enfin migrer ces différentes techniques d'amélioration de robustesse et de diversité du monde de la recherche à un

monde plus pratique. Ainsi, se plaçant dans le cas d'une transmission sur une pile radio 802.11n, standard qui intègre l'usage d'antennes multiples, on pourra s'intéresser aux performances des algorithmes de décodage robuste et ce notamment en modifiant de manière adéquate, via un mécanisme trans-couches, les différentes couches protocolaires pour permettre la remontée d'informations des couches basses, telles que des paquets contenant des informations erronées, voire les informations souples correspondantes. Egalement, en se plaçant dans le cas plus particulier dans le cas des transmissions de point à multipoints, le problème du développement de solutions adaptées par insertion de diversité de codage (via notamment des codes Fontaine) dans un routage ad hoc de type pair à pair et de la comparaison d'une telle approche avec l'emploi de la diversité par codage à descriptions multiples. Ces travaux pourront en particulier mettre en avant les intérêts ou inconvénients d'employer une séparation de l'information sous forme de codage scalable (introduisant des dépendances entre les couches) ou sous forme de codage par descriptions multiples (sans dépendance entre les descriptions mais moins efficace en rendement de compression). Enfin de manière plus générale, il sera intéressant de se pencher sur le problème de la modélisation théorique de l'ajout de protection et de diversité selon les variations de la source à transmettre, du canal de transmission et des demandes spécifiques de l'utilisateur ou de l'application considérée.

VI. Remerciements

Les auteurs remercient Messieurs Pierre Duhamel et Michel Kieffer, du laboratoire LSS pour leur ample documentation et leur disponibilité ainsi que Madame Roxana Ojeda de la société COMSIS pour ses informations concernant la norme 802.11n. Les auteurs tiennent également à remercier les relecteurs anonymes dont les commentaires et remarques détaillées ont permis d'améliorer la clarté et l'organisation de cet article.

REFERENCES

- [1] Video coding for low bit rate communication (itu-t rec. h.263), November 1995. version 1.
- [2] Hypertext markup language (html) specification, December 1999.
- [3] Advanced video coding for generic audiovisual services (itu-t rec. h.264, iso/iec 14496-10 avc), November 2003.
- [4] Draft standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements (ieee p802.11n/d2.00), February 2007.
- [5] N. Abramson. *Information Theory and Coding*. McGraw-Hill, U.S.A., 1963.
- [6] E. Akyol, A.M. Tekalp, and M.R. Civanlar. A flexible multiple description coding framework for adaptive peer-to-peer video streaming. *IEEE Journal of Selected Topics in Signal Processing*, 1(2) :231–245, August 2007.
- [7] J. B. Anderson and S. Mohan. *Source and channel coding : an algorithmic approach*. Kluwer Academic Publishers, U.S.A., 1991.
- [8] J. Apostolopoulos. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In *Proceedings of Visual Communications and Image Processing (VCIP'01)*, pages 392–409, January 2001.
- [9] J.G. Apostolopoulos, W. Tan, and S.J. Wee. Performance of a multiple description streaming media content delivery network. In *Proceedings of IEEE International Conference on Image Processing (ICIP'02)*, pages 189–192, Rochester, U.S.A., September 2002. vol. 2.
- [10] J.G. Apostolopoulos and S.J. Wee. Unbalanced multiple description video communication using path diversity. In *Proceedings of IEEE International Conference on Image Processing (ICIP'01)*, pages 966–969, Thessaloniki, Greece, October 2001. vol. 1.
- [11] J.G. Apostolopoulos, T. Wong, W. Tan, and S.J. Wee. On multiple description streaming with content delivery networks. In *Proceedings of IEEE International Conference on Image Processing (ICIP'02)*, pages 1736–1745, Rochester, U.S.A., June 2002.

- [12] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20(2) :284–287, March 1974.
- [13] I.V. Bajic and J.W. Woods. Domain-based multiple description coding of images and video. *IEEE Transactions on Image Processing*, 12(10) :1211–1225, October 2003.
- [14] R. Bauer and J. Hagenauer. On variable length codes for iterative source/channel decoding. In *Proceedings of the IEEE Data Compression Conference (DCC'98)*, pages 272–282, Snowbird, U.S.A., 1998.
- [15] R. Bauer and J. Hagenauer. Symbol-by-symbol map decoding of variable length codes. In *Proceedings of the 3rd ITG Conference Source and Channel Coding*, pages 111–116, Mnchen, Germany, 2000.
- [16] R. Bauer and J. Hagenauer. Turbo-fec/vlc-decoding and its application to text compression. In *Proceedings of the Conference on Information Sciences and Systems (CISS'00)*, pages WA66 –WA–11, Princeton University, U.S.A., 2000.
- [17] A.C. Begen, M.U. Demircin, and Y. Altunbasak. Packet scheduling for multiple description video streaming in multipoint-to-point networks. In *Proceedings of International Conference on Communications (ICC'04)*, pages 1340–1344, Paris, France, June 2004. vol. 3.
- [18] T.C. Bell, I. Whitten, and J. Cleary. *Text Compression*. Prentice Hall, 1990.
- [19] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Serial concatenation of interleaved codes : performance analysis, design, and iterative decoding. *IEEE Transactions on Information Theory*, 44(3) :909–926, May 1998.
- [20] C. Bergeron and C. Lamy-Bergot. Soft-input decoding of variable-length codes applied to the h.264 standard. In *Proceedings of the IEEE International Workshop on Multimedia Processing (MMSP'04)*, pages 87–90, Sienna, Italy, September 2004.
