Mémoire d'Habilitation à Diriger Les Recherches

Institut National Polytechnique de Toulouse

Spécialité : Informatique et Réseaux

présenté par

Jérôme Lacan

Codage dans les réseaux

soutenue le 5 décembre 2008 devant le jury composé de

Marie-Laure Boucheret	Professeur, INP Toulouse	Rapporteur
Pierre Duhamel	Directeur de Recherche CNRS, Supélec	Rapporteur
Muriel Médard	Professeur, MIT, USA	Rapporteur
Michel Diaz	Directeur de Recherche CNRS, LAAS	Examinateur
Vincent Roca	Chargé de Recherche INRIA	Examinateur
Patrick Sénac	Professeur, ISAE	Examinateur
Francis Castanié	Professeur, INP Toulouse	Correspondant INPT

Institut National Polytechnique de Toulouse

Résumé

La fiabilité des transmissions est un des principaux problèmes qu'ont à résoudre les concepteurs de systèmes de communication. Parmi les mécanismes de fiabilité, les codes correcteurs d'erreurs permettent de protéger les données transmises de manière pro-active contre les erreurs de transmission. Historiquement, ces codes étaient principalement utilisés sur la couche physique. L'augmentation de la puissance des machines a permis de les intégrer sur les couches hautes des piles de protocoles de communication depuis le milieu des années 90.

Cette intégration a ouvert de nouvelles problématiques de recherche. L'une d'entre elles est la conception de codes adaptés aux contraintes des systèmes dans lesquels ils sont intégrés. La première partie des travaux présentés dans ce mémoire concerne ce thème. Nous avons en particulier fait plusieurs propositions pour améliorer les vitesses de codage et de décodage en logiciel des codes MDS (dont les représentants les plus connus sont les codes de Reed-Solomon). Une RFC est en cours de publication à l'IETF sur ce sujet. Une modification de la structure de ces codes nous a permis de les adapter aux transmissions multimédia en introduisant des niveaux de protection variables entre les symboles d'un même mot de code. Enfin, en relâchant au maximum leur structure, nous avons construit un système de codage "à la volée" s'intégrant particulièrement bien dans des protocoles de communication classiques.

La seconde thématique concerne la distribution des mécanismes de fiabilité et de la redondance sur les différentes couches protocolaires. Nous avons par exemple étudié la possibilité de laisser des paquets corrompus remonter les couches pour être corrigés ou simplement traités par les couches hautes. Lors de collaborations avec le CNES et Thalès Alénia Space, nous avons étudié le cas des transmissions multimédia de satellites vers des mobiles (SDMB et DVB-SH) en analysant les différentes solutions de distribution de la redondance sur les couches physique, liaison et les couches hautes. Différentes applications de ce travail ont débouché sur le dépôt de 2 brevets.

Le dernier volet de nos recherches concerne les applications des codes à effacement. Nous avons présenté des contributions sur l'utilisation de codes à effacement dans les réseaux pair-à-pair. En particulier, dès 2002, nous avons montré comment les codes permettaient d'accélérer les temps de téléchargement dans ce type de réseau. Nous avons aussi proposé une application particulière du codage réseau en montrant que cette technique peut réduire les bornes des délais de bout-en-bout des paquets dans des réseaux fournissant des garanties sur la qualité de service.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Curriculum Vitae

1.1 Etat civil

Prénom, Nom	Jérôme Lacan
date et lieu de naissance	4 août 1970 à Rodez (Aveyron)
Nationalité	Français
Situation familiale	marié, 2 enfants
adresse personnelle	37, avenue de Courrège, 31400 Toulouse
adresse professionnelle	(postale) 10 av. Edouard Belin - BP 54032 31055 Toulouse Cedex 4 (physique) 1, place E. Blouin, 31056 Toulouse cedex 5
Téléphone (professionel)	+33 5 61 33 92 20
Adresse de messagerie	jerome.lacan@isae.fr
Site Web	http ://dmi.ensica.fr/auteur.php3 ?id_auteur=5
Laboratoires	- Département Mathématiques, Informatique et Automatique de l'Institut Supérieur de l'Aéronautique et de l'Espace - Laboratoire de Télécommunications Spatiales et Aéronautiques (TéSA) - Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), CNRS, groupe Outils Logiciels pour la Communication (OLC)

1.2 Expérience professionnelle

2001-2008	Maître de Conférences détaché à l'ENSICA puis à l' Institut Supérieur de l'Aéronautique et de l'Espace (ISAE).
1999-2001	Maître de Conférences à l'Université de Franche-Comté (IUT de Belfort-Montbéliard)
1998-1999	Enseignant-Chercheur Contractuel à l'Université de Technologie de Belfort-Montbéliard
1996-1998	A.T.E.R. à l'IUT d'Informatique de Belfort-Montbéliard
1993-1996	Vacataire à l'IUT A d'Informatique de Toulouse et à l'Université Paul Sabatier de Toulouse

1.3 Cursus Universitaire

déc. 1997 Doctorat de l'université Paul Sabatier de Toulouse, (Mention Très Honorable)
Spécialité : Informatique fondamentale
Titre : Contribution à l'Etude des Codes Correcteurs d'Erreurs
Jury :

M. VIGNOLLE	Professeur de l'Univ. Toulouse III	Président
M. GLAVIEUX	Professeur à L'ENST de Bretagne	Rapporteur
M. BENETEAU	Professeur de l'INSA de Toulouse	Rapporteur
M. PYNDIAH	Maître de conf. ENST Bretagne	Examinateur
M. POLI	Professeur de l'Univ. Toulouse III	Dir. Recherche

juin 1994 DEA Informatique Fondamentale et Parallélisme, Université Paul Sabatier
Effectué au Laboratoire IRIT / AAECC, Toulouse, sous la direction du Pr. A. Poli.
Titre : Combinaisons de Codes Correcteurs d'Erreurs

juin 1993 Maîtrise d'Ingénierie Mathématique, Université Paul Sabatier
juin 1992 Licence de Mathématiques, Université Paul Sabatier,
juin 1991 Deug A section Mathématiques, Mécanique, Physique, Informatique,
juin 1988 BAC C, Lycée Ste Marie (Rodez)

1.4 Encadrements d'étudiants

1.4.1 DEA/Master

Hanaa El-Natour	" Evaluation des Mécanismes de Transmission Multipoint Fiable dans les Réseau février 2003 - juin 2003
Ali Mahmino	"Codage réseau pour les systèmes embarqués", février 2004 - juin 2004.
Amine Bouabdallah	" protection inégale de données multimédia " février 2005- juin 2005

1.4.2 Thèses soutenues

Juan Cantillo "Fiabilité multi-couches pour les communications par satellite",
Dates : novembre 2004 - mai 2008.
Co-encadrant : Marie-Laure Boucheret
Participation à l'encadrement : 80%

1.4.3 Thèses en cours

Ali Mahmino	"Codage réseau et applications " Dates : octobre 2004 - décembre 2008 Co-encadrant : Christian Fraboul Participation à l'encadrement : 80%
Amine Bouabdallah	"Qualité de service et protection inégale de données multimédia " Dates : octobre 2005- mai 2009 Co-encadrant : Michel Diaz Participation à l'encadrement : 80%
Alexandre Soro	"Codage à effacements et compression d'entêtes dans les réseaux ". Dates : juin 2007- juin 2010 Co-encadrant : Olivier Besson Participation à l'encadrement : 80%

1.5 Enseignements

1.5.1 Avant 2001 :

1- Contenu
- Codes correcteurs d'erreurs, cryptographie, protection de l'information.
- Algorithmique, programmation objet de base.
- langages : C, CAML, Prolog, Fortran, Pascal, Maple, Delphi
- Algèbre linéaire, Probabilités.

2- Niveau du public concerné
- Université de Technologie, 1-2-4-5ème année
- Maîtrise d'Ingénierie Mathématique
- IUT Informatique 1ère, 2ème Année et Année Spéciale
- IUT Génie Civil 1ère Année
- DEUG B, 1ère Année

1.5.2 Depuis 2001 :

1- Contenu
- Programmation objet
- Systèmes multimédia
- Théorie de l'Information
- Introduction au codage et à la cryptographie

2- Niveau du public concerné
- ENSICA/ISAE : 1-2 ème année
- ENAC : 1ere année
- IUT Blagnac : licence professionnelle

Total des enseignements : environ 1700 h eq. TD.

