Protection des flux audiovisuels MPEG-4 transportés sur Internet.

Vincent Lecuire[*] — Toufik Ahmed[**] — Ahmed Mehaoua[**]

[*]*CRAN (CNRS UMR 7039)*
Université Henri Poincaré,
Nancy 1 BP 239
F-54506 Vandoeuvre-lès-Nancy Cedex
vincent.lecuire@cran.uhp-nancy.fr

[**]*PRiSM (CNRS FRE 2510)*
Université de Versailles Saint-Quentin-En-Yvelines
F-78035 Versailles
toufik.ahmed@prism.uvsq.fr

RÉSUMÉ. *Le transport de contenus multimédia codés au standard MPEG-4 sur Internet est soumis à des contraintes de débit, de délai et de fiabilité. La charge et la nature du trafic sur le réseau ont des effets sur les pertes et les délais d'acheminement des paquets qui doivent être contrôlés. Dans cet article, nous proposons une technique de protection des objets audiovisuels (AVO) adaptée aux caractéristiques de MPEG-4. La technique définit comment un mécanisme FEC et une architecture Diffserv peuvent être utilisés ensemble pour que leur combinaison fournisse aux AVO l'indice de protection le plus satisfaisant. L'évaluation de performance montre que notre technique de double protection est bien meilleure qu'un système de protection simple ou sans protection pour le niveau de qualité qui est conservé en présence de pertes de paquets.*

ABSTRACT. *Delivering MPEG-4 multimedia content over the Internet is sensitive to bandwidth, packet loss and delay. Packet loss and delay are affected by the volume and the pattern of traffic load on the network. In this paper, we propose a double error protection scheme to reduce the packet loss and the delay effects on the quality of the played-out MPEG-4 Audio-Visual Objects (AVO). This scheme performs a seamless combination of the both FEC and Diffserv mechanisms. Simulation results indicate that when using our error protection scheme, the quality of the received MPEG-4 scene degrades more gracefully when the packet loss increases, compared to a traditional configuration.*

MOT-CLÉS : *Correction anticipée des erreurs, Diffserv, Qualité de Service, Flux continu audiovisuel, codage MPEG-4.*

KEYWORDS: *FEC, Diffserv, Quality of Service, Streaming media, MPEG-4 coding.*

1. Introduction

Le déploiement des applications engageant des flux audiovisuels sur l'Internet est une tendance de plus en plus forte. Mais l'Internet étant un réseau fondé sur le concept de *best-effort*, les paquets sont acheminés sans aucune garantie de délai ni de fiabilité. Cela pose le problème des pertes de paquets, dues à des écartements en cours de route ou à des arrivées en retard. En effet, la tolérance des flux d'images et de sons aux pertes de paquets est faible car la compression effectuée en amont élimine les redondances naturelles de l'information.

On peut distinguer deux approches techniques pour réaliser la protection des flux audiovisuels pendant leur acheminement sur un réseau à commutation de paquets : utiliser un mécanisme de correction anticipée d'erreurs ou utiliser une architecture de Qualité de Service.

La correction d'erreurs par anticipation consiste, côté émetteur, à adjoindre systématiquement dans le flux des données envoyées une certaine quantité de redondances. Côté récepteur, ce supplément d'information va servir au recouvrement des pertes du réseau si elles restent dans une certaine limite. Les mécanismes de ce type, appelés FEC (*Forward Error Correction*), fournissent un contrôle en boucle ouverte. L'opération consiste en fait à réintroduire sous une autre forme des redondances qui avaient été éliminées par le processus de compression. Cela a un coût en terme de bande passante. Quand les pertes sont dues aux congestions du réseau, on va subir à situation égale, plus de pertes pour un flux protégé par FEC que pour un flux non protégé mais moins volumineux. L'efficacité de la protection va dépendre de son aptitude à modifier la distribution des erreurs. Des solutions basées sur les FEC sont présentées dans (Hardman *et al.*, 1995), (Bolot *et al.*, 1996) et (Boutremans *et al.*, 2001) pour les flux audio et dans (Albanese *et al.*, 1994), (Girod *et al.*, 1999) et (Pillai *et al.*, 1999) pour les flux vidéo.

La deuxième approche est d'utiliser une architecture de Qualité de Service, telle que Diffserv, Intserv ou MPLS qui sont les plus connues. Une telle architecture dépasse le concept traditionnel d'un service unique de type *best-effort* rendu par l'Internet pour introduire de nouvelles classes de services prenant en compte des exigences de débit, de délai et de fiabilité. Elle va fixer les mécanismes permettant de faire la discrimination des services et d'allouer les ressources nécessaires à l'exécution de politiques différenciées. On peut ainsi favoriser le trafic lié aux flux audiovisuels au détriment d'autres utilisateurs. Pour les applications audiovisuelles ayant des contraintes d'interactivité, les travaux de (Shin *et al.*, 2000), (Albrecht *et al.*, 2000), (Shao *et al.*, 2000) et (Ahmed *et al.*, 2001) ont montré que Diffserv est l'architecture la plus appropriée.

Dans cet article, nous traitons le cas particulier du transport sur Internet de contenus multimédia codés par MPEG-4, qui sont représentés sous forme d'objets audiovisuels, appelés AVO (*Audio-Visual Objects*). Nous proposons une technique combinant un mécanisme FEC et une architecture Diffserv pour réaliser une double

protection des AVO. La technique s'articule sur deux systèmes originaux : un système d'ajustement dynamique des codes Reed-Solomon dans l'aspect de la protection des AVO par FEC, et un système de réseau de neurones pour la classification des AVO sur l'ensemble des services proposés par Diffserv. Ces deux systèmes vont se compléter pour donner aux AVO le juste niveau de protection en regard des caractéristiques de compression et de la scène. L'évaluation de performance de notre solution est obtenue par des simulations réalisées avec ns2 (*network simulator*) (Fall *et al.*, 2002).

L'article est organisé en quatre parties et une conclusion. La deuxième partie présente un aperçu du standard MPEG-4 et discute des besoins liés à la protection des objets audiovisuels. La troisième partie développe le mécanisme FEC que nous avons conçu pour protéger les AVO et son système d'ajustement dynamique des codes Reed-Solomon. La quatrième partie présente en détail notre algorithme de classification des AVO par un système de réseaux de neurones. Les résultats de l'évaluation de performance sont analysés dans la cinquième partie.

2. Système de classification des flux MPEG-4

MPEG-4 est un standard ISO/IEC qui fournit une spécification des outils d'encodage naturels et synthétiques des flux audio et vidéo. MPEG-4 cible un large rayon d'applications multimédia telles que la vidéoconférence classique et la télédiffusion. L'intérêt principal de ce standard est de fournir un langage commun de description pour l'encodage 2D et 3D des objets audiovisuels (ISO/IEC 14496-1/14496-2/14496-3, 1998). Cette section présente les caractéristiques des flux MPEG-4 et définit un système de classification qui sera utilisé dans notre proposition.

2.1. *La notion d'objets audiovisuels*

Une approche orientée objet est adoptée dans MPEG-4 en vue de représenter des scènes complexes. Une scène est décrite comme une composition d'objets audiovisuels élémentaires, les AVO (*Audio-Visual Objects*), qui sont combinés selon une structure particulière. La figure 1 montre un exemple de scène MPEG-4 composée de quatre AVO. Ces objets sont un logo, un arrière plan, un présentateur et son discours. L'utilisateur ne perçoit pas de la même manière tous les AVO constituant la scène. Dans l'exemple utilisé, la qualité du discours audio est plus importante que la qualité d'image du présentateur, elle-même étant plus importante que la qualité de l'image en arrière plan et du logo.

Il est légitime que le transport des AVO soit assuré à un niveau de protection en rapport avec la valeur que l'utilisateur accorde à chacun d'eux. Encore faut-il que cette valeur soit connue. Cela est discuté dans la section 2.3.

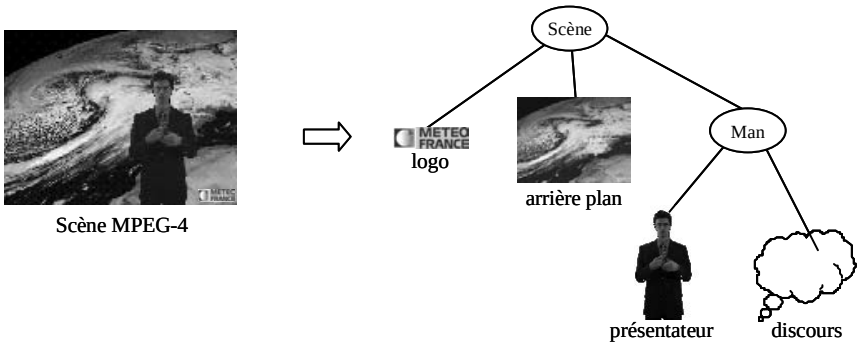


Figure 1. *Exemple de représentation d'une scène MPEG-4 en plusieurs AVO.*

2.2. *L'architecture standard*

Un terminal MPEG-4 est composé de trois couches, comme cela est présenté sur la figure 2 : Compression, Synchronisation et Livraison (ISO/IEC 14496-1, 1998). La couche Compression génère la représentation des données audiovisuelles dans des flux élémentaires, les ES (*Elementary Streams*).