- [21] C. Bergeron and C. Lamy-Bergot. Modelling h.264/avc sensitivity for error protection in wireless transmissions. In *Proceedings of the IEEE International Workshop on Multimedia Processing (MMSP'06)*, pages 302–305, Victoria, Canada, October 2006.
- [22] C. Berrou, A. Glavieux, and P. Thitimajhima. Near shannon limit error correcting coding and decoding : Turbo-codes. In *Proceedings of International Conference on Communications (ICC'93)*, pages 1067–1070, Geneva, Switzerland, May 1993.
- [23] R.C. Bose and D.K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1) :68–79, 1960.
- [24] C. Boyd, J. Cleary, I. Irvine, I. Rinsma-Melchert, and I. Witten. Integrating error detection into arithmetic coding. *IEEE Transactions on Communications*, 45(1) :1–3, January 1997.
- [25] V. Buttigieg and P.G. Farrell. A map decoding algorithm for variable-length error-correcting codes. In The Institute of Mathematics and its Applications, editors, *In Codes and Cyphers : Cryptography and Coding IV*, pages 103–109, Essex, United Kingdom, 1995.
- [26] J.W. Byers, M.G. Luby, and M. Mitzenmacher. Accessing multiple mirror sites in parallel : using tornado codes to speed up downloads. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'99)*, pages 275–283, New York, U.S.A., March 1999. vol. 1.
- [27] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles : Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2) :489–509, February 2006.
- [28] J. Chakareski, S. Han, and B. Girod. Layered coding vs. multiple descriptions for video streaming over multiple paths. In *Proceedings of the eleventh ACM international conference on Multimedia (MULTIMEDIA'03)*, pages 422–431, Berkeley, U.S.A., November 2003.
- [29] H.-T. Chan, C.-M. Fu, and C.-L. Huang. A new error resilient video coding using matching pursuit and multiple description coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(8) :1047–1052, April 2005.
- [30] C.J. Chang and J-P. Thomas. Huffman algebras for independent random variables. *Discrete Event Dynamic Systems : Theory and Applications*, 4(1) :23–40, February 1994.
- [31] S.S. Channappayya, L. Joonsoo, R.W. Heath Jr., and A.C. Bovik. Frame based multiple description image coding in the wavelet domain. In *Proceedings of IEEE International Conference on Image Processing (ICIP'05)*, Genoa, Italy, September 2005. vol. 3.
- [32] Z. Chen, C. Guillemot, and R. Ansari. Multiple description coding based on multiwavelet transforms. In *Proceedings of Picture Coding Symposium (PCS'01)*, Seoul, Korea, April 2001.
- [33] J. Chou and K. Ramchandran. Arithmetic coding-based continuous error detection for efficient arq-based image transmission. *IEEE Journal on Selected Areas in Communications*, 18(6) :861–867, June 2000.
- [34] P. Chou, H. Wang, and V. Padmanabhan. Layered multiple description coding. In *Proceedings of Packet Video workshop*, Nantes, France, April 2003.

- [35] P.A. Chou, S. Mehrotra, and A. Wang. Multiple description decoding of overcomplete expansions using projections onto convex set. In *Proceedings of the IEEE Data Compression Conference (DCC'99)*, pages 72–81, Snowbird, U.S.A., March 1999.
- [36] D-M. Chung and Y. Wang. Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(6) :895–908, September 1999.
- [37] D-M. Chung and Y. Wang. Lapped orthogonal transforms designed for error resilient image coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(9) :752–764, September 2002.
- [38] D. Comas, R. Singh, and A. Ortega. Rate-distortion optimization in a robust video transmission based on unbalanced multiple description coding. In *Proceedings of IEEE International Workshop on Multimedia Signal Processing (MMSP'01)*, pages 581–586, Cannes, France, October 2001.
- [39] D. Comas, R. Singh, A. Ortega, and F. Marqués. Unbalanced multiple-description video coding with rate-distortion optimization. *EURASIP Journal on Applied Signal Processing*, 2003(1) :81–90, January 2003.
- [40] T.M. Cover and J. M. Thomas. *Elements of Information Theory*. Wiley, New-York, U.S.A., 1991.
- [41] C. Demiroglu, W. Hoffman, and K. Sayood. Joint source channel coding using arithmetic codes and trellis coded modulation. In *Proceedings of the IEEE Data Compression Conference (DCC'01)*, pages 302–311, Snowbird, U.S.A., March 2001.
- [42] M. Van der Schaar, S. Krishnamachari, S. Choi, and X. Xu. Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 wlans. *IEEE Journal on Selected Areas for Communications*, 21(10) :1752–1763, December 2003.
- [43] P. Deutsch. Deflate compressed data format specification version 1.3. RFC 1951 (Informational), May 1996.
- [44] P. Deutsch. Gzip file format specification version 4.3. RFC 1952 (Informational), May 1996.
- [45] P. Deutsch. Zlib compressed data format specification version 3.3. RFC 1950 (Informational), May 1996.
- [46] B. Dongsheng, W. Hoffman, and K. Sayood. State machine interpretation of arithmetic codes for joint source and channel coding. In *Proceedings of the IEEE Data Compression Conference (DCC'06)*, pages 143–152, Snowbird, U.S.A., March 2006.
- [47] D.L. Donoho. Compressed Sensing. *IEEE Transactions on Information Theory*, 52(4) :1289–1306, April 2006.
- [48] P. Duhamel and M. Kieffer. Joint source and channel decoding : an idea whose time has come. Half Day Tutorial given at IEEE Conference ICASSP 2007, with paper copy distributed to participants, April 2007.