1.6 Publications

1.6.1 Revues internationales et nationales

1. J. Lacan and E. Delpeyroux. Permutation Group of the q-ary Image of Some qm-ary Cyclic Codes. *"Finite Field : Theory, Applications, and Algorithms", Contemporary Mathematics*, 225 :165–176, 1998.
2. D. G. Arquès, J. Lacan, and C. J. Michel. Identification of Protein Coding Genes in Genomes with Statistical Functions Based on the Circular Code . *Biosystems*, pages 159–170, 2001.
3. J. Lacan and C. Michel. Analysis of a circular code model. *Journal of Theoretical Biology*, 213 :159–170, 2001.
4. J. Lacan and E. Delpeyroux. The q-ary image of some q^m-ary cyclic codes : Permutation group and soft decision decoding algorithm. *IEEE Transactions on Information Theory*, 48 :2069–2078, 2002.
5. J. Lacan and J. Fimes. Systematic mds erasure codes based on vandermonde matrices. *IEEE Communications Letters*, 8 :570– 572, September 2004.
6. Fabrice Arnal, Laurent Dairaine, Jérôme Lacan, and Gérard Maral. Cross-layer reliability management for multicast over satellite. *Computer Networks*, 48(1) :29–43, May 2005.
7. T. Pérennou, J. Lacan, and H. Elnatour. Evaluation de Mécanismes de Contrôle d'Erreur pour des Transmissions Multipoints sur des Réseaux de Mobiles. *Technique et Science Informatiques (TSI) 24/2005*, pages 865–886, 2005.
8. L. Dairaine, L. Lancérica, J. Lacan, and J. Fimes. Content-Access QoS in Peer-to-Peer Networks Using a Fast MDS Erasure Code. *Computer Communications*, 28(15) :1778–1790, september 2005.
9. J. Cantillo, J. Lacan and I. Buret. Cross-layer enhancement of error control techniques for adaptation layers of DVB satellites. *International Journal of Satellite Communications and Networking*, 24 :579–590, 2006.
10. A. Bouadallah and J. Lacan. Dependency-Aware Unequal Erasure Protection Codes. *Journal of Zhejiang University - Science A*, 7 :27–33, 2006.
11. A. Bouabdallah, M. Kieffer, J. Lacan, G. Sabeva, G. Scot, C. Bazile, and P. Duhamel. Evaluation of Cross-Layer Reliability Mechanisms for Satellite Digital Multimedia Broadcast. *Broadcasting, IEEE Transactions on*, 53(1) :391–404, March 2007.

1.6.2 Conférences internationales

12. J. Lacan and Emmanuelle Delpeyroux. A note on normal bases. In *AAECC*, pages 334–340, 1995.
13. J. Lacan. Systolic Binary BCH Decoder. In *ISITA'96, International Symposium on Information Theory and Applications*, 1996.
14. J. Lacan et E. Delpeyroux. Permutation Group of Expanded Cyclic Codes. In *4th International Conference On Finite Fields : Theory, Applications and Algorithms*, 1997.
15. J. Lacan. Diagonals of 2d-abelian codes. In *Information Theory Workshop, 1998*, pages 110–111, Jun 1998.

16. J. Lacan and Pascal Chatonnay. Search of optimal error correcting codes with genetic algorithms. In *Fuzzy Days*, pages 93–98, 1999.
17. E. Delpeyroux and J. Lacan. Permutation Soft Decision Decoding of Some Expanded Reed-Solomon Codes. In *Workshop on Coding and Cryptography 99*, 1999.
18. E. Delpeyroux and J. Lacan. Soft Decision Decoding Algorithm of Reed-Solomon Codes. In *SCI'2000/ISAS'2000*, 2000.
19. J. Lacan. Permutation group of some concatenated block codes. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, pages 33–, 2002.
20. J. Lacan, L. Lancérica, and L. Dairaine. When FEC speed up data access in p2p networks. In *IDMS'02 Conference (Interactive Distributed Multimedia Systems)*, 2002.
21. J. Lacan and J. Fimes. A construction of matrices with no singular square submatrices. In *Finite Fields and Applications : 7th International Conference, Fq7, Toulouse, France. Revised Papers. LNCS 2948/2004*, pages 145–147, May 2003.
22. L. Lancérica, L. Dairaine, and J. Lacan. Evaluation of content-acces qos for various dissemination strategies in peer to peer networks. In *11th IEEE International Conference on Networks ICON*, 2003.
23. F. de Belleville, L. Dairaine, C. Fraboul, and J. Lacan. Une approche hybride satellite/terrestre pour le transport fiable multipoint à grande échelle. In *Colloque Francophone sur l'Ingénierie des Protocoles*, 2003.
24. L. Dairaine, L. Lancérica, and J. Lacan. Enhancing peer to peer parallel data access with peerfect. In *Proceedings of the COST264 International Workshop on Networked Group Communications, NGC 2003*. LNCS, 2003.
25. J. Lacan, L. Lancerica, and L. Dairaine. Speedup of data access using error correcting codes in peer-to-peer networks. pages 471–, June-4 July 2003.
26. F. de Belleville, L. Dairaine, J. Lacan, and C. Fraboul. Reliable multicast transport by satellite : a hybrid satellite/terrestrial solution with erasure codes. In *IEEE conference on High Speed Networks and Multimedia Communications (HSNMC)*. Springer-Verlag, 2004.
27. J. Lacan and T. Pérennou. Amélioration de la fiabilité des transmissions point-à-point sur un canal à diffusion. In *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'05)*, mars 2005.
28. J. Lacan and T. Pérennou. Evaluation of Error Control Mechanisms for 802.11b Multicast Transmissions . In *Second International Workshop On Wireless Network Measurement (WinMee 2006)*, April 2006.
29. A. Bouabdallah and J. Lacan. Dependency-Aware Unequal Erasure Protection Codes. In *Proceedings of the* 15^{th} *Packet Video Workshop, Hanghzou, China*, April 2006.
30. J. Cantillo, J. Lacan and I. Buret. A CRC Usefulness Assessment for Adaptation Layers in Satellite Systems. In *Proceddings of AIAA's* 24^{th} *International Communications Satellite Systems Conference, San Diego, California*, June 2006.
31. J. Lacan. Fault-tolerant distributed computing scheme based on erasure codes. In *NOTERE, Toulouse*, Juin 2006.
32. A. Mahmino, J. Lacan, and C. Fraboul. Calculus of service guarantees for network coding. In *Proc. ISITA 2006*, October 2006.
33. Y. Oster, J. Lacan, and A. Duverdier. Benchmark of reed-müller codes for short packet transmission. In *25th AIAA International Communications Satellite Systems Conference, Seoul, Corée*, avril 2007.

34. J. Cantillo, J. Lacan, I. Buret and F. Arnal. Design Issues for the Generic Stream Encapsulation (GSE) of IP Datagrams over DVB-S2. In *Proceedings of the* 4^{th} *International Workshop on Satellite and Space Communications (IWSSC-07)*, 2007.
35. A. Soro, J. Lacan, E. Chaput, C. Donny, and C. Baudoin. Evaluation of a generic unidirectional header compression protocol. In *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*, pages 126–130, Sept. 2007.
36. Ali Mahmino, Jérôme Lacan, and Christian Fraboul. Enhancing guaranteed delays with network coding. In *International Conferences on Networking - Networking 2007*, pages 1229–1232, 2007.
37. J. Lacan and T. Perennou. Reducing Retransmissions in Point-to-point Satellite Transmissions. In *Proceddings of AIAA's* 25^{th} *International Communications Satellite Systems Conference, San Diego, California*, June 2008.
38. J. Lacan and E. Lochin. Rethinking reliability for long-delay networks. In *International Workshop on Satellite and Space Communications, 2008. IWSSC '08.*, octobre 2008.
39. M. Cunche, V. Savin, V. Roca, G. Kraidy, A. Soro, and J. Lacan. Low-rate coding using incremental redundancy for GLDPC codes. In *Fourth International Workshop on Satellite and Space Communications 2008*, Toulouse, France, 10 2008.
40. A. Mahmino, J. Lacan, and C. Fraboul. Guaranteed packet delays with network coding. In *First IEEE International Workshop on Wireless Network Coding : WiNC 2008*, pages 1–6, June 2008.
41. L. Zhang, P. Sénac, E. Lochin, J. Lacan, and M. Diaz. Cross-layer based erasure code to reduce the 802.11 performance anomaly : when FEC meets ARF. In *The 6th ACM International Workshop on Mobility Management and Wireless Access*, Vancouver, British Columbia, Canada, 10 2008.

1.6.3 Documents de standardisation

42. J. Lacan, V. Roca, J. Peltotalo, and S. Peltotalo. *Reed-Solomon Forward Error Correction (FEC), draft-ietf-rmt-bb-fec-rs-05*, 2008. Work in progress, draft-ietf-rmt-bb-fec-rs-01, Internet draft, in the RFC queue.
43. J. Cantillo and J. Lacan. A Design Rationale for Providing IP Services Over DVB-S2 Links. IETF draft, draft-cantillo-ipdvb-s2encaps-04.txt, expired, December 2007.

1.6.4 Brevets

44. J. Cantillo, J. Lacan, I. Buret and F. Arnal. Procédé et module de correction d'erreurs de transmission dans un flux de données, système de communication comprenant ledit module, December 2007. French patent application FR0708623.
45. J. Cantillo, J. Lacan, I. Buret and F. Arnal. Procédé et dispositif de délinéation d'un flux de données et système de communication comprenant ledit dispositif, 2008. French patent application FR0800968.

Rapport techniques et/ou en cours de soumission

46. J. Cantillo and J. Lacan. CRC-32 Performance Assessment for DVB-S Links. Technical report, Thales Alenia Space, 2005.

47. J. Cantillo, J. Lacan and A. Bouabdallah. Taking the Most out of Protocol Headers Redundancy. Technical report, May 2006. Unpublished.
48. Amine Bouabdallah, Jérôme Lacan and Michel Diaz. Adaptive Dependency-Aware Unequal Erasure Protection Code. 15th "Joint Conference on Communications and Coding" (JCCC) 12-18 March 2006
49. A. Mahmino, J. Lacan, and C. Fraboul. End-to-End packet delay bounds with network coding. soumis à une revue en août 2008.
50. J. Lacan and E. Lochin. On-the-Fly Coding to Enable Full Reliability Without Retransmission. soumis à une conférence en août 2009, disponible sur http ://arxiv.org/, 2008.
51. A. Soro, J. Lacan, E. Chaput, and C. Baudoin. Header Compression Protocols Performance Modeling. soumis à une conférence en août 2008, disponible sur http ://oatao.univ-toulouse.fr/, 2008.
52. H. Petander, E. Lochin, and J. Lacan. Safetynet version 2, a packet error recovery architecture for vertical handoffs. soumis à une conférence en septembre 2008.
53. J. Cantillo, J. Lacan and M.-L. Boucheret and I. Buret. Enhancing Delineation and Error Control with Header Redundancy : HERACLES. soumis à une revue en novembre 2008.