Les relations hiérarchiques, l'emplacement et les propriétés des ES sont décrits par un ensemble de descripteur d'objets, les OD (*Object Descriptors*). Les OD sont eux-mêmes transportés dans un ou plusieurs flux élémentaires. Une scène MPEG-4 est décrite par un descripteur de scène, le SD (*Scene Description*), qui contient les informations sur l'adresse et l'organisation des objets audiovisuels de la scène, en terme d'emplacement spatial et temporel. Le descripteur de scène utilise un langage basé sur le VRML appelé BIFS (*Binary Format for Scene*). Les ES sont fractionnés en une ou plusieurs unités d'accès, les AU (*Access Units*). Une AU est la plus petite entité de données à laquelle on peut assigner une information de temps. Un flux élémentaire MPEG-4 peut être une vidéo MPEG-1 par exemple, et dans ce cas chaque AU va correspondre à une image compressée de type I, B ou P.

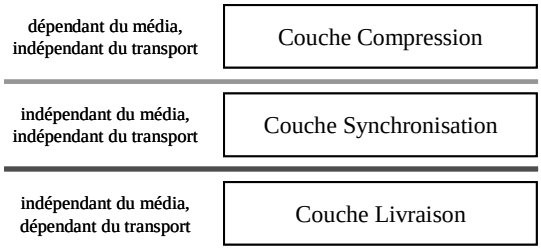


Figure 2. *Architecture en couches d'un système MPEG-4.*

Sur la couche Synchronisation, les AU sont transportées en une séquence de paquets appelés SL-PDU. Un SL-PDU est constitué d'une entête de paquet SL et de données dans le paquet SL. L'entête contient des informations telles qu'un numéro de séquence et une estampille temporelle pour les besoins liés au contrôle des pertes de paquets, ainsi qu'à l'ordre et à l'échéance de leur livraison.

Enfin, l'interface entre la couche Synchronisation et la couche Livraison offre des procédures indépendantes qui permettent d'établir des sessions MPEG-4 et d'accéder à des canaux de transport comme RTP/UDP/IP.

2.3. *Les niveaux de classification des flux*

Deux niveaux de classifications sont utilisés dans notre système. Le premier niveau correspondant à une classification des AVO selon leur importance dans la scène, et le deuxième correspond à une classification inter-AVO, c'est-à-dire entre les unités qui composent le flux d'un même AVO.

2.3.1 Classification des AVO

Dans l'architecture MPEG-4, un AVO est décrit par un descripteur d'objet (OD) qui spécifie la configuration du flux de données. Dans un OD, plusieurs descripteurs coexistent, on y trouve par exemple *DecoderConfigDescriptor* qui décrit le type du décodeur utilisé ainsi que les ressources nécessaires pour bien interpréter le flux de données du coté du client. Le descripteur qui nous intéresse pour les besoins de la classification des flux est celui qui décrit les paramètres de QoS demandés par chaque objet MPEG-4. Ce descripteur est appelé *QoSDescriptor* (voir annexe). *QoSDescriptor* encapsule des paramètres de qualité de service sur le canal de transport demandé par un objet. La table 1 illustre la structure de descripteur de QoS.

Métrique	Description
MAX_DELAY	Délai maximum accepté de bout en bout
PREF_MAX_DELAY	Délai maximum préférable de bout en bout
LOSS_PROB	Probabilité de perte des AU acceptée
MAX_GAP_LOSS	Nombre de perte d'AU consécutives acceptée
MAX_AU_SIZE	Taille maximale d'une AU
AVG_AU_SIZE	Taille moyenne d'une AU
MAX_AU_RATE	Débit maximal des AU
PRIORITY	Priorité du flux

Table 1. *Les paramètres de qualité de service associés aux AVO.*

Nous exploitons les informations contenues dans ce descripteur pour classer les différents objets existants dans une scène MPEG-4. Nous utilisons pour cela un algorithme de classification à base de réseaux de neurones. Cet algorithme est décrit en détail dans la section 4.

2.3.2 Classification des AU

La technique de codage utilisée dans MPEG-4 permet le codage de chaque AVO par un codeur adapté pour obtenir le meilleur ratio de compression. Dans une scène MPEG-4, on peut donc trouver un objet vidéo codé en H263 et un autre en MPEG-1 par exemple.

Le codage MPEG génère trois types d'images codées, fournissant des niveaux de scalabilité temporelle. Ces trois types d'images sont appelées : I-VOP (*Intra coded Video Object Plane*), P-VOP (*temporally Predicted Video Object Plane*), et B-VOP (*Birectionally predicted Video Object Plane*). Les dépendances entre ces différents types d'images sont montrées sur la figure 3.

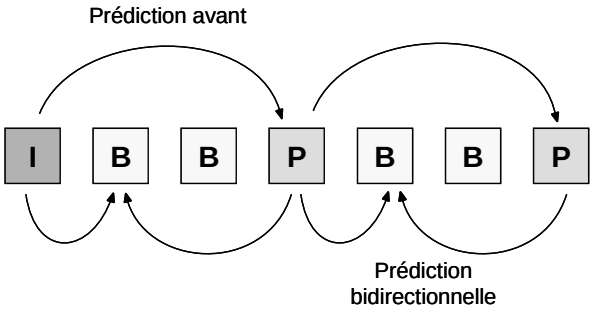


Figure 3. *Dépendances entre les objets de type I, P et B dans un codage MPEG.*

Le codeur MPEG autorise la configuration de la fréquence et de la position des images clés. Cette configuration va déterminer comment les images sont rassemblées en groupes appelés GOP (*Group Of Pictures*), une séquence vidéo étant considérée comme une suite de GOP. Un GOP est constitué de N images. Il commence par une image de type I suivie par des séquences constituées par un nombre (M-1) d'images B et d'une image P. Par exemple, une configuration fixée à M=3 et N=12 donne des groupes d'images formés sur la série IBBPBBPBBPBB. Il est clair que l'importance des AU appartenant aux différents types d'images n'est pas la même. L'image de type I-VOP est la plus importante puisque sans elle, c'est tout le GOP (et même un peu plus, N+M−1 images précisément) qui ne pourra pas être reconstitué. Les images de type P-VOP ont une importance qui dépend de leur position dans le GOP.

D'une manière générale, les sources vidéo avec MPEG-4 sont codées selon un modèle de représentation multiniveau. La source vidéo est codée en plusieurs flux compressés correspondant chacun à un incrément de débit et de qualité en termes de corrélation temporelle, de résolution spatiale, ou de rapport signal sur bruit. On distingue un flux de base (BL ou *Base Layer*), qui est le plus important, et un ou plusieurs flux d'améliorations (EL ou *Enhancement Layer*). Nous exploitons cette propriété pour assurer une protection qui dépend de l'importance de chaque couche. Nous utilisons pour cela un mécanisme de type FEC qui va réaliser une protection inégale. Ce mécanisme est décrit en détail dans la section 3.

3. Protection des AVO par un mécanisme de type FEC

Les mécanismes de correction d'erreurs par anticipation se divisent en deux classes : ceux qui exploitent les caractéristiques du média pour faire les corrections (Hardman *et al.*, 1995), (Bolot *et al.*, 1996), (Pillai *et al.*, 1999) et ceux qui utilisent les codes correcteurs sur le flux de données indépendamment du contenu (Albanese *et al.*, 1994), (Girod *et al.*, 1999). Les solutions basées sur les codes correcteurs sont bien adaptées au contexte des flux MPEG-4 qui intègre de nombreux types de codage. Dans cette section, nous présentons d'abord les principes généraux d'utilisation des codes correcteurs pour se protéger des pertes de paquets. Puis nous décrivons en détail le système que nous proposons pour protéger individuellement les objets audiovisuels MPEG-4.

3.1. *Principes d'utilisation des codes correcteurs*

Les codes correcteurs ont été principalement développés pour corriger des bits erronés ou manquants à l'intérieur des messages. Ils sont largement utilisés dans les systèmes de transmission qui utilisent des canaux fortement bruités (GSM, ADSL, etc..). Ils peuvent aussi être exploités au niveau du transport de bout en bout pour reconstruire les paquets perdus dans le réseau.

3.1.1 Le traitement par blocs de paquets

Le processus d'encodage est un traitement par blocs de paquets (BOP), schématisé sur la figure 4. Considérons un code correcteur quelconque, noté $C(n,k)$. Le BOP est constitué initialement de k paquets de données placés horizontalement. Le codage s'effectue ensuite dans le sens transversal des paquets, c'est-à-dire verticalement sur un rang de k éléments de données pour produire $h = n - k$ éléments de redondance. Cette opération répétée sur toute la largeur du BOP va générer de fait h paquets supplémentaires. Les n paquets du BOP sont alors prêts à être expédiés sur le réseau. Avec un tel traitement par blocs, tous les

paquets doivent nécessairement avoir la même taille. Cette condition est satisfaite au besoin en ayant recours au bourrage.

Le codage proprement dit fait appel à des notions d'algèbre et à des propriétés combinatoires que l'on peut étudier en détail dans (Blahut, 1983), (Lin *et al.*, 1983), ou (Poli *et al.*, 1988). Deux types de codes sont très souvent utilisés dans la protection des flux multimédia : les codes de parité et les codes Reed-Solomon. Les codes Reed-Solomon sont les plus efficaces qui soient connus pour la protection d'erreurs, c'est-à-dire qu'ils réussissent à reconstruire des éléments manquants avec la plus petite fraction d'éléments reçus (Blahut, 1983).