- [49] P. Duhamel and O. Rioul. Codage conjoint source/canal : enjeux et approches. In *Actes du seizième colloque GretsI*, pages 699–704, Grenoble, France, September 1997. http://irevues.inist.fr/bitstream/2042/12708/1/175_1195.pdf.
- [50] I. Fernández Daz, D. Epema, and J. de Jongh. Multipath routing and multiple description coding in ad-hoc networks : a simulation study. In *Proceedings of the first ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN'04)*, pages 46–51, October 2004.
- [51] P. Elias. Coding for noisy channels. In *Proceedings of the IRE National Convention Record*, pages 37–46, March 1955. vol. 3, part 4.
- [52] P. Elias. Universal codeword sets and representations of the integers. *IEEE Transactions on Information Theory*, 6(3) :194–203, March 1975.
- [53] R. Fielding et al. Hypertext transfer protocol – http/1.1. RFC 2616 (Draft Standard), June 1999.
- [54] N. Farvardin and V. Vaishampayan. On the performance and complexity of channel optimized vector quantizers. *IEEE Transactions on Information Theory*, 37(1) :155–160, January 1991.
- [55] G. Feideropoulou, J.E. Fowler, B. Pesquet-Popescu, and J-C. Belfiore. Joint source-channel coding with partially coded index assignement for robust scalable video. *IEEE Signal Processing Letters*, 13(4) :201–204, April 2006.
- [56] M. Fleming and M. Effros. Generalized multiple description vector quantization. In *Proceedings of the IEEE Data Compression Conference (DCC'99)*, pages 3–12, Salt Lake City, U.S.A., March 1999.
- [57] G.D. Forney. Concatenated codes. Ph.d. report, M.I.T. Press, Cambridge, U.S.A., 1966.
- [58] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro. Multiple description video coding for scalable and robust transmission over ip. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(3) :321–334, March 2005.
- [59] R.G. Gallager. Low-density parity-check codes. Ph.d. report, M.I.T. Press, Cambridge, U.S.A., 1963.
- [60] M. Gallant and F. Kossentini. Rate-distortion optimized layered coding with unequal error protection for robust internet video. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3) :357–372, March 2001.

- [61] M.J.E. Goyal. Notes on digital coding. In *Proceedings of the Institute of Radio Engineers (IRE)*, page 657, 1949. vol. 37.
- [62] Andrea Goldsmith. *Wireless communications*. Cambridge University Press, 2005.
- [63] V.K. Goyal. Multiple description coding : compression meets the network. *IEEE Signal Processing Magazine*, 18(5) :74–93, September 2001.
- [64] V.K. Goyal, J. Kovacevic, R. Arean, and M. Vetterli. Multiple description transform coding of images. In *Proceedings of IEEE International Conference on Image Processing (ICIP'98)*, pages 674–678, Chicago, U.S.A., October 1998.
- [65] V.K. Goyal, J. Kovacevic, and J.A. Kelner. Quantized frame expansions with erasures. *Journal of Applied and Computational Harmonic Analysis*, 10(3) :203–233, May 2001.
- [66] M. Grangetto and P. Cosman. Map decoding of arithmetic codes with a forbidden symbol. In *Proceedings of Advanced Concepts for Intelligent Vision Systems conference (ACVIS'02)*, Ghent, Belgium, September 2002.
- [67] M. Grangetto, P. Cosman, and G. Olmo. Joint source/channel coding and map decoding of arithmetic codes. *IEEE Transactions on Communications*, 53(6) :1007–1016, June 2005.
- [68] M. Grangetto, E. Magli, and G. Olmo. Error resilient mq coder and map jpeg-2000 decoding. In *Proceedings of IEEE International Conference on Image Processing (ICIP'04)*, pages 2551–2554, Singapore, October 2004. vol.4.
- [69] M. Grangetto, E. Magli, and G. Olmo. Joint source-channel iterative decoding of arithmetic codes. In *Proceedings of International Conference on Communications (ICC'04)*, pages 20–24, Paris, France, June 2004.
- [70] T. Guionnet and C. Guillemot. Soft decoding and synchronization of arithmetic codes : application to image transmission over noisy channels. *IEEE Transactions on Image Processing*, 12(12) :1599–1609, December 2003.
- [71] T. Guionnet and C. Guillemot. Soft and joint source-channel decoding of quasi-arithmetic codes. *EURASIP Journal on Applied Signal Processing*, 2004(3) :393–411, March 2004.
- [72] T. Guionnet, C. Guillemot, and E. Fabre. Soft decoding of multiple descriptions. In *Proceedings of IEEE International Conference on Multimedia & Expo (ICME'02)*, Lausanne, Switzerland, August 2002.
- [73] T. Guionnet, C. Guillemot, and S. Pateux. Embedded multiple description coding for progressive image transmission over unreliable channels. In *Proceedings of IEEE International Conference on Image Processing (ICIP'01)*, pages 94–97, Thessaloniki, Greece, October 2001. vol. 1.
- [74] L. Guivarch, P. Siohan, and J.C. Carlach. Low complexity soft decoding of huffman encoded markov sources using turbo-codes. In *Proceedings of International Conference on Telecommunications (ICT'00)*, pages 872–876, Acapulco, Mexico, May 2000. vol.2.
- [75] J. Hagenauer and P. Hoehner. A viterbi algorithm with soft-decision outputs and its applications. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'89)*, pages 1680–1686, Dallas, U.S.A., November 1989.
- [76] J. Hagenauer and T. Stockhammer. Channel coding and transmission aspects for wireless multimedia. *Proceedings of the IEEE*, 87(10) :1764–1777, October 1999.
- [77] R.W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2) :147–160, April 1950.