1.7 Contrats, Etudes et Projets

Contrat CNES " Simulation de transmissions à travers un canal gaussien utilisant des codes correcteurs d'erreurs Reed-Solomon démultipliés corrigés avec un décodeur à décision pondérée. ",
dates : 1995.
Sujet : développement d'un code Reed-Solomon à décision douce.

Projet ESA TRANSAT (Transport protocol and Resource mANagement for mobile SAtellite neTworks)
dates : septembre 2001 - Mars 2003
Sujet de notre contribution : participation à la conception et implémentation de codes à effacements dans une couche liaison pour des transmissions par satellite

Participation au réseau d'excellence NEWCOM NoE - responsable du WPR 5.3 : " Robustification tools adapted to the case of mixed wired/wireless links, incorporating Quality of Service (QoS) and more generally to channels with feedback",
dates : 2004-2007
Sujet de notre contribution : mécanismes de codage conjoint source-canal

Etude CNES - "Codage cross-layer-SDMB",
dates : décembre 2005-juin 2006
Réponse TéSA-Supélec : Bouabdallah, A., Kieffer, M. Lacan, J., Sabeva, C. and Duhamel, P.
Sujet : Evaluation des mécanismes de codage sur les différentes couches pour des transmissions satellite-mobile

Etude CNES, " Techniques de compression et satellite ",
dates : septembre 2006-juin 2007
Réponse TéSA-TAS-B2i : A. Soro, J. Lacan, E. Chaput, C. Baudoin, F. Arnal
Sujet de notre contribution : Modélisation des performances des protocoles de compression d'entête

Etude CNES " Approfondissement Cross-Layer "
dates : novembre 2006-mars 2007
Réponse TéSA-Supélec : Bouabdallah, A., Kieffer, M. Lacan, J., Sabeva, C. and Duhamel, P.,
Sujet : Evaluation des mécanismes de codage sur les différentes couches pour DVB-SH

Projet ANR - " Capri-FEC "
dates : Février 2007-Février 2010
participation ISAE : A. Soro, J. Lacan
Sujet : conception, application et analyse de performances des codes à effacement

Etude CNES : " Lutte contre les affaiblissements en canal mobile bande S "
dates : décembre 2007-octobre 2008
Participants TéSA : W. Chauvet, J. Lacan
Sujet : Modélisation et simulation rapide de la couche physique DVB-SH

Etude Rockwell-Collins : "support Etude Modem : décodage souple de codes Reed-Solomon"
dates : juin 2008-juillet 2008
Participant TéSA : M. L. Boucheret, J. Lacan
Sujet de notre contribution : développement et analyse d'un décodeur Reed-Solomon GMD.

1.8 Animation de la recherche

- Participation à l'organisation de la conférence AAECC 12, Toulouse, 2003
- Organisation du workshop NEWCOM " Source Coding and Reliable Delivery of Multimedia Contents", Toulouse, 2006.
- Responsable du work package WP 5.3 " Source Coding and Reliable Delivery of Multimedia Contents", du réseau d'excellence NEWCOM
- Présentation à des séminaires de recherche
  - séminaire cryptographie - DGA-université de Rennes, 2004
  - séminaire Codes, Crypto et Algorithmique INRIA Rocquencourt, 2002
- Responsable de plusieurs contrats avec des industriels (voir section ci-dessous)
- Relecteur pour les revues : IEEE Transactions on Information Theory, IEEE Transactions on Communications, IEEE Journal of Selected Areas in Communications, IEEE Communication letters, Computer network, Computer communications ainsi que pour diverses conférences.

Chapitre 2

Introduction

Les travaux présentés dans ce mémoire ont été réalisés depuis septembre 2001, date de mon recrutement à l'ENSICA, désormais ISAE. AAAA

2.1 Généralités sur la théorie de l'information et le codage

Le cadre général de ce travail concerne la protection de données face aux erreurs et aux pertes lors des transmissions.

Comme tous les travaux dans ce domaine, le point de départ est la théorie de l'Information introduite par C.E. Shannon [2]. Cette théorie a généré un nombre extrêmement élevé de travaux depuis plus de 50 ans. La plus grande partie s'est focalisée sur le canal à erreurs où les données sont transmises sous la forme de symboles qui peuvent être modifiés lors de la transmission. La problématique de recherche des codes correcteurs d'erreurs optimaux s'est au départ révélée difficile. Cependant, elle a été partiellement résolue par les dernières générations de codes, comme les turbo-codes [3] ou les LDPC (Low-Density Parity-Check codes) [4][5], qui ont quasiment atteint la borne de Shannon. De nombreux problèmes connexes restent ouverts, mais nous pouvons considérer que ce résultat est une avancée considérable.

Un autre type de canal souvent étudié est le canal à effacements où certains symboles peuvent être "effacés". Dans ce contexte, on connaît la position des symboles corrompus mais on ne peut récupérer leur valeur. Sur ce canal, la borne de Shannon a été atteinte (beaucoup plus facilement que sur le canal gaussien) par la famille des codes MDS (Maximum Distance Separable) dont les représentants les plus connus sont les codes Reed-Solomon [6]. Toutefois, certaines applications de ces codes ont des contraintes auxquels ces derniers ne sont pas adaptés. Ceci a conduit à l'introduction de nouveaux codes, un peu moins efficaces en termes de capacité de correction, mais plus faciles à utiliser dans la pratique.

2.2 Couches réseaux

Les deux canaux abordés ci-dessus sont des abstractions de canaux que l'on rencontre dans les systèmes de communication réels. En effet, la fiabilité n'est qu'un des nombreux problèmes devant être résolus par deux entités distantes communicantes. Par exemple, si l'on considère le cas représenté sur la Figure 2.1, pour que Alice transmettre des données à Bob, celle-ci doit :

– connaître l'adresse de Bob

- trouver un chemin de sa machine vers celle de Bob
- coder les données afin de les transmettre de manière fiable sur chacun des liens de communications comme cable coaxial sur l'Ethernet, fibre optique sur le cœur du réseau internet, lien satellite, lien sans fil (Wifi ou Wimax)
- gérer le débit de la transmission
- éventuellement gérer le problème de la sécurité des données

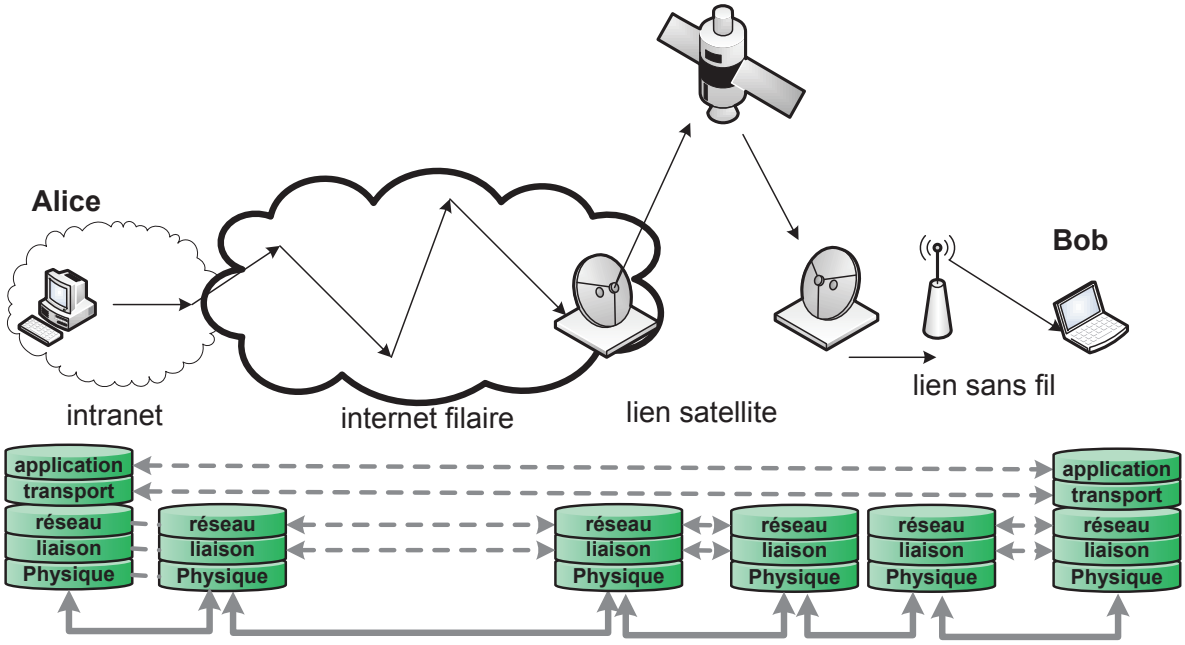


Fig. 2.1 – Exemple de transmission réelles et application du principe des couches

Pour aborder ces problèmes dans leur ensemble, le concept d'architecture en couche a été introduit. Ce principe, formalisé par le modèle OSI [7], consiste à segmenter autant que possible la gestion des différents services en différentes entités appelées *couches*. Chaque couche est représentée par une entité coté émetteur et coté récepteur. Ces deux entités communiquent de manière virtuelle via un *protocole* pour fournir le service dont ils ont la charge. Pour cela, ils considèrent qu'un certain nombre de services sont fournis, c'est-à-dire qu'un certain nombre de problèmes sont résolus. Les couches sont virtuellement empilées de telle sorte à ce que les couches inférieures fournissent un service aux couches supérieures. Le nombre de couches proposées dans le modèle de référence OSI [7] était initialement de 7 ; il est maintenant réduit à 4. En pratique, on considère plutôt que ce nombre est de 5, comme sur la Figure 2.1. Dans la suite de ce mémoire, nous prendrons ce modèle à 5 couches comme référence.