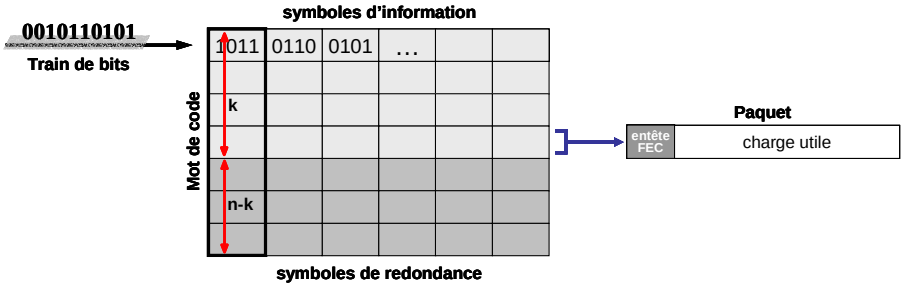


Figure 4. *Construction d'un bloc de paquets (BOP).*

3.1.2 Les codes Reed-Solomon

Ils appliquent des opérateurs polynomiaux dans le corps de Galois $GF(2^m)$, où m est le nombre de bits des éléments manipulés. Dans ce corps fini, tout code $C_{RS}(n,k)$ respectant la condition $n \le 2^m - 1$ est valide. L'usage conduit très souvent à adopter l'octet comme unité élémentaire, et donc les valeurs $m = 8$ et $n \le 255$, car c'est un format naturel qu'il est aisé de manipuler.

L'encodage du i^{ème} rang de données $x_i = \left[x_i^1, \ldots x_i^k\right]$ par un code $C_{RS}(n,k)$ est une opération engageant une matrice génératrice ***G*** de dimension $(n \times k)$ dont les coefficients sont liés au code. Une construction simple pour ***G*** adopte une matrice de Vandermonde sur une forme canonique réduite, comme cela est présenté par la relation [1].

$$
\begin{aligned}
&G : GF(2^8)^k \rightarrow GF(2^8)^n \\
&x_i \mapsto y_i = x_i \cdot G, \quad G = \left[I_k \mid F_{k,h}\right] \\
&\qquad\qquad \text{avec } F_{k,h} = \left\{f_{r,c}\right\} \text{une matrice } (k \times h) \\
&\qquad\qquad \text{de coefficients } f_{r,c} = c^{r-1}, \quad r \in [1,k], c \in [1,h]
\end{aligned}
\qquad [1]
$$

Dans la forme canonique réduite, ***G*** est constituée en partie de la matrice identité I_k. Elle produit donc des éléments codés $y_i = \left[x_i^1, \ldots x_i^k, y_i^{k+1}, \ldots y_i^n\right]$ qui conservent la copie des données d'origine. L'usage de cette forme standard est recommandé car il réduit la complexité du processus de décodage, qui devient même trivial quand tous les paquets de données sont bien reçus.

Ce processus de décodage est engageable par le récepteur, connaissant le code $C_{RS}(n,k)$, si au moins k paquets sont reçus sur les n constituant le BOP. Dans le cas où les k premiers paquets du BOP font partie des paquets reçus, alors toutes les données sont à disposition et il n'y a rien à faire. Dans le cas où un nombre p de paquets parmi les k premiers paquets du BOP sont manquants, le récepteur dispose d'au moins p paquets de redondance. La relation [1] permet de générer un système de p équations à p inconnues qui se résout par élimination gaussienne.

Les délais introduits par les codes Reed-Solomon ont été évalués dans (Rizzo, 1997) pour étudier la performance de leur implantation logicielle dans un protocole de communication. Les résultats ont montré qu'ils pouvaient être utilisés pour des flux au-delà de 8 Mbytes/sec. Les performances sont très largement supérieures aujourd'hui vu l'accroissement de la puissance des systèmes informatiques.

3.1.3 Protection inégale aux erreurs

Protéger un flux de données de manière uniforme coûte cher en terme de bande passante. Cette approche s'avère d'ailleurs inadéquate lorsque les pertes sont dues aux congestions du réseau. Il vaut mieux protéger seulement les unités d'information les plus importantes car la distribution des erreurs est alors modifiée dans le sens d'une plus grande densité sur les informations non protégées. La dégradation de la qualité du signal reconstruit sera donc moins forte.

La protection inégale aux erreurs s'applique bien sur les sources vidéo codées selon un modèle de représentation multiniveau, comme nous l'avons dit dans la section 2.3.2. Le taux de protection assigné à un flux dépend de son niveau de priorité. Le rapport entre la bande passante utilisée et la distorsion induite par les pertes est le critère à optimiser.

Les solutions les plus connues sont basées sur le système PET (*Priority Encoding Technique*), qui a été spécifié formellement dans (Albanese *et al.*, 1994). Ce système définit une structure particulière de BOP qui permet à l'utilisateur de fixer des priorités sur les unités vidéo. Une première application du système PET a été réalisée par (Leicher, 1994) pour le cas des flux vidéo codés par MPEG-1. Une méthode d'optimisation des paramètres de PET est proposée dans (Girod *et al.*, 1999).

Avec le système PET, le traitement est réalisé par groupes contenant toujours le même nombre d'images. Un groupe d'images (GOP) forme donc un intervalle temporel constant, dans lequel on peut déterminer le débit moyen pour chacun des

L flux hiérarchiques. Mais cela introduit nécessairement une latence incompatible avec les contraintes des applications interactives. Le paragraphe suivant présente un nouveau système, appelé TAPIOCA, que nous proposons pour protéger les flux audiovisuels.

3.2. *Le système TAPIOCA*

Comme PET, le système TAPIOCA (Transport Audiovisuel avec Protection Inégale des Objets et Contrôle d'Admission) assure la protection inégale d'une source vidéo codée en L flux hiérarchisés selon leur niveau d'importance. Mais il vise le domaine des applications interactives et les unités vidéo sont codées individuellement. Surtout, il laisse à l'utilisateur le contrôle direct du volume du trafic de redondance qui sera ajouté sur le réseau. Celui-ci intervient sur la consigne $R = [R_1, \ldots R_L]$ donnée au système. Chaque élément R_ℓ de la consigne représente le taux du trafic de redondance qu'on fixe pour le flux de rang ℓ, et que le système va appliquer strictement. Le volume et la répartition du trafic de redondance sont de cette manière parfaitement contrôlés. Le débit moyen du trafic de redondance ajouté par TAPIOCA est $r_{TAPIOCA} = \sum_{\ell=1}^{L} R_\ell \cdot r_\ell$, en posant r_ℓ le débit moyen du flux de rang ℓ fourni par le système de compression.

Pour appliquer la consigne de débit, le système va jouer sur les deux dimensions du BOP. Cela consiste techniquement à faire un ajustement dynamique du code correcteur et de la taille des paquets sur chaque unité vidéo. Ce mécanisme d'ajustement est l'élément déterminant du système TAPIOCA. Puisque l'émetteur va protéger chaque unité vidéo avec un code particulier, ce code doit être joint dans les paquets de transport pour que le récepteur puisse effectuer correctement le décodage. La figure 5 montre le format des paquets transmis sur le réseau. Ils sont constitués par l'encapsulation d'un entête spécifique au FEC puis les entêtes des protocoles impliqués dans le transport proprement dit. Le transport peut être basé sur RTP par exemple. L'entête spécifique au FEC a une taille de quatre octets. Il contient le numéro de l'unité codée dans la séquence vidéo, ainsi que la valeur du code utilisé pour sa protection.

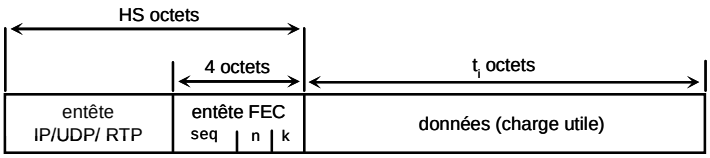


Figure 5. *Format des paquets préparés par le système TAPIOCA.*

Le fonctionnement du système dans sa détermination du code correcteur à adopter pour une unité vidéo particulière est le suivant. Considérons U_ℓ^i la i$^{\text{ème}}$ unité vidéo dans le flux de rang ℓ. Elle va être rangée dans un BOP et protégée par un code correcteur $C_{RS}(n_\ell^i, k_\ell^i)$ avant d'être transportée sur le réseau. Ce code correcteur établit une des dimensions du BOP comme cela a été détaillé dans le paragraphe 3.1.1 : il sera formé de n_ℓ^i paquets dont $n_\ell^i - k_\ell^i$ embarquent des informations de redondance. Pour optimiser la protection des unités vidéo tout en respectant la contrainte R_ℓ posée par l'utilisateur, les valeurs k_ℓ^i et n_ℓ^i du code qui protégera l'unité U_ℓ^i sont déterminées dynamiquement à partir de quatre paramètres connus : m_ℓ^i la taille en octets de l'unité U_ℓ^i, MSS (*Maximal Segment Size*) la taille maximale du champ de données dans un paquet de transport, HS (*Header Size*) la taille de l'entête dans un paquet de transport, et bien sûr R_ℓ le taux du trafic de redondance.

