

Communications multimédias  
**MPEG scalability & video on demand**  
**Résumé du projet (VLC-SMPTE 2022 CoPR3)**

Résumé du travail personnel de  
**M. David Fischer ingénieur HES en télécommunications**

dirigé par prof. Andrés Revuelta  
laboratoire de télécommunications de hepia

---

**Descriptif**

La popularisation des techniques de transmission de type *xDSL* permettront à beaucoup d'applications récentes de connaître un regain d'intérêt. Ce sera certainement le cas d'IPTV, la télévision sur IP.

Afin de garantir une qualité suffisante des programmes de TV transmis par le réseau IP, les milieux professionnels concernés par la TV numérique étudient la possibilité d'ajouter un système de compensation d'erreurs de transmission au niveau applicatif, c'est-à-dire en sus des corrections prévues par la norme MPEG-2, travaillant sur plusieurs niveaux de codage des programmes vidéo.

Plusieurs normes étant actuellement pressenties, il serait opportun d'en étudier la mise en application pratique à l'aide d'un lecteur vidéo couramment employé sur le marché des PC.

**Objectif**

Réalisation d'un décodeur de type *SMPTE 2022* pour la plateforme VLC de *Videolan*.

**Contraintes techniques**

- La matrice de correction devra être extensible ;
- Les débits à envisager sont très variables (entre 100 kbps et 100 Mbps) ;
- Environnement de développement LINUX ;
- Cible de type *multi-operating system & multi-architectures* ;
- Numéros des ports utilisés pour les flux de correction programmables.

**Travail réalisé**

- Réalisation du cahier des charges.
- Etude des principales normes existantes.
- Etude de la norme SMPTE 2022.
- Prise en main de la plateforme *open source* de streaming vidéo *VideoLan*.
- Etude des modules nécessaires à l'utilisation des normes proposées.

**En cours**

- Implémentation des modules nécessaires à l'utilisation des normes proposées.

**A faire**

- Validation à l'aide de flux de test conformes à la norme SMPTE 2022.

**Remarque**

Le travail finale réalisé est destiné dans la mesure du possible à être intégré au projet VLC et redistribué à la communauté *OpenSource* selon la licence GPL mise en place par *Videolan*.

## Possibilités offertes par le projet VLC-SMPTE, modules de démonstration

✚ Tous les modules de démonstration utilisent la librairie fournie libSmp2-2022-CoPR3.a

Le module *FecGenerator* segmente le fichier source des paquets média. Il produit un fichier comprenant des paquets media et des paquets *FEC* calculés. La méthode d'encapsulation est propriétaire, de type RTP simplifié ;

Le module *ErrorsGenerator* reçoit en entrée le fichier produit par *FecGenerator*. Il simule la perte de l'ordre d'arrivée des paquets ainsi que la perte de certains paquets (erreurs ou perte de paquets par le réseau). Il produit un fichier de format équivalent à celui produit par *FecGenerator*.

Le module *FecDecoder* prend en charge un fichier de format *FecGenerator* et corrige à l'aide des paquets FEC les données utiles. Il peut produire jusqu'à trois fichiers de format identiques au fichier source donné à *FecGenerator* :

1. sans correction aucune ;
2. avec correction produite par l'algorithme optimisé appelé **David** ;
3. avec correction produite par un algorithme simple appelé **Brute**.

Ces modules permettent de simuler divers cas de figure :

- *FecGenerator* permet de jouer avec la taille et le décalage de la matrice de FEC, ainsi que sur la taille du payload encapsulé en pseudo RTP ;
- *ErrorsGenerator* permet de changer le taux et la taille moyenne d'une rafale de pertes, ainsi que la probabilité et l'étendu des dérangements de séquence RTP ;
- *FecDecoder* permet de modifier la taille de la fenêtre du buffer de média (jusqu'aux limites de la mémoire physique à disposition) et les délais entre traitements par l'algorithme **Brute** (si activé).

## Solution proposée

La librairie est facilement intégrable car le code source est structuré et considérablement commenté (document *Doxygen* enrichi des commentaires du code source, guide d'utilisateur et rapport final).

J'ai implémenté un tampon média, différent de celui proposé par VLC (qui est une liste chaînée), sous la forme d'un arbre rouge-noir optimisant les temps d'accès non séquentiels, tels ceux effectués par l'algorithme de FEC.

J'ai aussi implémenté un tampon de FEC qui stocke les paquets de FEC et les lie aux paquets média sous la forme de trois arbres rouge-noirs liés. Ceux-ci permettent d'optimiser le temps de consultation du tampon lors de l'arrivée d'un paquet média, du traitement de FEC et de cascades de FEC.

## Non réalisé

- encapsulation et décapsulation les flux média et FEC dans un flux RTP normalisé ;
- gestion des sockets réseau en vue du transport et de la réception des flux multimédia ;
- intégration dans un *player* (par exemple VLC).

La plupart des *players* gèrent ces fonctions.

## Conclusion

Les modules de démonstration permettent de tester l'efficacité de la librairie fournie. J'espère que le rapport final produit permettra d'intégrer aisément cette librairie à n'importe quel type de *player* multimédia.