Un des points importants de ce système est que chaque couche fait *abstraction* de la manière dont les couches inférieures assurent leur service. Ceci permet de modifier un couche sans changer les couches supérieures.

Si l'on reprend l'exemple de la Figure 2.1, la couche application fait abstraction de la manière dont est géré le routage des données dans le réseau ou comment est gérée la fiabilité. Elle se focalise uniquement sur le service applicatif qu'elle doit fournir.

Ce système basé sur les couches a fait ses preuves. Il est même de plus en plus nécessaire il permet par exemple l'intégration de nouveaux protocoles, de nouvelles technologies et autorise aussi la convergence de technologies.

2.3 Gestion de la fiabilité dans une architecture en couches protocolaires

2.3.1 Gestion de la fiabilité par les différentes couches protocolaires

Dans une architecture en couche, la fiabilité peut être gérée à plusieurs niveaux en fonction des contraintes du ou des canaux ou de l'application.

Si l'on considère l'ensemble des couches, la **couche physique** est le niveau où se pose principalement la question de la fiabilité. Le canal vu par cette couche dépend de la technologie utilisée. C'est à ce niveau que sont mis en œuvre la plupart du temps des codes correcteurs d'erreurs combinés avec des modulations adéquates.

Le principal rôle de la **couche liaison** concernant la fiabilité est la vérification de l'intégrité des données. Cette vérification est réalisée via des mécanismes de type CRC (Cyclic Redundancy Checks) ou éventuellement checksums (somme de contrôle). Les unités de données détectées comme erronées sont généralement supprimées. C'est la règle préconisée par le RFC 3819 [8]. Suivant les protocoles, la couche liaison peut se charger de retransmettre les unités de données non reçues ou supprimées pour améliorer le niveau de fiabilité en termes de taux d'erreur paquet vu par les couches supérieures. L'intérêt de cette approche est qu'il est plus intéressant de retransmettre les paquets au niveau liaison sur un seul lien de communication plutôt que de laisser cette opération à un protocole de transport, qui va solliciter des retransmissions de paquets de bout en bout.

La **couche réseau**, en l'occurrence IP [9], ne gère pas la fiabilité. Elle se contente de vérifier l'intégrité de son entête avec un checksum de 16 bits. Ceci n'est plus réalisé dans IPv6 [10] où le contrôle de l'intégrité est laissé aux couches supérieures.

La **couche transport** est la première couche à gérer la transmission de bout en bout. En fonction des besoins de l'application, elle peut fournir différents services liés à la fiabilité.

Le protocole UDP [11] ne gère pas la fiabilité. Il implémente simplement un mécanisme de détection basé sur un checksum qui s'applique sur la totalité du paquet. Il faut noter que ce checksum peut être désactivé ou sa zone de protection peut être réduite [12].

La couche transport peut aussi fournir un service de fiabilité totale à la couche application. C'est, par exemple, le cas de TCP [13] qui retransmet les paquets perdus. La gestion de la fiabilité de TCP est particulière car ce protocole fait l'hypothèse que les pertes de paquets sont dues à des congestions dans le réseau. Lorsqu'il observe une perte, TCP retransmet le paquet et réduit son débit de transmission.

Dans certains types de transmission, comme les transmissions multipoints où les transmissions temps-réels, il peut être nécessaire d'utiliser des protocoles spécifiques. Dans ces cas-là, l'utilisation de codes à effacement est souvent une meilleure solution que les retransmissions classiques [14] [15]. Les groupes RMT (Reliable Multicast Transport), AVT (Audio Video Transport) et plus récemment Fecframe, de l'IETF ont proposé un certain nombre de solutions sur ces sujets.

Classiquement, la fiabilité n'est pas gérée par **la couche application** qui considère que ce service est fourni par les couches inférieures. Toutefois, certaines applications telles que les applications multimédia, ont des contraintes et des propriétés (temps-réel, tolérance aux pertes) telles qu'elle peuvent envisager une gestion propre de la fiabilité.

2.3.2 Codes vs retransmissions

Chaque couche ayant des contraintes et des objectifs différents, les mécanismes de fiabilité choisis différeront. Globalement, ces mécanismes peuvent être classés en trois catégories [16] :

- les mécanismes réactifs qui réagissent à la dégradation d'une donnée en retransmettant celle-ci. Cette classe contient notamment tous les mécanismes de type ARQ (Automatic Repeat reQuest).
- les mécanismes proactifs, qui anticipent les dégradations des données en ajoutant de la redondance aux données transmises. Cette classe se compose de tous les codes correcteurs d'erreurs et d'effacements. Ces codes sont souvent notés FEC (Forward Error Correcting codes).
- les mécanismes hybrides FEC-ARQ, ou H-ARQ qui réagissent en ajustant dynamiquement la quantité de redondance du code.

La comparaison de ces mécanismes est difficile à faire de manière globale. Toutefois, on peut distinguer des avantages et des inconvénients pour ces différentes méthodes.

La principale force des mécanismes basés sur les retransmissions est qu'ils combinent à la fois la simplicité (implémentation, complexité de calcul) et la performance (lorsque le canal de retour est fiable, il atteint la capacité du canal $(1-p)$ où p est le taux de perte du canal). Toutefois, pour être utilisé, chaque unité de donnée doit pouvoir être identifiée et détectée comme erronée. Un canal de retour doit aussi être disponible. Enfin, les retransmissions peuvent générer un délai gênant pour certaines applications.

Au contraire, les mécanismes proactifs n'ont pas besoin de canal de retour et le décodage (si il est réussi) génère simplement un retard dû à l'attente de la fin du bloc codé. La correction peut être réalisée sans identifier (directement) les données corrompues. Enfin, lorsque le taux de redondance est choisi de manière appropriée, ces codes peuvent aussi atteindre la capacité du canal. Les principaux inconvénients sont que le mécanisme ne fonctionne pas lorsque la quantité de redondance reçue n'est pas suffisante et que le décodage peut être coûteux en temps de calcul. D'autre part, un excès de redondance peut résulter en une mauvaise utilisation de la bande passante.

Les mécanismes FEC-ARQ étant des mécanismes hybrides, le poids accordé à chacun de ces deux mécanismes définit une implémentation. En effet, une implémentation orientée ARQ autorisera plusieurs rapports de réception et de retransmissions par mot. En conséquence, ses propriétés la rapprocheront du mécanisme ARQ lui-même. Au contraire, si les retransmissions sont rarement utilisées, ce mécanisme se rapprochera du mécanisme FEC.

2.4 Domaines d'utilisation des codes correcteurs d'erreurs et des codes à effacement et contributions

Notre principal thème de recherche étant lié aux techniques de codage, nous présentons dans cette partie les applications de ces techniques sur les différentes couches protocolaires. Pour chaque couche, nous présentons aussi nos contributions.

2.4.1 Utilisation du codage sur les couches basses

Couche physique

La couche physique observe généralement un canal où les erreurs sont situées sur des "petits" symboles, typiquement des bits ou des octets. Il est impossible, en considérant un bit

seul ou un octet seul, de déterminer si ce symbole est erroné ou pas et donc de solliciter sa retransmission. La retransmission ne peut s'appliquer qu'à un ensemble de symboles qui peut ne contenir qu'une faible proportion de bits erronés. Dans ces conditions, les codes correcteurs d'erreurs sont très nettement supérieur en termes de capacité de correction. Rappelons par ailleurs que les Turbo-codes ou les codes LDPC sont quasiment à la limite de Shannon. Pour des canaux variables, l'utilisation de mécanismes de type H-ARQ est probablement la meilleure solution. Ce mécanisme a d'ailleurs été introduit dans les standards HSDPA (3G+) [17] et 802.16e (Wimax mobile) [18].

Contributions personnelles -

1. Nos contributions sur la couche physique ont été réalisées dans le cadre d'études et/ou encadrements de thèses financées par le CNES ou Thalès Alenia Space. Le premier résultat, présenté dans le chapitre 4, est l'utilisation de la redondance contenue dans les entêtes des couches hautes pour améliorer le décodage de la couche physique [19]. Ce travail a notamment fait l'objet du dépôt de 2 brevets par Thalès Alenia Space [20]-[21].
2. Un autre résultat (non présenté) est la modélisation des performances d'un turbo-code et d'un entrelaceur sur un canal variable dans le temps.
3. Nous avons également contribué au décodage à décision pondérée des codes de Reed-Solomon [22]-[23] ainsi qu'au décodage récursif des codes Reed-Muller [24]. Ces contributions ne sont pas présentées dans ce mémoire.