La valeur k_ℓ^i est la première à être identifiée. Elle est définie comme étant le nombre minimal de paquets qui sont nécessaires pour transmettre l'unité U_ℓ^i sans aucune protection. Son calcul est donné par la relation [2], dans laquelle la notation $\lceil x \rceil$ désigne l'arrondi à l'entier supérieur de la valeur x.

$$k_\ell^i = \left\lceil \frac{m_\ell^i}{MSS} \right\rceil \qquad [2]$$

Il est alors établi que l'unité U_ℓ^i est rangée dans un BOP de k_ℓ^i paquets de données faisant tous la même taille $t_\ell^i = \left\lfloor m_\ell^i / k_\ell^i \right\rfloor$, le dernier paquet contenant une quantité $t_\ell^i \cdot k_\ell^i - m_\ell^i$ d'octets de bourrage qui est minimisée.

Il reste à déterminer la valeur n_ℓ^i et le code $C_{RS}(n_\ell^i, k_\ell^i)$ pourra être appliqué sur le BOP. La valeur R_ℓ de la consigne permet de calculer le budget $\overset{*}{\varsigma}_{U_\ell^i}$ qui est réservé au trafic de redondance, exprimé par une quantité d'octets donnée par la relation [3], dans laquelle la notation $\lfloor x \rceil$ désigne l'arrondi à l'entier le plus proche de la valeur x.

$$\overset{*}{\varsigma}_{U_\ell^i} = \lfloor R_\ell \cdot (k_\ell^i \cdot HS + m_\ell^i) \rceil \qquad [3]$$

Comme la technique de codage par BOP produit nécessairement des paquets ayant tous la même taille, la dépense réelle $\varsigma_{U_\ell^i}$ de $n_\ell^i - k_\ell^i$ paquets de redondance, en comptant les octets de bourrage, est donnée par la relation [4].

$$\varsigma_{U_\ell^i} = (n_\ell^i - k_\ell^i)\cdot\left(\left\lceil \frac{m_\ell^i}{k_\ell^i}\right\rceil + HS\right) + \left(k_\ell^i \cdot \left\lceil \frac{m_\ell^i}{k_\ell^i}\right\rceil - m_\ell^i\right) = n_\ell^i \cdot \left\lceil \frac{m_\ell^i}{k_\ell^i}\right\rceil - m_\ell^i + \left(n_\ell^i - k_\ell^i\right)\cdot HS \quad [4]$$

Il y a donc une certaine marge d'erreur entre le budget alloué $\overset{*}{\varsigma}_{U_\ell^i}$ et la dépense réelle $\varsigma_{U_\ell^i}$, calculée par $\varepsilon_{U_\ell^i} = \overset{*}{\varsigma}_{U_\ell^i} - \varsigma_{U_\ell^i}$. Cette erreur peut être positive ou négative et va s'accumuler sur le long de la séquence des unités vidéo qui arrivent au système de codage. A partir des relations [3] et [4], la fluctuation de l'erreur peut être écrire sous la forme d'une suite représentée par la formule [5].

$$\varepsilon_{U_\ell^i}(n_\ell^i, k_\ell^i, m_\ell^i) = \begin{cases} 0 \quad , \ i = 0 \\ \left(R_\ell + 1\right)\cdot m_\ell^i - n_\ell^i \cdot \left\lceil \dfrac{m_\ell^i}{k_\ell^i}\right\rceil + \left(R_\ell \cdot k_\ell^i - n_\ell^i + k_\ell^i\right)\cdot HS \\ \qquad + \varepsilon_{U_\ell^{i-1}}(n_\ell^{i-1}, k_\ell^{i-1}, m_\ell^{i-1}) \quad , \ i > 0 \end{cases} \quad [5]$$

Pour que le débit du trafic de redondance soit maintenu au plus près de la consigne, il faut adopter pour n_ℓ^i la valeur pour laquelle l'erreur $\varepsilon_{U_\ell^i}$ sera la plus proche de zéro. La valeur n_ℓ^i est donc déterminée par l'équation [6] qui se résout facilement.

$$\forall m_\ell^i, \quad n_\ell^i \ \textit{tel que} \quad \min_{n_\ell^i} \left| \varepsilon_{U_\ell^i}(n_\ell^i, k_\ell^i, m_\ell^i) \right| \quad \textit{avec} \quad k_\ell^i = \left\lceil m_\ell^i / MSS \right\rceil \quad [6]$$

On peut remarquer que le code correcteur utilisé pour coder les unités vidéo est ajusté en fonction de leur taille. Cela rend le système TAPIOCA très flexible car il est applicable sur des flux continus constitués d'unités d'accès de toute taille. Il est valable pour des flux vidéo quel que soit le taux de compression, et peut même être utilisé pour des flux mixant l'image et le son comme ceux codés par MPEG-2. Les unités audio vont alors former un flux dont la protection doit être traitée avec la plus grande importance. Avec la forme de la consigne, l'importance que l'utilisateur accorde à chaque flux peut être exprimée finement. Il dispose de tous les moyens pour régler les niveaux de protection et de trafic pour obtenir le plus grand indice de protection.

4. Protection des AVO par un mécanisme de type DiffServ

Les récentes recherches sur le support de la qualité de service sur Internet a conduit à deux approches distinctes : l'architecture Intserv (Wroclawski, 1997) accompagnée de son protocole de signalisation RSVP (*Resource Reservation*

Protocol) (Braden *et al.*, 1997), et l'architecture Diffserv (Blake *et al.*, 1998). Contrairement à la gestion flux par flux de RSVP, les réseaux Diffserv séparent le trafic en plusieurs classes de paquets. La classe à laquelle un paquet est agrégé est marquée dans un champ de l'entête IP, le DSCP (*DiffServ Code Point*). Quand un paquet arrive sur un routeur Diffserv, celui-ci effectue le relayage du paquet en adoptant un comportement qui est fixé par la classe du paquet, appelé son PHB (*Per-Hop Behavior*).

Deux schémas de PHB ont été proposés pour fournir des traitements différenciés sur les paquets :

- Le PHB EF (*Expedited Forwarding*) : ce type de comportement est destiné aux applications qui ont besoin de garanties sur la bande passante, avec un délai d'acheminement des paquets réduit, peu d'effets de gigue et peu de pertes (Jacobson *et al.*, 1999).
- Le PHB AF (*Assured Forwarding*) : ce type de comportement est destiné aux applications qui ont besoin d'un service meilleur que le *best-effort* (Heinanen *et al.*, 1999-a). Il distingue trois niveaux de priorité de perte et quatre niveaux de priorité temporelle à travers les classes de services AF1, AF2, AF3 et AF4. Les paquets de plus faible priorité sont écartés en premier en cas de congestion du réseau.

Diffserv ne fournit pas de garantie absolue sur la qualité du service rendu mais son mécanisme résiste au facteur d'échelle. C'est essentiel dans l'Internet. Avec la gamme des services proposés, Diffserv permet de prendre en compte les besoins de qualité de service des applications audiovisuelles. Reste à définir objectivement comment le trafic va être classifié car cela conditionnera en grande partie les performances. Les fonctionnalités nécessaires aux extrémités d'un domaine Diffserv pour classifier le trafic et affecter les ressources réseau sont définies dans une recommandation de l'IETF (Blake *et al.*, 1998). Plusieurs algorithmes de marquage au niveau des routeurs de bordure sont proposés, parmi lesquels **TSWTCM** (*Time Sliding Window Three Color Marker*) (Fang *et al.*, 2000) et **TRTCM** (*Two Rate Three Color Marker*) (Heinanen *et al.*, 1999-b). Mais ce marquage des paquets est effectué de manière systématique, sans aucune connaissance du type de flux, et en conséquence peut se relever inadapté.

Dans cette section, nous proposons un algorithme de classification du trafic audiovisuel MPEG-4. L'unité d'information considérée est l'AVO MPEG-4. Cette classification effectuée à la source permet d'assurer l'affectation des ressources réseau de manière appropriée. Ainsi, les AVO les plus exigeants en terme de qualité de service auront un marquage conséquent et ainsi de suite. Dans cette section, nous présentons d'abord les principes de la classification à l'aide d'un exemple, puis nous développons l'algorithme adopté, basé sur les réseaux de neurones.

4.1. *Les principes de la classification du trafic MPEG-4*

D'un point de vue lié à l'application, les objets audiovisuels MPEG-4 ont individuellement des besoins de qualité de service qui sont directement donnés par certains champs des descripteurs MPEG-4. Les paramètres indiqués dans ces champs ont été présentés dans la section 2.3.1. Il y a par exemple les valeurs maximales du délai et du taux de pertes tolérés de bout en bout (voir table 1).

D'un autre point de vue, lié cette fois au transport par les réseaux Diffserv, un modèle de classification du trafic en fonction de paramètres de qualité de service est défini dans la recommandation ITU-T Y.1541 (ITU-T Y.1541, 2001). Ce modèle est illustré table 2.