- [78] Z. He, J. Cai, and C.W. Chen. Joint source channel rate-distorsion analysis for adaptive mode selection and rate control in wireless video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6) :511–523, June 2002.
- [79] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava. Promise : peer-to-peer media streaming using collectcast. In *Proceedings of the eleventh ACM international conference on Multimedia (MULTIMEDIA'03)*, pages 45–53, Berkeley, U.S.A., November 2003.
- [80] B.D. Pettijohn W. Hoffman and K. Sayood. Joint source/channel coding using arithmetic codes. *IEEE Transactions on Communications*, 49(5) :826–836, May 2001.
- [81] T. Holliday, A.J. Goldsmith, and H.V. Poor. Joint source and channel coding for mimo systems : Is it better to be robust or quick? *Soft and joint source-channel decoding of quasi-arithmetic codes*, 54(4) :1393–1405, April 2008.
- [82] P.G. Howard and J.S. Vitter. Practical implementations of arithmetic coding. *Image and Text Compression*, 13(7) :85–112, 1992.
- [83] T.C. Hu, D.J. Kleitman, and J.K. Tamaki. Binary trees optimum under various criteria. *SIAM Journal on Applied Mathematics*, 37(2) :246–256, April 1979.
- [84] D.A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers (IRE)*, (1098–1102), September 1952.

- [85] A. Jagmohan and K. Ratakonda. Multiple description coding of predictively encoded sequences. In *Proceedings of the IEEE Data Compression Conference (DCC'02)*, pages 13–22, Snowbird, U.S.A., April 2002.
- [86] S. Ben Jamaa, M. Kieffer, and P. Duhamel. Décodage arithmétique séquentiel avec réglage objectif du compromis complexité-efficacité. In *Actes des Journées en Compression et représentation des signaux audiovisuels (CORESA'05)*, pages 159–164, Rennes, France, November 2005.
- [87] S. Ben Jamaa, M. Kieffer, and P. Duhamel. Controlled complexity map decoding of cabac encoded data. In *Proceedings of IEEE International Conference on Multimedia & Expo (ICME'06)*, pages 1441–1444, Toronto, Canada, July 2006.
- [88] S. Ben Jamaa, M. Kieffer, and P. Duhamel. Exact map decoding of cabac encoded data. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, pages 569–572, Toulouse, France, May 2006.
- [89] S. Ben Jamaa, M. Kieffer, and P. Duhamel. Improving sequential decoding of cabac encoded data via objective adjustment of the complexity-efficiency trade-off. In *Proceedings of Visual Communications and Image Processing (VCIP'06)*, San Jose, U.S.A., January 2006.
- [90] N. Jayant. Subsampling of a dpcm speech channel to provide two 'self-contained' half-rate channels. *Bell System Technical Journal*, 60(4) :501–509, April 1981.
- [91] M. Jeanne, J.-C. Carlach, and P. Siohan. Joint source-channel decoding of variable-length codes for convolutional codes and turbo codes. *IEEE Transactions on Communications*, 53(1) :10–15, January 2005.
- [92] D.S. Parker Jr. Conditions for optimality of the huffman algorithm. *SIAM Journal on Computing*, 9(3) :470–189, August 1980.
- [93] S. Kaiser and M. Bystrom. Soft decoding of variable-length codes. In *Proceedings of International Conference on Communications (ICC'00)*, pages 1203–1207, New Orleans, U.S.A., June 2000. vol.3.
- [94] V. Kawadia and P.R. Kumar. A cautionary perspective on cross-layer design. *IEEE Wireless Communications*, 12(1) :3–11, February 2005.
- [95] S. Khan, R. Schollmeier, and E. Steinbach. A performance comparison of multiple description video streaming in peer-to-peer and content delivery networks. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME'04)*, pages 503–506, Taipei, Taiwan, June 2004. vol.1.
- [96] B.J. Kim, Z. Xiong, and W.A. Pearlman. Low bit-rate scalable video coding with 3-d set partitioning in hierarchical trees (3-d spiht). *IEEE Transactions on Circuits and Systems for Video Technology*, 10(8) :1374–1387, December 2000.
- [97] C.-S. Kim, R.-C. Kim, and S.-U. Lee. Robust transmission of video sequence using double-vector motion compensation. *IEEE Transactions on Circuits Systems for Video Technology*, 11(9) :1011–1021, September 2001.
- [98] C.-S. Kim and S.-U. Lee. Multiple description coding of motion fields for robust video transmission. *IEEE Transactions on Circuits Systems for Video Technology*, 11(9) :999–1010, September IEEE Transactions on Circuits Systems for Video Technology.
- [99] I.K. Kim and N.I. Cho. Error resilient video coding using optimal multiple description of dct coefficients. In *Proceedings of IEEE International Conference on Image Processing (ICIP'03)*, pages II–663–II–666, Barcelona, Spain, September 2003. vol.3.
- [100] I.K. Kim and N.I. Cho. Hybrid multiple description video coding using optimal dct coefficient splitting and sd/md switching. *Signal Processing : Image Communication*, 21(4) :293–305, April 2006.
- [101] J. Kim, R.M. Mersereau, and Y. Altunbasak. Distributed video streaming using multiple description coding and unequal error protection. 2005, 14(7) :849–861, July IEEE Transactions on Image Processing.
- [102] Y.K. Kim and J.B. Ra. Adaptive learning method in self-organizing map for edge preserving vector quantization. *IEEE Transactions on Neural Networks*, 6(1) :278–280, January 1995.
- [103] D.E. Knuth. Huffman's algorithm via algebra. *Journal of Combinatorial Theory, Series A*, 32(2) :216–224, March 1982.