Couche liaison

Sur la couche liaison, la problématique est différente. Les unités de données considérées par cette couche sont des ensembles de symboles. En utilisant les informations de la couche physique ou des mécanismes de détection, chaque unité de donnée est identifiée comme correcte ou corrompue. La solution la plus communément utilisée est la suppression des unités de données corrompues et la retransmission de celles-ci par l'émetteur. Les principaux cas de couches liaisons normalisées utilisant des codes correcteurs ou à effacement sont celles où la transmission est uni-directionnelle, comme DVB-H [25] ou DVB-SH [26].

Contributions personnelles -

1. Avec Tanguy Pérennou, nous avons évalué l'intérêt de l'utilisation des codes correcteurs d'erreurs sur la couche liaison de 802.11b dans le cas de transmissions multipoints [27] (non présenté).
2. Dans un cadre moins spécifique, nous avons proposé un mécanisme de fiabilité défini pour réduire le nombre de retransmissions lorsque plusieurs transmissions unipoints partagent le même canal à diffusion [28]. Ce travail est présenté dans le chapitre 5.
3. Nous avons réalisé plusieurs travaux et études sur le cas de la couche liaison de DVB-SH et de sa complémentarité avec la couche physique. Une première partie des résultats issus de l'étude [29], a été publiée dans [30]. Les résultats obtenus dans l'étude suivante [31] n'ont pas encore été soumis. Ces travaux sont présentés dans le chapitre 4.
4. Sur le même thème des relations couche physique-couche liaison, nous avons évalué les performances des codes algébriques Reed-Solomon et BCH sur la couche physique en termes de capacité de détection afin d'estimer l'intérêt de mettre un CRC sur la couche liaison [32]. Ce travail, qui s'est déroulé dans le cadre de la thèse de Juan Cantillo, a fait partie d'une contribution plus large à la conception du protocole d'encapsulation GSE [33][34]. Cette contribution a fait l'objet de plusieurs drafts IETF [35] et a eu une conséquence directe sur le protocole d'encapsulation GSE (non présenté).

2.4.2 Utilisation du codage sur les couches hautes

Utilisation de codes à effacement sur les couches hautes

Historiquement, les codes correcteurs d'erreurs étaient implémentés au niveau physique. Grâce à l'augmentation de la puissance des processeurs, l'implémentation d'algorithmes de codes à effacement a pu être réalisée au niveau logiciel à partir du milieu des années 90. Depuis cette date, plusieurs familles de codes plus ou moins spécialisées vers telle ou telle applications ont vu le jour. Malgré ça, la conception de codes approchant les performances optimales tout en supportant des algorithmes de décodage très rapides est une problématique restant ouverte.

Contributions personnelles -

- Notre première contribution sur ce sujet a été réalisée dans le but d'améliorer les vitesses de calcul des codes MDS (Maximum Distance Separable). Pour cela, avec Jérôme Fimes, nous avons proposé une implémentation logicielle d'un code utilisant les optimisations des processeurs récents définis pour les entiers [36].
- Nous avons ensuite défini un algorithme de codage/décodage de codes MDS basé sur des matrices de Vandermonde optimal en termes de complexité algorithmique [37].
- Ces travaux ont été étendus pour être intégrés dans le draft IETF sur les codes de Reed-Solomon qui devrait être publié en tant que RFC dans les mois suivant l'écriture de mémoire [38].

Ces travaux sont présentés dans le chapitre 3.

Transmissions multipoints

Dans le cadre de transmissions multipoints, les mécanismes basés sur les retransmissions ne sont plus optimaux pour deux raisons principales. La première est qu'un paquet perdu par au moins un des récepteurs doit être retransmis par l'émetteur. Ceci réduit l'intérêt du multipoint par rapport au point-à-point car dans le cas d'un grand nombre de récepteurs, chaque paquet risque d'être retransmis une ou plusieurs fois. La deuxième raison est que si tous les récepteurs acquittent simultanément un paquet reçu (ou un paquet perdu), des congestions peuvent apparaître au niveau de l'émetteur, ce qui perturbe fortement la transmission générale.

Par rapport à ce problème, les codes à effacement sont très intéressants car les paquets de redondance peuvent compenser des pertes distinctes chez différents récepteurs. Le nombre de paquets émis en plus des paquets sources initiaux est donc largement diminué. Il faut noter que cette méthode permet de délivrer des données de manière fiable sans canal de retour. Le groupe RMT (Reliable Multicast Transport) a notamment standardisé le protocole ALC/FLUTE [39] dans ce but.

Contributions personnelles -

- Le premier travail a consisté à étudier l'utilisation de codes à effacements pour des transmissions multicast par satellite vers un grand nombre de récepteurs terrestres. Cette analyse a été complétée par la proposition d'un architecture hybride satellite-terrestre permettant d'atteindre une fiabilité totale à un moindre coût [40][41]. Ce travail, réalisé avec F. de Belleville et ses encadrants, n'est pas présenté dans ce mémoire.
- Dans un travail intégrant plusieurs couches protocolaires lors de transmissions multipoints, nous avons analysé, avec Fabrice Arnal, la possibilité de laisser remonter des paquets de donnés erronés jusque vers les couches hautes (transport et applications). La conclusion principale, présentée dans le chapitre 4, a montré que cette approche peut

être très intéressante, mais seulement lorsque les entêtes des paquets sont protégés contre les erreurs [42] (voir chapitre 4).

Transmissions multimédia temps-réel

Les transmissions multimédia ont souvent des contraintes temps-réel qui s'accommodent mal de délais liés aux retransmissions. Les codes à effacement ont été naturellement proposés pour renforcer la fiabilité de ce type de transmissions (voir par ex. [15]).

Contributions personnelles -

- La première des 2 contributions de ce domaine concerne les codes à protection inégale. Ces codes, construits avec Amine Bouabdallah et présentés dans le chapitre 3, ont été définis à partir de codes de Reed-Solomon emboités [43]. Leur application aux transmissions vidéo a montré qu'ils surpassent le système PET, référence de ce domaine.
- La deuxième proposition, baptisée Tetrys, est une construction de code à la volée assurant une fiabilité totale tout en supportant des pertes de paquets d'acquittement [44]. Une propriété intéressante de ces codes est que le délai de récupération des paquets perdus ne dépend pas du RTT. Ces codes ont été construits avec Emmanuel Lochin. Ils sont présentés dans le chapitre 3.

Réseaux pair-à-pair et codage réseau

Le domaine de recherche qui a généré le plus de travaux sur les codes à effacement est probablement le codage réseau (Network coding) où les paquets sont codés par des différentes machines, ces paquets étant ensuite transmis par des chemins différents vers un ou plusieurs récepteurs. Ce schéma général s'applique particulièrement bien dans les réseaux pair-à-pair où l'utilisation de paquets de redondance permet d'améliorer la persistance des données dans le réseau ainsi que les temps de téléchargement.

Contributions personnelles -

- Avec Laurent Lancérica et ses encadrants, nous avons proposé une contribution sur l'utilisation des codes à effacement dans les réseaux pair-à-pair pour améliorer les temps de téléchargement [45]-[36].
- L'autre contribution dans ce domaine est plus centrée sur le codage réseau. Avec Ali Mahmino, nous avons montré que le codage réseau peut réduire les bornes maximales du temps de traversée de réseau. Pour cela, nous avons développé une nouvelle stratégie de codage qui s'est aussi avérée efficace pour les temps de traversée moyens.

Ces deux applications sont présentées dans le chapitre 5.

2.4.3 Autres contributions

- Dans une "vie antérieure", nous avons eu l'occasion d'appliquer certains concepts de théorie de l'information à l'analyse des séquences d'ADN. Ces travaux sur ce sujet passionnant ont fait l'objet de deux publications dans des revues internationales [46]-[47].
- Dans le cadre de projets de recherche, nous avons étudié une autre application de la théorie de l'information dans les réseaux : la compression d'en-tête [48]. Nous avons en particulier proposé une modélisation d'un protocole de compression d'en-tête générique permettant de paramétrer et de comparer les principaux protocoles d'en-tête [49] [50].

– Une dernière application des codes à effacements que nous avons proposé est l'amélioration de la tolérance aux fautes d'une application exécutant des processus en parallèle. Dans ce modèle, les entités à protéger sont des processus. Le système proposé décrit comment générer des processus redondants, de telle sorte à ce que lorsque plusieurs processus sont exécutés en parallèle, si un des processus défaille, le système n'ait pas à attendre la ré-exécution complète de ce processus [51].

Ces contributions ne sont pas présentées dans ce mémoire.

2.5 Plan de ce mémoire

Le chapitre suivant regroupe tous les résultats que nous avons obtenus sur la construction de codes à effacement. Ces travaux concernent principalement les couches hautes. Les travaux intégrant plusieurs couches sont détaillés dans le chapitre 4. Enfin, nous consacrerons le chapitre 5 aux applications des codes à effacement.

Chapitre 3

Contributions aux Codes à Effacement

3.1 Etat de l'art des codes à effacement

Introduction

Les codes à effacement ont généré relativement peu de travaux par rapport aux codes correcteurs d'erreurs. La principale raison est que ces codes s'utilisent généralement sur les couches hautes, et que dans ce contexte, toutes les opérations sont réalisées en logiciel. Leur application réelle a été rendue possible uniquement lorsque les ordinateurs personnels ont eu suffisamment de puissance pour réaliser "rapidement" les opérations de codage et de décodage, c'est-à-dire vers le milieu des années 90. Les idées fondamentales avaient toutefois été proposées largement avant. Le principe de récupération pour des transmissions multipoints avait été proposé par Metzner [52] en 1984 alors que celui de la dissémination des données dans les réseaux avait été proposé par Rabin [53] en 1989.