	Classes de trafic				
Paramètre de QoS	**EF**	**AF11**	**AF12**	**AF13**	**Best Effort**
IPTD	100ms	250ms	300ms	400ms	Unspecified
IPDV	50ms	50ms	Unspecified	Unspecified	Unspecified
IPLR	1.10^{-3}	1.10^{-3}	2.10^{-3}	3.10^{-3}	Unspecified
BW	1,5Mbits	1,5Mbits	1,5Mbits	1,5Mbits	4Mbits

(1) **IPTD** (*IP packet transfer delay*) : délai de transfert d'un paquet IP
(2) **IPDV (***IP packet delay variation*) : gigue ou variation du délai
(3) **IPLR** (*IP packet loss ratio*) : probabilité de perte de paquets
(4) **BW** (*Bandwidth used*) : bande passante allouée à la classe de service

Table 2. *Classification du trafic en fonction de paramètres de qualité de service.*

Considérons une scène MPEG-4. Nous prendrons l'exemple décrit en annexe : la scène est constituée de deux AVO ayant chacun ses propres paramètres de qualité de service, récapitulés table 3. Nous les appelons arbitrairement O_A pour l'objet de type H263 et O_B pour celui de type G723.

Objet	**Paramètres contenus dans QoS_Descriptor**	**Valeur**
O_A de type H263	MAX_DELAY PREF_MAX_DELAY LOSS_PROB PRIORITY	350 ms 250 ms 0,01 0 (unspecified)
O_B de type G723	MAX_DELAY PREF_MAX_DELAY LOSS_PROB PRIORITY	150 ms 100 ms 0,001 0 (unspecified)

Table 3. *Exemple de besoins de qualité de service pour une scène MPEG-4.*

Exploitons maintenant le modèle de classification du trafic illustré table 2 pour associer une classe de trafic Diffserv à chaque AVO. Quelle classe Diffserv choisir pour l'objet O_A ? Quelle classe Diffserv choisir pour l'objet O_B ? Il n'y a pas règle universelle pour réaliser le classement. Si l'on prend en considération le seul critère du délai (ce critère prime pour les applications interactives), il est alors recommandé d'associer l'objet O_A à la classe de trafic AF11 et l'objet O_B à la classe de trafic EF. Si l'on prend par contre en considération le seul critère du taux de perte (le critère du délai n'est pas déterminant pour les applications telles que la vidéo à la demande), il devient recommandé d'associer l'objet O_A à la classe de trafic AF13, l'objet O_B pouvant être associé aussi bien à la classe EF qu'à la classe AF11. Si l'on prend en considération plusieurs critères à la fois, le problème devient plus complexe d'autant que l'importance donnée à chaque critère va dépendre de l'application. Les AVO ne seront pas classés de la même manière selon qu'on privilégie la contrainte du temps, la contrainte de fiabilité ou une contrainte d'utilisateur.

C'est un problème de classification courant, qui consiste à déterminer, par rapport à un ensemble de classes définies (les classes Diffserv), l'appartenance d'un élément (un AVO) à l'une de ces classes en visant la plus forte ressemblance entre les caractéristiques de l'élément et celles des modèles imposés.

Ce type de problème peut se résoudre en utilisant les réseaux de neurones à base de fonctions radiales qui fournissent la « mesure de ressemblance ». Ils apportent une solution simple pour éviter l'explosion du nombre de paramètres.

4.2. *L'algorithme de classification des AVO*

Nous avons utilisé un algorithme de classification basé sur la technique de classification RBF (*Radial Basis Function*). RBF permet de trier un ensemble d'objets en utilisant une fonction basée sur un vecteur de distance. Nous exploitons la caractéristique du réseau de transport sous-jacent (Diffserv dans notre cas) pour marquer chaque objet avec une étiquette – le label – qui définit la classe de service de l'objet. Les objets sont classés un par un, indépendamment les uns des autres.

Considérons un objet MPEG-4 à classifier, caractérisé par un vecteur $\mathbf{X} \in \Re^n$. Ce vecteur $\mathbf{X}$ est défini par $(x_1, x_2, \ldots x_n)^T$, où chaque composante x_i représente une caractéristique de qualité de service associée à l'objet, parmi celles données table 1, section 2.3.1. Soit T, l'ensemble des étiquettes qu'un objet peut recevoir. Notre système de classification correspond à une fonction $C : \Re^n \rightarrow T$ qui assigne une étiquette $c \in T$ à chaque vecteur $\mathbf{X} \in \Re^n$ en se basant sur un ensemble de vecteurs caractéristiques des modèles $\{U_1, U_2, \ldots U_m\}$.

La figure 6 illustre le principe de fonctionnement du classificateur RBF. La méthode utilise un ensemble de blocs reliés entre eux pour former un réseau. Le chemin parcouru de l'entrée à la sortie du réseau est appelé neurone. Un bloc va

correspondre à un traitement particulier identifié par une fonction de transfert. Cette fonction de transfert doit prendre en compte les caractéristiques de l'objet MPEG-4 en entrée.

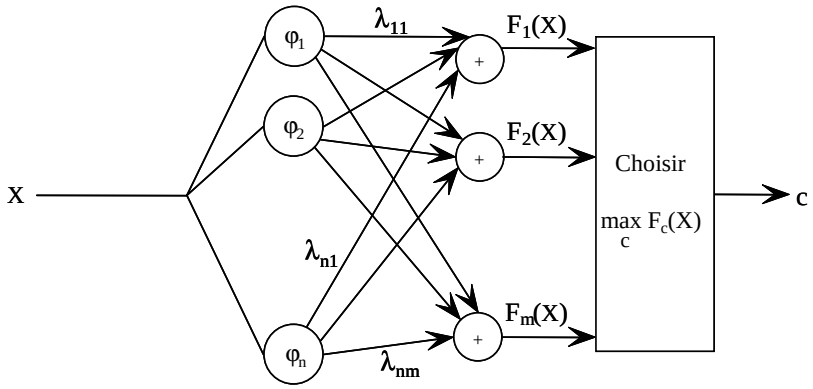


Figure 6. *Architecture du classificateur RBF.*

La fonction de transfert $\varphi_{j=1,2,\ldots n}$ est une fonction gaussienne centrée sur un point de l'espace d'entrée. Pour une entrée donnée, chaque sortie du bloc φ_j est calculée selon la formule [7] :

$$\varphi_j(x) = e^{-\frac{D^2}{2\sigma_j^2}} \qquad [7]$$

où $D = \|X - U_k\|^2$ est la distance entre le vecteur X et un vecteur modèle U_k pour la caractéristique de rang j, et σ_j est la déviation standard permettant de régler la sensibilité du résultat aux données d'entrées.

La fonction gaussienne permet aux neurones de ne répondre qu'à une petite région de l'espace d'entrée, région sur laquelle la gaussienne est centrée. La sortie du réseau est simplement une combinaison linéaire des sorties des neurones RBF multipliés par le poids λ_{jk} de leur connexion respective. Le poids permet à l'utilisateur de donner plus d'importance à certains paramètres de qualité de service qu'à d'autres.

Avec notre classificateur RBF, l'équation générale du résultat d'un neurone est donnée par la relation [8] :

$$F_k(X) = \sum_{j=1}^{n} \lambda_{jk} \varphi_j(X) \qquad [8]$$

Le classificateur choisit l'étiquette correspondant au résultat du neurone la plus grand.

4.3. *Application de l'algorithme sur le cas Diffserv*

Nous avons appliqué notre algorithme en modélisant la couche de transport Diffserv. Nous avons défini des vecteurs caractéristiques modélisant les classes de service Diffserv EF, AF11, AF12, AF13 et BE comme indiqué table 2, c'est-à-dire en respectant la recommandation ITU-T Y.1541. Chacune de ces classes fait référence à une étiquette correspondant à un DSCP particulier.

Nous avons considéré une classification des AVO MPEG-4 à partir de quatre caractéristiques de qualité de service qui font partie de celles indiquées dans la table 1 de la section 2.3.1. Le vecteur $X \in \Re^4$ caractérisant un AVO est ainsi représenté par $X = (MAX_DELAY, MAX_GIGUE, LOSS_PROB, PRIORITY)^T$, c'est-à-dire par le délai de transfert maximal des paquets, la gigue maximale tolérée qui est calculée par $MAX_GIGUE = MAX_DELAY - PREF_MAX_DELAY$, la taux de perte accepté et enfin la priorité du paquet. La priorité constitue une caractéristique d'utilisateur. Lorsqu'elle est spécifiée, elle exprime l'importance de l'AVO dans la qualité subjective de la scène audiovisuelle sur une échelle de 1 (la plus forte priorité) à 7 (la plus faible).

En prenant l'exemple que nous avons exposé dans la section 4.1, les objets O_A et O_B sont caractérisés respectivement par les vecteurs $X_A = (350, 100, 10^{-2}, MAXINT)^T$ et $X_B = (150, 50, 10^{-3}, MAXINT)^T$, sachant qu'une caractéristique qui n'est pas spécifiée est substituée par une « grande valeur », la constante $MAXINT = 2^{32} - 1$. De la même manière, l'ensemble des vecteurs modèles est défini par $\{U_1, U_2, U_3, U_4, U_5\}$, avec $U_1 = (100, 50, 10^{-3}, 1)^T$ caractéristique de la classe EF, $U_2 = (250, 50, 10^{-3}, 3)^T$ de la classe AF11, $U_3 = (300, MAXINT, 2.10^{-3}, 5)^T$ de la classe AF12, $U_4 = (400, MAXINT, 3.10^{-3}, 7)^T$ de la classe AF13 et $U_5 = (MAXINT, MAXINT, MAXINT, MAXINT)^T$ de la classe BE.