- [104] C. Lamy and O. Pothier. Reduced complexity maximum a posteriori decoding of variable-length codes. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'01)*, pages 1410–1413, San Antonio, U.S.A., Nov. 2001. vol. 2.
- [105] C. Lamy-Bergot, N. Chautru, and C. Bergeron. Unequal error protection for h.263+ bitstreams over a wireless ip network. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*, pages V–377–V–380, Toulouse, France, May 2006.
- [106] F. Le Léanec. Codage vidéo robuste et hiérarchique pour la transmission sur réseaux hétérogènes. Ph.d. report, IRISA, Rennes, France, 2001.

- [107] C.M. Lee. Outils de codage source-canal conjoint pour la transmission robuste de vidéos. Ph.d. report, Université de Paris-Sud, Orsay, France, 2004.
- [108] C.M. Lee, M. Kieffer, and P. Duhamel. Soft decoding of vlc encoded data for robust transmission of packetized video. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, pages III-737-III-740, Philadelphia, U.S.A., March 2005.
- [109] Y.-C. Lee, Y. Altunbasak, and R.M. Mersereau. A two-stage multiple description video coder with drift-preventing motion compensated prediction. In *Proceedings of IEEE International Conference on Image Processing (ICIP'02)*, pages 557-560, Rochester, U.S.A., June 2002. vol.3.
- [110] Y.-C. Lee, Y. Altunbasak, and R.M. Mersereau. A drift-free motion-compensated predictive encoding technique for multiple description coding. In *Proceedings of IEEE International Conference on Multimedia & Expo (ICME'03)*, pages III-581-III-584, Baltimore, U.S.A., July 2003.
- [111] Y.-C. Lee, K. Joohee Kim, Y. Altunbasak, and R.M. Mersereau. Layered coded vs. multiple description coded video over error-prone networks. *Signal Processing : Image Communication*, 18(5) :337-356, May 2003.
- [112] Y. Li, S. Mao, and S.S. Panwar. The case for multipath multimedia transport over wireless ad hoc networks. In *Proceedings of International Conference on Broadband Networks (BROADNETS'04)*, pages 486-495, San Jose, U.S.A., October 2004.
- [113] S. Lin and D.J. Costello. *Error Control Coding : Fundamentals and applications*. Prentice-Hall, Englewood Cliffs, U.S.A., 1983.
- [114] S. Lonardi and W. Szpankowski. Joint source-channel lz'77 coding. In *Proceedings of the IEEE Data Compression Conference (DCC'03)*, pages 273-282, Snowbird, U.S.A., March 2003.
- [115] M. Luby. Lt codes. In *IEEE Symposium on Foundations of Computer Science*, pages 271-282, November 2002.
- [116] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proceedings of Symposium on Theory of Computing*, pages 150-159, El Paso, U.S.A., May 1997.
- [117] D.J.C. MacKay. Fountain codes. *IEE Proceedings Communications*, 152(6) :1062-1068, December 2005.
- [118] A. Majumdar, R. Puri, and K. Ramchandran. Distributed multimedia transmission from multiple servers. In *Proceedings of IEEE International Conference on Image Processing (ICIP'02)*, pages III-177-III-180, Rochester, U.S.A., June 2002.
- [119] S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41(12) :3397-3415, December 1993.
- [120] S. Mao, X. Cheng, Y.T. Hou, and H.D. Sherali. Multiple description video multicast in wireless ad hoc networks. In *Proceedings of International Conference on Broadband Networks (BROADNETS'04)*, pages 671-680, San Jose, U.S.A., October 2004.
- [121] S. Mao, X. Cheng, Y.T. Hou, and H.D. Sherali. Multiple description video multicast in wireless ad hoc networks. *ACM/Kluwer Mobile Networks and Applications Journal (MONET)*, 11(1) :63-73, February 2006.
- [122] S. Mao, X. Cheng, Y.T. Hou, H.D. Sherali, and J.H. Reed. On joint routing and server selection for multiple description video streaming in wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 6(1) :338-347, January 2007.
- [123] S. Mao, S. Lin, S. Panwar, Y. Wang, and E. Celebi. Video transport over ad hoc networks : multistream coding with multipath transport. *IEEE Journal on Selected Areas in Communications*, 21(10) :1721-1737, December 2003.
- [124] S. Mao, S. Lin, Y. Wang, and S. Panwar. Reliable transmission of video over ad-hoc networks using automatic repeat request and multi-path transport. In *Proceedings of Vehicular Technology Conference (VTC Fall'01)*, pages 615-619, Atlantic City, U.S.A., October 2001.
- [125] D. Marpe, G. Blattermann, G. Heising, and T. Weigand. Further results for cabac entropy coding scheme, 2001. Austin, U.S.A.
- [126] M.G. Martini and M. Chiani. Rate-distortion models for unequal error protection for wireless video transmission. In *Proceedings of Vehicular Technology Conference (VTC Spring'04)*, pages 1049-1053, Milan, Italy, May 2004.
- [127] M.G. Martini, M. Mazzotti, C. Lamy-Bergot, J. Huusko, and P. Amon. Content adaptive network aware joint optimization of wireless video transmission. *IEEE Communications Magazine*, 45(1) :84-90, January 2007.
- [128] J.L. Massey. Joint source and channel coding. NATO Advanced Studies Institutes Series E25, Commun. Systems and Random Process Theory, 1978. J.K. Skwirzynski editor.
- [129] K.R. Matty and L.P. Kondi. Balanced multiple description video coding using optimal partitioning of the dct coefficients. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(7) :928-934, July 2005.