De manière générale, on distingue deux grandes classes de codes à effacement : les codes MDS (Maximum Distance Separable) et les codes à matrice génératrice ou à matrice de parité creuse tels que les LDPC (Low-Density Parity-Check matrix) [54] ou les codes Raptor [55].

Pour ces deux classes de codes, on considère que l'opération de codage transforme un ensemble de k symboles d'information (ou source) en un ensemble de n symboles codés. Dans toutes les applications réseau, le codage est de type systématique, ce qui veut dire que les k symboles d'information font partie des n symboles codés. Nous pouvons donc considérer que le codage rajoute $n-k$ symboles de redondance.

Un code peut être défini par sa matrice génératrice G composée de k lignes et de n colonnes. Le codage d'un vecteur d'information de k symboles $i = (i_1, \ldots, i_k)$ appartenant au corps fini $\mathbb{F}_{q^m}$ (voir [56] comme référence sur les corps finis) est réalisé par l'opération :

$$(i_1, \ldots, i_k) \times G = (c_1, \ldots, c_n)$$

où $(c_1, \ldots, c_n)$ représente le vecteur (ou le mot) codé. Clairement, si G est systématique, on doit avoir $c_j = i_j$ pour $j = 1, \ldots, k$, ce qui entraîne que les k premières colonnes de G forment la matrice identité I_k.

Lorsqu'on applique ce codage à des paquets de données, on considère que l'information est composée d'un ensemble de k paquets où chaque paquet est composé de r symboles du corps fini. Pour générer les $n-k$ paquets de redondance, on réalise r codages, tel que le $i^{\text{ème}}$ codage ait en entrée le vecteur formé des $i^{\text{ème}}$ symboles de chaque paquet d'information. Les $n-k$ symboles de redondance générés par ce codage forment alors les $i^{\text{ème}}$ composantes des

paquets de redondance produits. Le codage (comme le décodage) de paquets se déduit donc naturellement du codage (ou du décodage) d'un seul bloc de symboles.

Pour le décodage, on considère que l'on reçoit un vecteur de k' symboles $(c_{u_1}, c_{u_2}, \ldots, c_{u_{k'}})$, où $1 \leq u_1 < u_2 < \ldots < u_{k'} \leq n$. Soit $G_{u_1,\ldots,u_{k'}}$ la matrice $k \times k'$ composée des colonnes d'indice $u_1, \ldots, u_{k'}$ de G. Par construction, on a :

$$(i_1, \ldots, i_k) \times (G_{u_1,\ldots,u_{k'}}) = (c_{u_1}, c_{u_2}, \ldots, c_{u'_k})$$

Pour que le décodeur puisse retrouver le vecteur d'information à partir du vecteur reçu, il faut et il suffit que la matrice $G_{u_1,\ldots,u_{k'}}$ soit de rang k. Ceci implique en particulier que $k' \geq k$. Si $G_{u_1,\ldots,u_{k'}}$ est de rang k, notons $G_{u'_1,\ldots,u'_k}$ la sous-matrice de $G_{u_1,\ldots,u_{k'}}$ de rang k. Le décodage se fait alors en calculant :

$$(i_1, \ldots, i_k) = (c_{u'_1}, c_{u'_2}, \ldots, c_{u'_k}) \times (G_{u'_1,\ldots,u'_k})^{-1}$$

Les deux problématiques liées à ce schéma sont les suivantes :

1. trouver une matrice génératrice telle que toute sous-matrice de k colonnes soit de rang k, ce qui implique que tout mot est décodable dès que k symboles codés sont reçus. On peut montrer qu'un code qui vérifie cette condition atteint la capacité du canal à effacement, qui est de $1 - p$, où p représente le taux de perte des paquets.
2. trouver des algorithmes de codage et de décodage qui soient rapides à réaliser en logiciel.

Les codes MDS vérifient la première propriété, par contre ils ne sont pas optimaux pour la seconde. Au contraire, les codes à matrice creuse ont des codages et des décodages très rapides, par contre, ils nécessitent de recevoir en moyenne $(1 + \epsilon)k$ symboles, où ϵ n'est pas forcément négligeable.

Rappels sur les codes MDS

Les premières constructions de codes MDS sont les codes de Reed-Solomon [6]. Ceux-ci sont définis sur le corps fini $\mathbb{F}_{q^m}$. La longueur de ces codes est $n = q^m - 1$ et leur dimension est $0 \leq k \leq n$. Ces codes sont caractérisés par leur matrice génératrice définie comme la matrice de Vandermonde suivante : $G = \left(\alpha^{i.j}\right)_{i=0,\ldots,k-1;j=0,\ldots,n-1}$ où α est un élément primitif de $\mathbb{F}_{q^m}$.

Cette construction est utilisée dans le logiciel libre réalisé par L. Rizzo [57] qui a popularisé ce type de code.

Les matrices de Vandermonde possèdent des propriété très intéressantes par rapport au décodage car elles supportent des algorithmes de multiplication matrice-vecteur et d'inversion rapides (voir partie 3.2.2). Malheureusement, cette matrice génératrice n'est pas systématique. Elle ne peut donc pas être utilisée tel quel pour le codage. Il faut donc réaliser les combinaisons linéaires permettant de faire apparaître l'identité sur les k premières colonnes. Cette opération casse la structure de Vandermonde sur les colonnes restantes.

Comme cette matrice génératrice systématique génère toujours un code MDS, toute sous-matrice de k lignes et k colonnes est inversible. Ceci implique que la partie de la matrice génératrice correspondant aux symboles de parité est telle que toute sous-matrice carrée $l \times l$, avec $l \leq k$, est inversible.

L'autre méthode de construction des codes MDS est justement basée sur la construction de matrices possédant cette propriété. Ces matrices sont les matrices de Cauchy [58]. On peut

noter que les matrices de Vandermonde possèdent cette propriété sur certains corps (par exemple le corps des réels), mais ce n'est pas le cas sur les corps finis de type $\mathbb{F}_2$.

Une matrice de Cauchy $r \times r$ est définie de la manière suivante : $G = \left(\frac{1}{\alpha_j + \beta_j}\right)_{i=1,\ldots,r-1;j=1,\ldots,r}$ où $(\alpha_1, \ldots, \alpha_r)$ and $(\beta_1, \ldots, \beta_r)$ sont des vecteurs de $(\mathbb{F}_q)^r$ tels que les α_i et les β_j, $i, j = 1, \ldots, r$ sont distincts.

Que les codes MDS soient construits à partir des matrices de Vandermonde ou à partir de celles de Cauchy, leur principal inconvénient est que les opérations de codage et de décodage peuvent être gourmandes en termes de calcul. Leur complexité sera détaillée dans la section 3.2.2. Cette contrainte limite la longueur des codes utilisables et donc réduit globalement leur performance. En effet, comme indiqué dans [59], il est plus intéressant d'utiliser un code LDPC avec une grande longueur plutôt que plusieurs codes de Reed-Solomon avec de petites longueurs.

Il est toutefois utile de replacer les codes dans le contexte d'applications réelles en notant que la complexité d'un décodeur de code MDS n'a rien à voir avec celle d'un décodeur vidéo. Par exemple, des tests réalisés par Nokia [60] indiquent que, sur une plate-forme ARM 11 pour téléphone mobile, un décodage logiciel d'un code de Reed-Solomon de longueur 255 utilise environ 10 fois moins de ressources CPU qu'un décodeur H.264/AVC vidéo.

3.2 Contributions sur les codes MDS

3.2.1 Codes MDS utilisant des calculs sur des entiers

Cette contribution a été réalisée avec Jérôme Fimes. L'idée de base est que les processeurs actuels classiques permettent des opérations sur les entiers bien plus rapides que sur les éléments de corps finis polynomiaux. Nous avons donc proposé de construire des codes à effacement où les calculs se font sur des nombres entiers (modulo un nombre premier).

La solution que nous avons proposée consiste à utiliser un code MDS caractérisé par une matrice de Cauchy définie sur un corps fini premier, c'est-à-dire tel que les opérations d'addition et de multiplication soient réalisées sur des entiers modulo un nombre entier premier. Nous avons choisi d'utiliser le corps fini $\mathbb{F}_{2^{16}+1}$ en représentant les éléments du corps par des entiers codés sur 16 bits.

Pour gérer le fait qu'un des éléments du corps, en l'occurrence 2^{16}, n'ait pas de représentation directe, et donc qu'il ne puisse pas être transmis directement dans des paquets, nous avons proposé la solution suivante. Nous avons tout d'abord fait l'hypothèse que les paquets ont un nombre d'éléments de 16 bits inférieur à 2^{16} (ce qui limite la taille de paquets à environ 130 KB). Avec cette hypothèse, il existe au moins un élément inférieur à 2^{16} qui n'apparaît pas dans le paquet. Nous ajoutons alors cet élément dans l'entête du paquet et nous remplaçons toute occurrence de l'élément 2^{16} (le seul qui pose problème) par ce nombre.