Comme les différentes composantes des vecteurs peuvent avoir des ordres de grandeur qui n'ont rien à voir, la « mesure de ressemblance » est réalisée sur les vecteurs normalisés. La table 4 présente les « mesures de ressemblance » obtenues en résultat de l'algorithme, en fixant les valeurs des poids $\lambda_{1k} = \lambda_{2k} = \lambda_{3k} = 1$ et $\lambda_{4k} = 0$ puisque la priorité n'est pas prise en compte dans cet exemple.

	EF	**AF11**	**AF12**	**AF13**	**Best Effort**
O_A (H263)	2,3243	2,5095	1,7160	1,7726	0
O_B (G723)	2,9460	2,8057	1,2131	0,2847	0

Table 4. *Calcul de la sortie du réseau de neurones RBF.*

L'étiquette choisie pour marquer un AVO sera celle de la classe ayant la valeur de ressemblance la plus grande, et en conséquence, l'objet O_A sera marqué de la classe AF11 et O_B de la classe EF.

En ce qui concerne l'évaluation du temps d'exécution de notre algorithme, et à titre indicatif, la classification de 10 AVO est traitée en 10 millisecondes environ et celle de 100 AVO est calculée en 30 millisecondes. La complexité de l'algorithme de classification est $O(n^2)$. Une évaluation de ses performances est présentée en détail dans (Ahmed *et al.*, 2003).

La phase d'apprentissage reste à faire, elle consiste à calculer les pondérations λ_{jk} optimales des différentes liaisons du réseau de neurones, c'est-à-dire identifier les poids sur les caractéristiques de qualité de service qui vont donner la meilleure classification, en utilisant des échantillons de référence. Cela relève d'une démarche d'optimisation qui va dépendre à la fois du contexte d'application, du type et des paramètres de compression. Elle sort du cadre de cet article.

5. Evaluation de performance

Les performances de notre technique de protection des objets audiovisuels MPEG-4 ont été évaluées par simulation avec l'outil ns2 (Fall *et al.*, 2002). Nous avons développé deux agents ns2, l'un jouant le rôle d'un serveur MPEG-4 et l'autre jouant le rôle d'un client. Des fichiers contiennent les traces du trafic particulier de chaque AVO. Le serveur lit ces traces pour envoyer les flux audiovisuels au client en respectant pour chaque paquet les indications relatives à sa date de production, à la taille de son champ de données et à sa classe de service Diffserv. Nous présentons dans cette section la topologie du réseau que nous avons simulée, les caractéristiques des objets audiovisuels MPEG-4 utilisés dans le scénario, puis nous faisons une analyse des résultats obtenus.

5.1. *Implantation du système de double protection*

Afin de valider notre proposition, nous avons implanté sous ns2 un serveur MPEG-4 (agent ns2) et un client MPEG-4 (agent ns). Des fichiers de trace de objets AVO MPEG-4 sont enregistrés sur le disque pour simuler une vraie scène MPEG-4. Le module « lecture et pompage » permet de lire les fichiers de trace pour générer les flux multimédia. Le module « classification » permet de donner une priorité au flux multimédia en fonction de ses critères de qualité de service. Ce module est implanté en utilisant la technique du réseau de neurone RBF. Le module FEC applique une protection inégale aux erreurs. L'ensemble des flux générés sera transmis dans des sessions RTP. Le module « marqueur » s'occupe du marquage des paquets de données avec un code Diffserv pour avoir un certain niveau de qualité de

service à l'intérieur du réseau. La figure 7 montre les différents modules développés sous ns2.

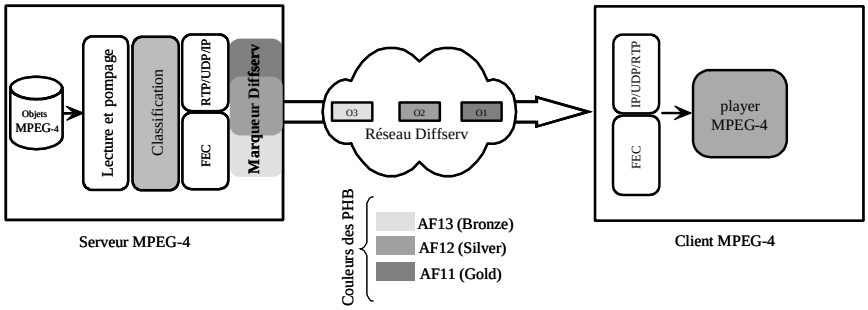


Figure 7. *Architecture du système de double protection.*

5.2. *La topologie du réseau*

L'architecture que nous avons définie pour les simulations est présentée sur la figure 8. Elle représente un serveur MPEG-4 rattaché au nœud S qui fournit un service de transport de données en point à point à un client MPEG-4 rattaché au nœud C.

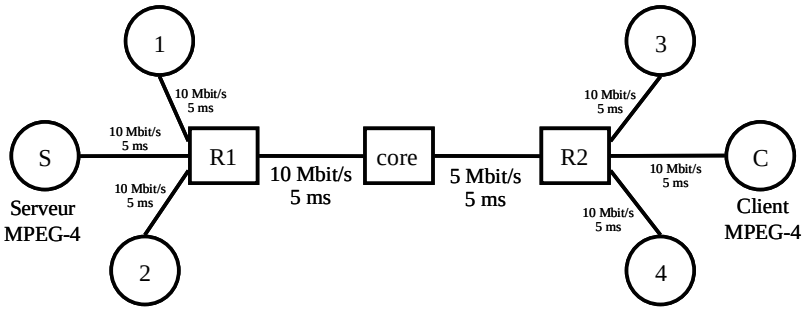


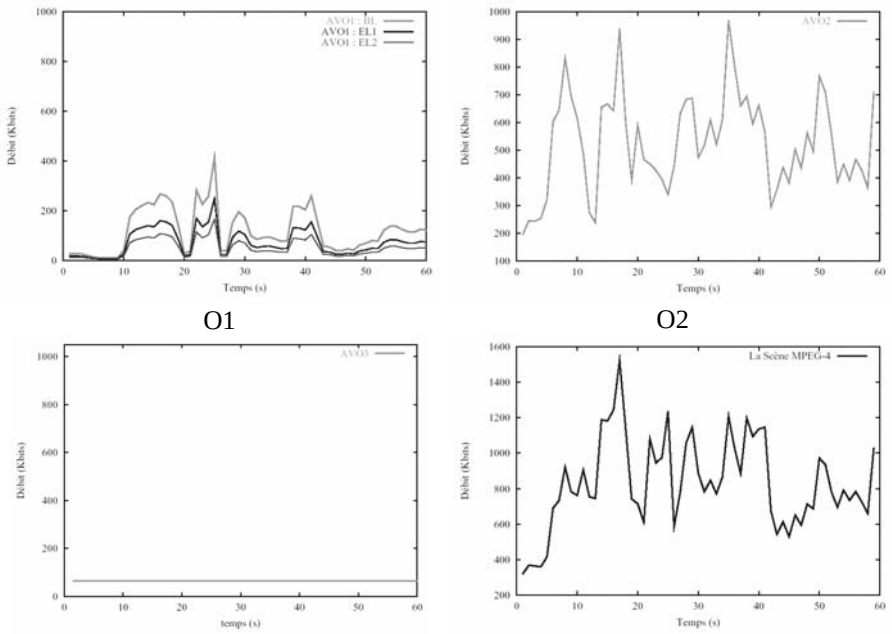
Figure 8. *Topologie du réseau définie pour les simulations.*

Les nœuds 1, 2, 3 et 4 vont générer du trafic de fond pour charger la liaison entre les routeurs R1 et R2. La charge est fixée par configuration. Elle est constituée à la fois par du trafic à débit constant (CBR) basé sur un transport UDP et du trafic de type transfert de fichier (FTP) basé sur TCP, dont le débit est fonction du RTT (*Round Trip Time*) et du taux de pertes du réseau selon le modèle donné par (Padhye *et al.*, 1998). Les sources FTP ont toujours des données à envoyer sur la durée de la simulation, les paquets ayant une taille maximale que nous avons fixée à 1000 octets. Les paquets sont immédiatement acquittés quand ils arrivent au récepteur. R1 est un

routeur de bordure (*Edge router*) du réseau Diffserv. Il prend en charge la classification et le marquage des paquets liés au trafic de fond entrants dans le domaine Diffserv. Il utilise l'algorithme TRTCM pour marquer ces paquets avec une distribution uniformément répartie sur les différentes classes du service AF. Quant aux paquets liés au trafic audiovisuel MPEG-4, ils sont marqués à la source par le serveur. En effet, c'est lui qui connaît les caractéristiques des AVO et qui effectue leur classification en appliquant l'algorithme présenté dans la section 4. Le routeur central (*Core Router*) forme le cœur du réseau Diffserv. Il examine la marque des paquets qui arrivent à lui et applique la politique de traitement associée à leur classe de service.

5.3. *Le trafic audiovisuel MPEG-4*

Le trafic MPEG-4 que nous avons représenté est composé de trois objets audiovisuels. La figure 9 présente le débit instantané de ces trois AVO sur la durée de la scène qui fait une minute. La scène MPEG-4 complète est envoyée sur le réseau avec un débit moyen de 770 kbps. Le pic le plus élevé atteint 1460 kbps.