- [130] P. Maymounkov. Online codes. Technical Report TR2002-833, New York University, New York, U.S.A., November 2002.
- [131] G.R. Mohammad-Khani, M. Kieffer, and P. Duhamel. Simplification of vlc tables with application to ml and map decoding algorithms. *IEEE Transactions on Communications*, 54(10) :1835–1844, October 2006.
- [132] A.E. Mohr, E.A. Riskin, and R.E. Ladner. Unequal loss protection : graceful degradation of image quality over packet erasure channels through forward error correction. *IEEE Journal on Selected Areas Communications*, 18(6) :819–828, June 2000.
- [133] H. Nguyen and P. Duhamel. Estimation of redundancy in compressed image and video data for joint source- channel decoding. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'03)*, pages 2198–2202, San Francisco, U.S.A., December 2003.
- [134] H. Nguyen and P. Duhamel. Iterative joint source-channel decoding of variable length encoded video sequences exploiting source semantics. In *Proceedings of IEEE International Conference on Image Processing (ICIP'04)*, pages 3221–3224, Singapore, October 2004.
- [135] T. Nguyen and A. Zakhor. Protocols for distributed video streaming. In *Proceedings of IEEE International Conference on Image Processing (ICIP'02)*, pages III–185–III–188, Rochester, U.S.A., June 2002.
- [136] T. Nguyen and A. Zakhor. Matching pursuits based multiple description video coding for lossy environments. In *Proceedings of IEEE International Conference on Image Processing (ICIP'03)*, pages 57–60, Barcelona, Spain, September 2003.
- [137] V.T. Nguyen, E.C. Chang, and W.T. Ooi. Layered coding with good allocation outperforms multiple description coding over multiple paths. In *Proceedings of IEEE International Conference on Multimedia & Expo (ICME'04)*, pages 1067–1070, Taipei, Taiwan, June 2004.
- [138] J. Ostergaard, J. Jensen, and R. Heusdens. n-channel entropy-constrained multiple-description lattice vector quantization. *IEEE Transactions on Information Theory*, 52(5) :1956–1973, May 2006.
- [139] V. Padmanabhan, H. Wang, P. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In *Proceedings of Network and Operating System Support for Digital Audio and Video (NOSSDAV'02)*, Miami, U.S.A., May 2002.
- [140] R.C. Pasco. Source coding algorithms for fast data compression. Ph.d. report, Dept. of EE, Stanford University, Palo Alto, U.S.A., 1976.
- [141] W.B. Pennebaker, J.L. Mitchell, G.G. Langdon, and R.B. Arps. An overview of the basic principles of the q-coder adaptive binary arithmetic coder. *IBM Journal of Research and Development*, 32(6) :717–726, November 1988.
- [142] M. Pereira, M. Antonini, and M. Barlaud. Multiple description image and video coding for wireless channels. *Signal Processing : Image Communication*, 18(10) :925–945, November 2003.
- [143] L. Perros-Meilhac and C. Lamy. Huffman tree based metric derivation for a low-complexity sequential soft vlc decoding. In *Proceedings of International Conference on Communications (ICC'02)*, pages 783–787, New York, U.S.A., April-May 2002. vol.2.
- [144] T. Petrisor, A. Fraysse, B. Pesquet-Popescu, and J.-C. Pesquet. Une autre approche au codage par descriptions multiples, utilisant des représentations creuses. In *Actes du vingt-et-une ième colloque Gretsi*, Troyes, France, September 2007.
- [145] T. Petrisor, B. Pesquet-Popescu, and J.-C. Pesquet. Wavelet-based multiple description coding of images with iterative convex optimization techniques. In *Proceedings of IEEE International Conference on Image Processing (ICIP'05)*, pages 924–927, Genoa, Italy, September 2005.
- [146] T. Petrisor, B. Pesquet-Popescu, and J.-C. Pesquet. A compressed sensing approach to frame-based multiple description coding. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'07)*, Hawaii, U.S.A., April 2007.
- [147] J.A. Pouwelse, J.R. Taal, R.L. Lagendijk, D.H.J. Epema, and H.J. Sips. Real-time video delivery using peer-to-peer bartering networks and multiple description coding. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 4599–4605, The Hague, The Netherlands, October 2004.
- [148] J.G. Proakis. *Digital Communications*. McGraw-Hill, Singapore, 1995.
- [149] R. Puri and K. Ramchandran. Multiple description source coding through forward error correction codes. In *Proceedings of 33rd Asilomar Conference on Signals, System and Computers (CSCC'99)*, Pacific Grove, U.S.A., October 1999.
- [150] L. Qiu, V. Padmanabhan, and G. Voelker. On the placement of web server replicas. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'01)*, pages 1587–1596, Anchorage, Alaska, April 2001.
- [151] I. Radulovic and P. Frossard. Fast index assignment for balanced n-description scalar quantization. In *Proceedings of the IEEE Data Compression Conference (DCC'05)*, page 474, Snowbird, U.S.A., March 2005.

- [152] I. Radulovic and P. Frossard. Multiple description coding with redundant expansions and application to image communications. *EURASIP Journal on Image and Video Processing*, 2007(Article ID 24863), 2007.
- [153] I. Radulovic, Y.-K. Wang, S. Wenger, A. Hallapuro, M.M. Hannuksela, and P. Frossard. Multiple description h.264 video coding with redundant pictures. In *Proceedings of the ACM International Workshop on Mobile Video*, pages 37–42, Augsburg, Germany, September 2007.