Les principales opérations à implémenter lors du codage et du décodage sont des multiplications vecteur-matrice ainsi qu'une inversion de matrice. Pour implémenter ces opérations de manière efficace, on peut tout d'abord remarquer que le produit de deux symboles modulo $2^{16}+1$ peut être stocké dans un entier 32 bits en vérifiant que les deux éléments soient différents de 2^{16}, auquel cas le résultat est 1. Pour faire le produit scalaire de deux vecteurs, si l'on impose que la longueur de ces vecteurs est inférieure à 2^{16}, le résultat peut être stocké sur 48 bits. La seule opération de "modulo", coûteuse en temps de calcul, est faite à ce moment-là. L'inversion de la matrice se fait avec un algorithme de complexité quadratique,

k	Codage (en Mb/s)			Décodage (en Mb/s)		
	code sur les entiers	Cauchy [61]	Vandermonde [57]	code sur les entiers	Cauchy [61]	Vandermonde [57]
16	305.18	99.83	91.55	457.76	299.48	249.69
32	166.46	54.45	55.49	332.92	87.65	85.83
64	107.71	26.62	25.97	209.26	67.81	46.36
128	55.07	13.54	11.48	115.34	33.82	21.39
256	28.28	6.75	4.94	71.81	19.95	11.80
512	13.93	3.41	1.93	37.56	10.27	5.02
1024	7.14	1.70	0.89	18.96	5.12	1.19
2048	3.45	0.82	0.43	9.09	2.40	0.17

Tab. 3.1 – Comparaison de performances de codage et de décodage des 3 familles de codes MDS (éléments stockés sur 16 bits) - tests réalisés en 2008 sur un processeur Centrino double coeur Intel T7200, cadencé à 2.00 GHz avec 3 Go de SDRAM

comme indiqué dans [61].

Les performances de ces codes ont été confrontées à celles des codes MDS de L. Rizzo [57] et de M. Luby [61]. La plateforme utilisée était un Pentium III à 933 Mhz, 256 Ko de cache avec 384 Mo de SDRAM utilisant Linux. Le compilateur C utilisé était gcc V2.96. Les codes ont été compilés avec des options d'optimisation maximale

Ces tests, réalisés en 2003 et publiés dans [36] montrent un rapport de vitesse de l'ordre de 3.5 et 7 pour le codage et le décodage par rapport à [57] et de l'ordre de 1.33 et 1.23 pour le codage et le décodage par rapport à [61] pour une longueur de $k = 512$.

Nous avons relancé ces tests sur une machine plus récente (processeur Centrino double coeur Intel T7200, cadencé à 2.00 GHz avec 3 Go de SDRAM) et les résultats, présentés sur le tableau 3.1, se sont montrés encore plus à notre avantage. En effet, les rapports de vitesse sont de l'ordre de 4 et 7 pour le codage et le décodage par rapport à [57] et [61] pour une longueur de $k = 512$. Cette variation du rapport de performance entre ces implémentations est une forte incitation à la poursuite des travaux sur ces codes.

3.2.2 Construction de codes MDS à partir de matrices de Vandermonde

Ce travail a aussi été réalisé avec Jérôme Fimes. Après l'amélioration de type "implémentation" proposée dans la partie précédente, nous avons proposé une construction de codes MDS basée sur des matrices de Vandermonde. Le principal avantage de cette construction est qu'elle supporte une complexité de codage et de décodage plus faible que les autres constructions de codes MDS [37].

Comme indiqué précédemment, construire un code MDS de paramètres n et k est équivalent à construire une matrice $k \times (n-k)$ telle que toute sous-matrice carrée soit inversible. Une telle construction a été proposée à partir des matrices de Vandermonde.

Une forme particulière de matrice de Vandermonde a été présentée dans la partie 3.1. De manière générale, une matrice de Vandermonde carrée $r \times r$ est caractérisée de la manière suivante :

$$V(a_1, \ldots, a_r) = \left(a_i^{j-1}\right)_{i,j=1}^{r}$$

Nous avons tout d'abord construit un code MDS systématique de paramètres k, et $n = 2k$. Pour cela, nous avons montré que la matrice :

$$R = V(a_1, \ldots, a_r)^{-1} \times V(b_1, \ldots, b_r) \tag{3.1}$$

est telle que toute ses sous-matrices sont inversibles si et seulement si tous les éléments a_i et b_j, $i, j = 1, \ldots, r$, sont distincts 2 à 2. Nous en avons déduit que la matrice $[I|R]$ est une matrice génératrice d'un code MDS systématique. Cette construction a ensuite été généralisée au cas où n est quelconque (tout en restant inférieur à q) en considérant la matrice de Vandermonde $k \times (n-k)$ $V(b_1, \ldots, b_{n-k})$.

Un des principaux intérêts de cette construction est la faible complexité de ses opérations de codage et de décodage. En effet, d'après [62], les opérations utilisées dans ce contexte sont plus rapides avec les matrices de Vandermonde qu'avec les matrices de Cauchy.

De plus, cette complexité peut être encore améliorée si l'on se place dans le cas où k est un diviseur de $q-1$. Prenons le vecteur $(a_1, \ldots, a_k)$ égal à $(1, \alpha, \ldots, \alpha^{k-1})$, où α est un élément de $\mathbb{F}_q$ d'ordre k. La matrice de Vandermonde correspondante est alors notée $V(\alpha)$. Ce choix a deux intérêts. L'inversion de $V(\alpha)$ est directe car $V(\alpha)^{-1} = \frac{1}{k} \times V(\alpha^{-1})$. Ceci permet aussi de réduire la complexité des multiplications par $V(\alpha)$ et $V(\alpha^{-1})$ à $O(k \log k)$. On peut noter, que dans ce cas, les éléments de la matrice $\Pi = V(\alpha)^{-1} \times V(b_1, \ldots, b_{n-k})$ peuvent être exprimées sous la forme $\pi_{i,j} = k^{-1} \times \frac{1-b_j^k}{1-b_j/\alpha^i}$.

La complexité totale peut être évaluée à partir des outils proposés dans [62] :
- $\epsilon(n) \leq O(n \log^2(n))$ est la complexité de l'algorithme d'évaluation.
- $\iota(n) \leq O(n \log^2(n))$ est la complexité de l'algorithme d'interpolation.
- $\phi(n) \leq O(n \log(n))$ est la complexité de la transformée de Fourier rapide (FFT).

Pour le codage, la multiplication du vecteur d'information par la matrice $V(\alpha)^{-1}$ (ou $V(a_1, \ldots, a_r)^{-1}$) a une complexité de $\phi(k)$ (ou $\iota(k)$). La complexité du second produit matrice-vecteur est égal à $\epsilon(\max(k, n-k))$. Le codage a donc une complexité totale de $\phi(k) + \epsilon(\max(k, n-k))$ (ou $\iota(k) + \epsilon(\max(k, n-k))$). Cette complexité doit être comparée à $2\iota(\max(k, n-k)) + 3\epsilon(\max(k, n-k)) + O(\max(k, n-k))$ qui est la complexité de la multiplication d'un vecteur par une matrice de Cauchy [62].

Les opérations réalisées pour le décodage sont similaires car aucune inversion de matrice n'est nécessaire grâce à l'utilisation de l'algorithme d'interpolation [62]. Cette complexité est égale à $\phi(k) + \iota(k)$ si $V(\alpha)^{-1}$ est utilisée ou $\epsilon(k) + \iota(k)$ si $V(a_1, \ldots, a_r)^{-1}$ est utilisée. Cette complexité peut être comparée à $2\iota(k) + 3\epsilon(k) + O(k)$ qui est requis avec une matrice de Cauchy.

Nous n'avons pas encore implémenté ces algorithmes car l'implémentation de la FFT sur un corps fini de la forme $\mathbb{F}_{2^r}$ ne peut être réalisée directement. Il faut en effet utiliser une autre méthode (de même complexité) du type de celle proposée dans [63]. Le développement d'un tel algorithme fait partie de nos futurs projets.

3.2.3 Contribution à l'IETF

Avec Vincent Roca (INRIA Rhône-Alpes), Sami et Jani Peltotalo (Université de Tampere, Finlande), nous sommes actuellement en train de normaliser ces codes dans le groupe RMT de l'IETF. Ce document [38], qui a fait l'objet de 6 drafts IETF a passé les étapes successives de l'admission par le groupe en tant que document du groupe, puis a été approuvé par le groupe, et enfin par l'IESG. Il est actuellement dans la file d'attente finale des RFC où sa

publication est seulement retardée par sa dépendance envers un autre draft RMT qui devrait être très prochainement étudié par l'IESG. Sa publication en tant que RFC est donc imminente.

3.3 Codes à protection inégale

Ce travail a été réalisé dans le cadre de la thèse d'Amine Bouabdallah, co-encadrée par Michel Diaz. Cette thèse, qui a débuté en septembre 2005, devrait être soutenue au printemps 2009.

3.3.1 Introduction aux codes à effacement à protection inégale

Les codes MDS sont optimaux dans le sens où, pour un code de dimension k, le décodage est possible dès que k symboles parmi les n envoyés sont reçus. En contre-partie, aucun symbole ne peut être récupéré si moins de k symboles sont reçus.

Cette propriété fait que ces codes ne sont pas forcément optimaux pour les transmissions multimedia où certaines données sont plus importantes que d'autres. Les spécificités de ce type de données font qu'il est préférable d'utiliser des protocoles ou des mécanismes spécifiquement construits dans cette optique [64] [65]. En particulier, par rapport au codage, il serait souhaitable de pouvoir décoder les données importantes dès qu'une certaine quantité de données est reçue. L'objectif de ce travail consiste à modifier les codes MDS pour construire des codes ayant ce type de propriété.