O1 O2

O3 La scène MPEG-4 complète

Figure 9. *Débit de sortie instantané des AVO.*

Le premier objet, O1, est un flux vidéo cadencé au rythme de 25 fps et compressé à débit variable. Il est codé selon une représentation sur trois niveaux, BL (*Base Layer*), EL1 (*Enhancement Layer 1*) et EL2 (*Enhancement Layer 2*), qui correspondent à une scalabilité du rapport signal sur bruit. Pour qu'une image soit présentable, le récepteur a besoin de récupérer sa représentation BL. Les deux autres éléments codés EL1 et EL2 de l'image vont servir au raffinement de l'image présentée. Ils n'ont aucune utilité au récepteur si l'élément codé BL n'a pas été reçu.

Le deuxième objet, O2, est aussi un flux vidéo cadencé à 25 fps et compressé à débit variable. Il est codé sur trois niveaux, BL, EL1 et EL2. Mais ces trois niveaux correspondent cette fois à une scalabilité temporelle. Les images codées en mode Intra forment les éléments du niveau BL. La distance entre deux images Intra est configurée à 12. Les niveaux de raffinement EL1 et EL2 sont constitués respectivement des images codées en mode Prédit et celles codées en mode Bidirectionnel. Dans notre configuration, les images Intra et Prédite sont intercalées par deux images Bidirectionnelles. La séquence vidéo est une répétition du motif d'images IBBPBBPBBPBB.

Enfin, le troisième objet, O3, est un flux audio compressé à un débit constant de 64 kbps. Chaque paquet de ce flux embarque un échantillon de son indépendant des autres. Ils ont tous la même importance pour la qualité du signal reconstitué au récepteur.

5.4. *Les scénarios exécutés*

Nous avons exécuté quatre scénarios de simulation pour étudier les performances de notre technique de protection sous des angles différents. Ces scénarios sont définis de la manière suivante :

- Scénario n°1 : il applique notre technique de protection différenciée des AVO combinant les mécanismes FEC et Diffserv.
- Scénario n°2 : les AVO sont protégés seulement par Diffserv, en appliquant l'algorithme de classification présenté dans la section 4. Les flux audiovisuels n'embarquent pas de paquets de redondance sur les éléments d'information les plus importants.
- Scénario n°3 : les AVO sont protégés seulement par FEC, en appliquant le mécanisme que nous avons présenté dans la section 3. Les flux audiovisuels embarquent donc des paquets de redondance. Le réseau fournit un service unique de type *best-effort* pour tous les paquets du trafic audiovisuel comme du trafic de fond.
- Scénario n°4 : les AVO sont transportés sur le réseau sans aucune protection du tout. Le réseau fournit alors un service unique de type *best-effort*.

Dans le cas où le scénario utilise Diffserv, la configuration du domaine est réalisée conformément au modèle présenté dans la section 4.3.

Quand TAPIOCA est utilisé dans les simulations pour protéger les objets vidéo, la consigne fixée pour sur chacune des trois couches BL, EL1 et EL2 sera $R = [0.2, 0.1, 0.0]$, et cela pour l'un et l'autre des objets O1 et O2. La consigne R a été calculée en fonction de la priorité du flux qui a été choisie par la couche de classification. Plus le flux est prioritaire, et plus le taux de redondance est important. Dans le cas des flux hiérarchiques, la couche BL est plus prioritaire que la couche EL1, elle-même étant prioritaire par rapport à EL2. Pour cette raison, on va insérer dans la couche BL une quantité de redondances plus élevée que dans la couche EL1. La couche EL2 ne contient aucune information de redondance. Cette consigne a été choisie par expérience. Ce n'est pas une consigne optimisée (cela sort du cadre de cet article) mais elle va donner de bons résultats.

5.5. *Les résultats*

Nous devons montrer que notre solution technique, qui associe deux systèmes de protection, l'un basé sur les FEC et l'autre basé sur Diffserv, les combine efficacement. Nous avons donc exploité les mesures obtenues par simulation pour faire trois analyses de performance.

La première analyse concerne la performance intrinsèque du système de protection par FEC présenté dans la section 3. Il s'agit de comparer les résultats du scénario n°3 (avec protection par FEC) et du scénario n°4 (sans protection). La figure 10 montre ces résultats.

L'axe des abscisses indique le débit fixé pendant la durée d'une simulation pour le trafic de fond. L'augmentation du trafic de fond va avoir tendance à congestionner le chemin entre le routeur R1 et R2 et provoquer des pertes de paquets. L'axe des ordonnées indique le taux d'images et d'échantillons audio décodés, donc présentables au récepteur. Ce taux est calculé sur l'ensemble des trois objets O1 (1501 images), O2 (1501 images) et O3 (480 échantillons audio) qui composent la scène audiovisuelle, ce qui fait 3482 objets à décoder au total. Pour O3, quand un paquet contenant un échantillon audio est perdu, alors cet échantillon ne peut pas être décodé. Pour O1, quand un morceau d'image est perdu, l'image reste décodable s'il s'agit d'un morceau de niveau EL1 ou EL2. L'image ne peut pas être décodée sinon. Pour O2, la perte d'un morceau d'image a des effets plus variables. S'il s'agit d'un morceau de niveau EL2 (image de type B), alors seule cette image ne pourra pas être décodée. S'il s'agit d'un morceau de niveau EL1 (image de type P), alors plusieurs images ne pourront pas être décodées en raison des corrélations temporelles, entre cinq et onze images selon les cas. Enfin, s'il s'agit d'un morceau d'image de niveau BL (image de type I), alors c'est quatorze images qui ne pourront pas être décodées.

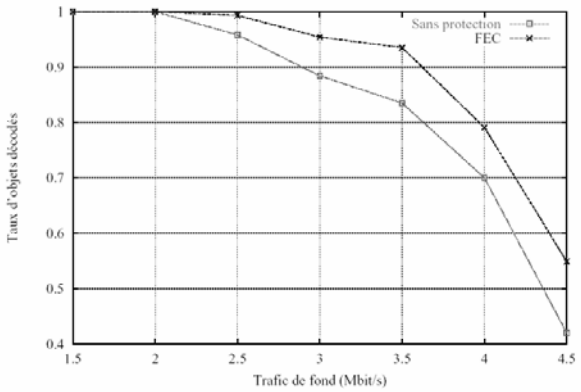
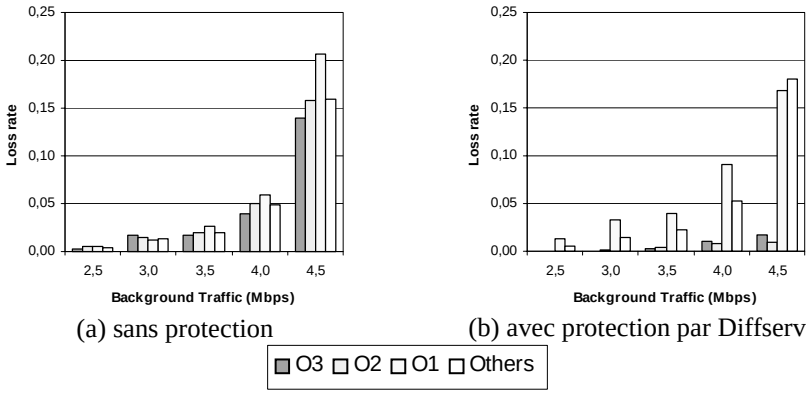


Figure 10. *Taux d'objets décodés* vs. *Charge du trafic de fond.*

Comme prévu, la figure 10 montre que le taux d'objets décodés diminue quand la charge du réseau augmente puisque le nombre de paquets perdus dans le réseau augmente aussi. Dans le scénario n°3, les objets vidéo O1 et O2 sont protégés par FEC. Le trafic audiovisuel est donc plus élevé (de 7 %) que dans le scénario n°4 pour un trafic de fond similaire. On va donc avoir globalement un plus grand nombre de paquets perdus dans le scénario n°3 que dans le scénario n°4. Toutefois, la protection par FEC va permettre de reconstruire les paquets perdus quand ils contiennent des morceaux d'image de niveau BL. En conséquence, le récepteur arrive à décoder un plus grand nombre d'objets. Cela est clairement apparent sur la figure 10 où le scénario n°3 donne toujours de meilleurs résultats. Le gain se stabilise à une valeur d'environ $1/10$.

La deuxième analyse concerne l'apport du système de protection par Diffserv présenté dans la section 4. Il s'agit de comparer les résultats du scénario n°2 (avec protection par Diffserv) et du scénario n°4 (sans protection). La figure 11 montre ces résultats. L'axe des abscisses indique le débit fixé pendant la durée d'une simulation pour le trafic de fond. L'axe des ordonnées indique par quatre barres les taux de perte de paquets relatifs respectivement au trafic O1, au trafic O2, au trafic O3 et au trafic de fond. Pour le scénario n°4, la partie (a) de la figure montre que le taux de pertes est approximativement uniforme pour les quatre flux quelque soit la charge du réseau. C'est dû à l'utilisation d'un seul service de type *best-effort* pour tous les flux. Pour le scénario n°3 où les services utilisés sont différenciés, les flux les plus importants, le flux audio O3 et le flux vidéo O2, sont privilégiés par Diffserv. Les pertes de paquets sont réparties majoritairement sur les flux les moins prioritaires. Le système de classification des AVO est donc bénéfique pour la qualité de la scène audiovisuelle qui est restituée au récepteur.