- [154] K. Ramchandran, A. Ortega, K.M. Uz, and M. Vetterli. Multiresolution broadcast for digital hdtv using joint source/channel coding. *IEEE Journal on Selected Areas in Communications*, 11(1) :6–23, January 1993.
- [155] R. Puri K. Ramchandran, K.W. Lee, and V. Bharghavan. Application of fec based multiple description coding to internet video streaming and multicast. In *Proceedings of Packet Video Workshop*, Forte Village Resort, Sardinia, Italy, May 2000.
- [156] I. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8 :300–304, 1960.
- [157] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, and R. Puri. Multiple-description video coding using motion-compensated temporal prediction. *IEEE Trans. on Circuits and Systems for Video Technology*, 12(3) :193–204, March 2002.
- [158] A. Reibman, Y. Wang, X. Qiu, Z. Jiang, and K. Chawla. Transmission of multiple description and layered video over an egprs wireless network. In *Proceedings of IEEE International Conference on Image Processing (ICIP'00)*, pages 136–139, Vancouver, Canada, September 2000.
- [159] A.R. Reibman, H. Jafarkhani, M.T. Orchard, and Y. Wang. Performance of multiple description coders on a real channel. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99)*, pages 2415–2418, Phoenix, U.S.A., March 1999. vol.5.
- [160] A.R. Reibman, H. Jafarkhani, M.T. Orchard, and Y. Wang. Multiple description video using rate-distortion splitting. In *Proceedings of IEEE International Conference on Image Processing (ICIP'01)*, pages 978–981, Thessaloniki, Greece, October 2001. vol.1.
- [161] J. Rissanen. Generalized kraft inequality and arithmetic coding. *IBM Journal of Research and Development*, 20(3) :198–203, May 1976.
- [162] J. Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, 30(4) :629–636, July 1984.
- [163] G. Sabeva, S. Ben-Jamaa, M. Kieffer, and P. Duhamel. Robust decoding of h.264 encoded video transmitted over wireless channels. In *Proceedings of the IEEE International Workshop on Multimedia Processing (MMSP'06)*, pages 9–13, Victoria, Canada, October 2006.
- [164] J. Sayir. On coding by probability transformation. Ph.d. report, EE Department, ETH Zurich, Zurich, Switzerland, 1999.
- [165] K. Sayood, H.H. Otu, and N. Demir. Joint source/channel coding for variable length codes. *IEEE Transactions on Communications*, 48(5) :787–794, May 2000.
- [166] R. Schaefer, T. Wiegand, and H. Schwarz. The emerging h.264/avc standard. Ebu technical review, European Broadcasting Union, December 2003. available at http://www.packetizer.com/codecs/h264/trev_293-schaefer.pdf.
- [167] T. Schierl, K. Gnger, C. Hellge, and T. Wiegand. Svc-based multisource streaming for robust video transmission in mobile ad hoc networks. *IEEE wireless communications*, 13(5) :96–103, October 2006.
- [168] E. Setton. Congestion-aware video streaming over peer-to-peer networks. Ph.d. report, Stanford University, Palo Alto, U.S.A., December 2006.
- [169] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 :379–423/623–656, 1948.
- [170] Y.Q. Shi and H. Sun. *Image and video compression for multimedia engineering : fundamentals, algorithms, and standards*. CRC press, Boca raton, U.S.A., 2000. 480 pages.
- [171] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52(6) :2551–2567, June 2006.
- [172] R. Singh, A. Ortega, L. Perret, and W. Jiang. Comparison of multiple description coding and layered coding based on network simulations. In SPIE, editor, *Proceedings of SPIE Image and Video Communications and Processing (IVCP)*, pages 929–939, San Jose, U.S.A., 2000. vol.3974.
- [173] A. Skodras, C. Christopoulos, and T. Ebrahimi. The jpeg 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5) :36–58, September 2001.
- [174] V. Stankovic, R. Hamzaoui, and Z. Xiong. Robust layered multiple description coding of scalable media data for multicast. *IEEE Signal Processing Letters*, 12(2) :154–157, February 2005.

- [175] T. Stockhammer, M.M. Hannuksela, and T. Wiegand. H.264/avc in wireless environments. *IEEE Wireless Communications*, 12(4) :6–13, August 2005.
- [176] J.R. Taal and R.L. Lagendijk. Fair rate allocation of scalable multiple description video for many clients. In *Proceedings of Visual Communications and Image Processing (VCIP'05)*, Beijing, China, July 2005.
- [177] J.R. Taal and R.L. Lagendijk. Asymmetric multiple description coding using layered coding and lateral error correction. In *Proceedings of Twenty-seventh Symposium on Information Theory in the Benelux*, pages 39–44, June 2006.
- [178] J.R. Taal, J.A. Pouwelse, and R.L. Lagendijk. Scalable multiple description coding for video distribution in p2p networks. In *Proceedings of Picture Coding Symposium (PCS'06)*, Beijing, China, April 2006.
- [179] W.-T. Tan and A. Zakhor. Error control for video multicast using hierarchical fec. In *Proceedings of IEEE International Conference on Image Processing (ICIP'99)*, pages 401–405, Kobe, Japan, October 1999.
- [180] W.-T. Tan and A. Zakhor. Real-time internet video using error resilient scalable compression and tcp-friendly transport protocol. *IEEE Transactions on Multimedia*, 1(2) :172–186, June 1999.
- [181] X. Tang and A. Zakhor. Matching pursuits multiple description coding for wireless video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6) :566–575, June 2002.
- [182] D. Taubman and M. Marcellin. *JPEG-2000 image compression fundamentals, standards and practice*. Kluwer Academic Publishers, Norwell, MA, U.S.A., 2002.
- [183] R. Thobanen and J. Kliewer. Robust decoding of variable-length encoded markov sources using a three-dimensional trellis. *IEEE Communications Letters*, 7(7) :320–322, July 2003.
- [184] C. Tillier, T. Petrisor, and B. Pesquet-Popescu. A motion-compensated overcomplete temporal decomposition for multiple description scalable video coding. *EURASIP Journal on Image and Video Processing*, 2007(Article ID 31319), 2007.
- [185] T. Tjalkens. The complexity of minimum redundancy coding. In *Proceedings of IEEE International Symposium on Information Theory (ISIT'00)*, page 373, Sorrento, Italy, June 2000.
- [186] V.A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Transactions on Information Theory*, 39(3) :821–834, May 1993.
- [187] V.A. Vaishampayan and S. John. Balanced interframe multiple description video compression. In *Proceedings of IEEE International Conference on Image Processing (ICIP'99)*, pages 812–816, Kobe, Japan, October 1999.
- [188] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2) :260–269, April 1967.
- [189] D. Wang, N. Canagarajah, and D. Bull. Slice group based multiple description video coding using motion vector estimation. In *Proceedings of IEEE International Conference on Image Processing (ICIP'04)*, pages 3237–3240, Singapore, October 2004.
- [190] Y. Wang and D. Chung. Non-hierarchical signal decomposition and maximally smooth reconstruction for wireless video transmission. in book : *Mobile Multimedia Communications*, 1997.
- [191] Y. Wang and S. Lin. Error-resilient video coding using multiple description motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6) :438–452, June 2002.
- [192] Y. Wang, M.T. Orchard, and A.R. Reibman. Multiple description image coding for noisy channels by pairing transform coefficients. In *Proceedings of the IEEE International Workshop on Multimedia Processing (MMSP'97)*, pages 419–424, Boston, U.S.A., June 1997.
- [193] Y. Wang, M.T. Orchard, V.A. Vaishampayan, and A.R. Reibman. Multiple description coding using pairwise correlating transforms. *IEEE Transactions on Image Processing*, 10(3) :351–366, March 2001.
- [194] Y. Wang, S. Panwar, S. Lin, and S. Mao. Wireless video transport using path diversity : multiple description vs layered coding. In *Proceedings of IEEE International Conference on Image Processing (ICIP'02)*, pages I–21–I–24, Rochester, U.S.A., September 2002.
- [195] Y. Wang, A.R. Reibman, and S. Lin. Multiple description coding for video delivery. *Proceedings of the IEEE*, 93(1) :57–70, January 2005.
- [196] Y. Wang, A.R. Reibman, M.T. Orchard, and H. Jafarkhani. An improvement to multiple description transform coding. *IEEE Transactions on Signal Processing*, 50(11) :2843–2854, November 2002.
- [197] Y. Wang and Q.-F. Zhu. Error control and concealment for video communication : a review. *Proceedings of the IEEE*, 86(5) :974–997, May 1998.

- [198] Y-K. Wang, M.M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj. The error concealment feature in the h.26l test model. In *Proceedings of IEEE International Conference on Image Processing (ICIP'02)*, pages II-729 – II-732, Rochester, U.S.A., September 2002.
- [199] M. Watson, M. Luby, and L. Vicisano. Forward error correction (fec) building block. RFC 5052 (Proposed Standard), August 2007.
- [200] S. Wenger. Video redundancy coding in h.263+. In *Workshop on Audio-Visual Services for packet networks (Packet Video'97)*, Aberdeen, U.K., 1997.
- [201] S. Wenger. H.264/avc over ip. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7) :645–656, July 2003.
- [202] S. Wenger, G. Knorr, J. Ott, and F. Kossentini. Error resilience support in h.263+. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(7) :867–877, November 1998.
- [203] T. Wiegand, G.J. Sullivan, G. Bjntegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Trans. on Circuits and Systems for Video Technology*, 13(7) :560–576, July 2003.
- [204] T. Wiegand, G.J. Sullivan, J. Reichel, H. Schwarz, and M. Wien. Joint draft 10 of svc amendment, April 2007. San Jose, U.S.A.
- [205] I.H. Witten, R.M. Neal, and J.G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6) :520–540, June 1987.
- [206] I.H. Witten, R.M. Neal, and J.G. Cleary. Arithmetic coding revisited. *ACM Transactions on Information Systems*, 16(3) :256–294, July 1998.
- [207] A.J . Wyner. The redundancy and distribution of the phrase lengths of the fixed database lempel-ziv algorithm. *IEEE Transactions on Information Theory*, 43(5) :1452–1464, September 1997.
- [208] L. Xu, W. Hoffman, and K. Sayood. Hard decision and iterative joint source channel coding using arithmetic codes. In *Proceedings of the IEEE Data Compression Conference (DCC'05)*, pages 203–212, Snowbird, U.S.A., March 2005.
- [209] X. Xu, Y. Wang, S.S. Panwar, and K.W. Ross. A peer-to-peer video-on-demand system using multiple description coding and server diversity. In 1759-1762, editor, *Proceedings of IEEE International Conference on Image Processing (ICIP'04)*, Singapore, October 2004.
- [210] R.W. Yeung. *A first course in Information Theory*. Springer, New York, U.S.A., 2002.
- [211] Y. Zhou and W.-Y. Chan. Performance comparison of layered coding and multiple description coding in packet networks. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'05)*, pages 2155–2159, Saint Louis, U.S.A., December 2005.
- [212] K.S. Zigangirov. Some sequential decoding procedures. *Problemy Peredachi Informatsii*, 2(4) :13–15, 1966.
- [213] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3) :337–343, May 1977.