(a) sans protection (b) avec protection par Diffserv

Figure 11. *Taux de perte de paquets* vs. *Charge du trafic de fond.*

Ces deux premières analyses ont montré que chaque système de protection utilisé séparément améliore de la qualité du service de transport des objets audiovisuels MPEG-4. La dernière analyse va étudier leur aptitude à opérer de manière conjointe pour constituer une technique de protection combinée. Il s'agit de comparer les résultats du scénario n°4 (sans protection), du scénario n°2 (protection par Diffserv) et du scénario n°1 (protection combinée). La figure 12 montre ces résultats.

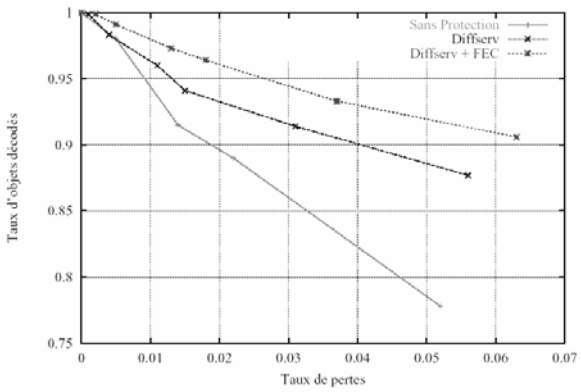


Figure 12. *Comparaison de performance des trois scénarios.*

Pour que les scénarios soient comparables, l'axe des abscisses va indiquer, non plus le débit fixé sur le trafic de fond, mais le taux de pertes relatif au trafic MPEG-4, c'est-à-dire sur O1, O2 et O3 à la fois. L'axe des ordonnées indique le taux d'images et d'échantillons audio décodés, donc présentables au récepteur. Les performances obtenues sont nettement à l'avantage de notre technique combinée. En comparaison à Diffserv utilisé seul, la technique combinée apporte un gain significatif, qui s'accroît légèrement quand les pertes subies par le trafic MPEG-4

augmentent. Les deux systèmes de protection sont donc bien adaptés pour s'assembler.

6. Conclusion

Dans cet article, nos avons proposé une technique de double protection des objets audiovisuels MPEG-4. Notre technique exploite les codes correcteurs Reed-Solomon et le service Diffserv de IP. Les objets audio et vidéo issus du codeur MPEG-4 sont classés en fonction de leur importance dans la scène et de leur niveau de scalabilité. Chaque flux AVO est traité en fonction d'une priorité calculée par notre système. Un AVO plus important qu'un autre sera mieux protégé, en lui accordant un taux de redondances plus élevé et une classe Diffserv fournissant une meilleure qualité de service dans le réseau de transport. Le système proposé permet de mieux tenir compte des caractéristiques des AVO et de leur pertinence dans la scène. Les résultats montrent qu'avec notre technique de double protection, les flux MPEG-4 sont mieux protéger contre les pertes de paquets, la qualité du service rendu étant meilleure par rapport à une technique sans ou avec simple protection.

7. Bibliographie

Ahmed T., Mehaoua A., Buridant G., « Implementing MPEG-4 Video on Demand over IP Differentiated Services », *Proc. of the 6th IEEE GLOBECOM Symposium*, pp. 2489-2493, nov. 2001.

Ahmed T., Nafaa A., Mehaoua A., « An Object-Based MPEG-4 Multimedia Content Classification Model for IP QoS Differentiation », to appear in *proc. of 8th IEEE Symposium on Computers and Communications*, june 2003.

Albanese A., Blömer J., Edmonds J., Luby M., Sudan M., « Priority Encoding Transmission », *Proc. of the 35th IEEE Symposium on Fundations of Computer Science*, pp. 604-612, nov. 1994.

Albrecht M., Köster M., Martini P., Frank M., « End-to-end QoS Management for Delay-sensitive Scalable Multimedia Streams over Diffserv », *Proc. of the 25th Annual IEEE Conference on Local Computer Networks*, pp. 314-323, nov. 2000.

Blahut R. E., *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, MA, 1983.

Blake S., Black D., Carlson M., Davies E., Wang Z., Weiss W., An Architecture for Differentiated Services, RFC 2475, IETF, dec. 1998.

Bolot J. C., Vega-Garcia A., « Control Mechanisms for Packet Audio in the Internet », *Proc. of the 15th IEEE INFOCOM*, pp. 232-239, april 1996.

Boutremans C., Le Boudec J. Y., « Adaptive Delay aware Error Control for Internet Telephony », *Proc. of the 2nd IP-Telephony workshop*, pp. 81-92, april 2001.

Braden R., Zhang L., Berson S., Herzog S., Jamin S., Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, RFC 2205, IETF, sept. 1997.

Fall K., Varadhan K., The ns Manual (Formerly ns Notes and Documentation), on-line: http://www.isi.edu/nsnam/ns/, april 2002.

Fang W., Seddigh N., Nandy B., A Time Sliding Window Three Colour Marker (TSWTCM), RFC 2859, IETF, june 2000.

Girod B., Stuhlmüller K., Link M., Horn U., « Packet Loss Resilient Internet Video Streaming », *Proc. of the SPIE Conference on Visual Communication and Image Processing*, pp. 833-844, jan. 1999.

Hardman V., A. Sasse M. A., Handley M., Watson A., « Reliable Audio for Use over the Internet », *Proc. of the International Networking Conference*, pp. 171-178, June 1995.

Heinanen J., Baker F., Weiss W., Wroclawski J., Assured Forwarding PHB Group, RFC 2597, IETF, june 1999.

Heinanen J., Guerin R., A Two Rate Three Color Marker, RFC 2698, IETF, sept. 1999.

ISO/IEC 14496-1, Coding of audio-visual objects - Part 1: Systems, Final committee draft, may 1998.

ISO/IEC 14496-2, Coding of audio-visual objects - Part 2: Visual, Final committee draft, may 1998.

ISO/IEC 14496-3, Coding of audio-visual objects - Part 3: Audio, Final committee draft, may 1998.

ITU-T Recommendation Y.1541, Internet Protocol Communication Service – IP Performance and Availability Objectives and Allocations, Oct 2001.

Jacobson V., Nichols K., Poduri K., An Expedited Forwarding PHB, RFC 2598, IETF, june 1999.

Leicher C., Hierarchical Encoding of MPEG Sequences Using Priority Encoding Transmission (PET), Technical Report TR-94-058, ICSI, Berkeley, 1994.

Lin J. C., Costello D. J., *Error Correcting Coding: Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1983.

Pahdye J., Firoiu V., Towsley D., Kurose J., « Modeling TCP Throughput: A Simple Model and its Empirical Validation », *Proc. of the ACM SIGCOMM*, pp. 303-314, August 1998.

Pillai R. R., Prabhakaran B., Qiang Q., « A Forward Error Recovery Technique for MPEG-II Video Transport », *Proc. of the 7th ACM International Conference on Multimedia*, pp. 59-61, nov. 1999.

Poli A., Huguet L., *Codes Correcteurs, Théorie et Applications*, Edition Masson, 1988.

Rizzo L., « Effective Erasure Codes for Reliable Computer Communication Protocols », *Computer Communication Review*, vol. 27(2), pp. 24-36, April 1997.

Shao H. R., Zhu W., Zhang Y. Q., « Scalable Object-based video Multicasting over the Internet », *Proc. of the IEEE International Conference on Image Processing*, sept. 2000.

Shin J., Won Kim J., Jay Kuo C. C., « Content-Based Packet Video Forwarding Mechanism in Differentiated Service Networks », *Proc. of the 10th IEEE Packet Video Workshop*, may 2000.

Wroclawski J., The use of RSVP with IETF Integrated Services, RFC 2210, IETF, Sept. 1997.

8. Annexe

Cette annexe décrit la structure d'un descripteur d'objet MPEG-4. Deux objets sont décrits : le premier représente un flux vidéo de type H263 et le deuxième un flux audio de type G723.

```
ObjectDescriptor {
             objectDescriptorID 2
             es_descriptor ES_Descriptor {
                  es_Number 1
                  decConfigDescr DecoderConfigDescriptor {
                       streamType 5    // VisualStream
                       bufferSizeDB 16000
                       specificInfo H263
                  }
                  alConfigDescr ALConfigDescriptor {
                       useAccessUnitEndFlag TRUE
                       useRandomAccessPointFlag TRUE
                       useTimeStampsFlag TRUE
                       timeStampResolution 1000
                       timeStampLength 10
                  }
                  QoS_Descriptor {
                       streamPriority 0
                       max_delay   350
                       pref_max_delay 250
                       loss_prob  0.01
                  }
           }
}

ObjectDescriptor  {
             objectDescriptorID 3
             es_descriptor ES_Descriptor {
                  es_Number 1
                  decConfigDescr DecoderConfigDescriptor {
                       streamType 6    // AudioStream
                       bufferSizeDB 500
                       specificInfo G723
                  }
                  alConfigDescr ALConfigDescriptor {
```

```
                    timeStampResolution 1000
                    accessUnitRate 30
                }
                IPI_Descriptor {
                    IPI_ES_Id 0
                }
                QoS_Descriptor {
                      streamPriority 0
                      max_delay   150
                      pref_max_delay 100
                      loss_prob  0.001
                }
            }
}
```