Un premier exemple de données avec plusieurs niveaux d'importance est le codage de la vidéo où certaines images sont codées en intra (indépendamment des autres images) et d'autres images sont codées en fonction d'autres images. La perte des images codées en intra engendre alors des dégradations sur les images codées en fonction de cette image intra (même si des techniques de masquage d'erreurs adoucissent ce phénomène). Un second exemple est le codage hiérarchique, où les données multimédia sont organisées en couches, une couche de base et une ou plusieurs couches d'amélioration successives. Le décodage d'une couche nécessite la présence des couches de niveau inférieur. Un exemple est le système FGS (Fine Granularity Scalability) utilisé par MPEG-4. Pour ces types de données, La gestion des pertes et, plus généralement, le transport de ce type de données, doit se faire en utilisant des protocoles spécifiques [65].

Pour ce type de données, il peut être intéressant d'appliquer des niveaux de protection différents. La principale solution utilisée dans ce domaine est la solution PET (Priority Encoding Transmission) introduite par M. Luby et ses collègues de l'Université de Berkeley en 1994. Cette solution définit des couches C_i, pour $i = 1, \ldots, r$, d'importances différentes et associe à chaque couche un niveau de protection ρ_i, où $0 \leq \rho_i \leq 1$, pour $i = 1, \ldots, r$. Ce système génère des paquets de redondance et assure que la couche C_i est décodée si $(\rho_i * 100)$ % de la totalité des paquets sont reçus.

Le principal intérêt de cette méthode est sa capacité de correction qui est supérieure à celle que l'on aurait avec des codes indépendants appliqués sur chacun des types de données. Le point négatif est le fait que les unités de données utilisées par le code ne correspondent pas à des unités de données de la couche supérieure. Par exemple, si les couches correspondent aux différents types d'images d'une vidéo (I, P, B), un paquet de données transmis contient des parties d'images I, P et B. Si seules les images I ont pu être décodées par le décodeur, les parties du paquet correspondant aux images P et B ne sont pas (directement) utilisables.

L'approche que nous avons proposée est différente de PET dans le sens où PET peut être

vu comme un système de protection basé sur des techniques de paquétisation et des codes à effacement alors que notre système est seulement un code à effacement construit de telle sorte à protéger les données de manière différentes.

3.3.2 Principe de la proposition

Notre proposition consiste à modifier une matrice génératrice de code MDS afin de créer des inégalités de protection. Ces modifications reviennent simplement à remplacer certains termes de la matrice génératrice par des 0. Comme la matrice génératrice décrit les relations entre les symboles d'information et les symboles codés, le fait de rajouter un 0 en position (i, j) indique que le $i^{\text{ème}}$ symbole d'information n'est plus protégé par le $j^{\text{ème}}$ symbole codé. Ce symbole codé protège alors moins de symboles d'information. Par conséquent, le niveau de protection qu'il assure à chacun est supérieur.

Cette approche a été généralisée pour être appliquée à un flux vidéo. Pour cela, nous avons défini la règle suivante : un paquet de redondance protégeant un paquet d'information issu d'une image doit :

- protéger tous les paquets issus de cette image
- protéger toutes les images (c'est-à-dire tous les paquets issus des images) dont l'image dépend.

Un exemple d'application de cette règle est présenté sur la Figure 3.1. On considère que chaque GOP (Group Of Pictures-groupe d'images) est composé de la suite d'images IBBPBBPBB où chaque image se décompose en le nombre de paquets indiqué sur la partie gauche de la Figure. Chaque GOP génère donc 15 paquets d'information. On suppose que le code génère 5 paquets de redondance et on décide d'affecter 2 paquets de redondance à la protection des images I, 2 paquets de redondance à la protection des images I et P, et 1 paquets de redondance à la protection des images I, P et B. En tenant compte du fait que les images P sont codées en fonction de l'image I ou P précédente et que les images B sont codées en fonction des images I ou P qui les encadre, on obtient la matrice génératrice de la Figure 3.1. Notons que les termes non nuls de la matrice sont des termes d'un matrice de Cauchy. Cette construction, que nous

Image	I	B	B	P	B	B	P	B	B
Nb de paquets	5	1	1	2	1	1	2	1	1

$$G_{20,15} = \left(\begin{array}{c|cccc} & \alpha_{1,5} & \cdots & \cdots & \alpha_{1,5} \\ & \vdots & \ddots & & \vdots \\ & \vdots & & \ddots & \vdots \\ & \alpha_{5,1} & \cdots & \cdots & \alpha_{5,5} \\ & 0 & 0 & 0 & \alpha_{6,5} \\ & 0 & 0 & 0 & \alpha_{7,5} \\ Id_{15} & 0 & 0 & \alpha_{8,3} & \alpha_{8,4} & \alpha_{8,5} \\ & 0 & 0 & \alpha_{8,4} & \alpha_{9,4} & \alpha_{9,5} \\ & 0 & 0 & 0 & 0 & \alpha_{10,5} \\ & 0 & 0 & 0 & 0 & \alpha_{11,5} \\ & 0 & 0 & 0 & \alpha_{12,4} & \alpha_{12,5} \\ & 0 & 0 & 0 & \alpha_{12,5} & \alpha_{13,5} \\ & 0 & 0 & 0 & 0 & \alpha_{14,5} \\ & 0 & 0 & 0 & 0 & \alpha_{15,5} \end{array}\right)$$

Fig. 3.1 – Construction d'un code à protection inégale pour un GOP particulier

avons baptisée DA-UEP (Dependency-Aware Unequal Erasure Protection), a été comparée à PET et à un code MDS par des simulations. Nous avons repris les GOP et les conditions sur PET présentées dans [66] où les images I, P et B sont respectivement décodées lorsque 60%, 80% et 90% des paquets sont reçus. Le calcul de la quantité de redondance donne 24.35% de

redondance. Sur ce GOP, nous avons simulé les 3 systèmes de protection (PET, UEP et MDS). Les résultats sont présentés sur la Figure 3.2. On peut observer que notre code DA-UEP est légèrement moins bon que les 2 autres lorsque la quantité de redondance est supérieure au taux de perte. Par contre, la tendance s'inverse lorsque le taux de perte devient supérieur au taux de redondance. En particulier, sur une échelle MOS (Mean Opinion Square), nous pouvons observer que notre schéma reste dans l'état "GOOD" avec un taux de perte supérieur de 12% à celui de PET. Pour l'état "FAIR", l'écart est de 10%. Ces résultats ont été publiés

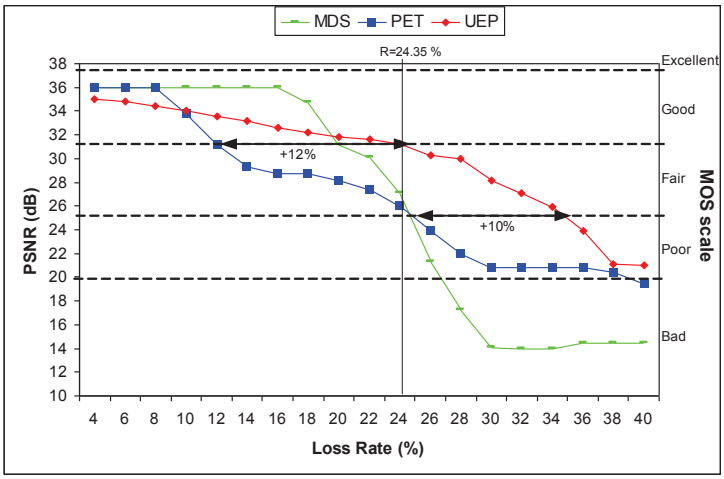


Fig. 3.2 – Comparaison des différents systèmes de codage

dans la conférences Packet Video 2006 [43]-[67]. Pour approfondir ces résultats de simulations prometteurs, nous avons proposé une modélisation des performances de cette technique.

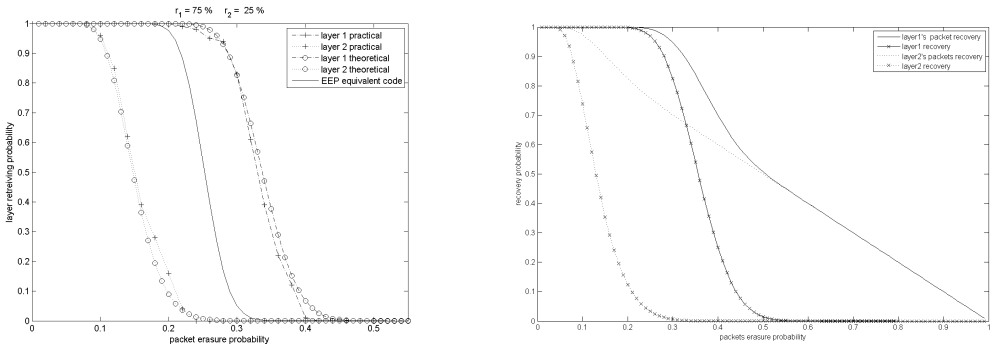
3.3.3 Modélisation des performances de DA-UEP

Pour cette modélisation, nous avons considéré le cas général de données "multimédia" organisées en couches (une de base et plusieurs d'amélioration). Nous avons supposé connaître la fonction débit/distortion qui donne le niveau de qualité observé pour une quantité de données reçue.

Pour faciliter la modélisation, nous avons choisi de représenter ces codes comme des codes "emboîtés". Le principe consiste à voir ce code comme plusieurs petits codes : le premier code protège la couche de base, le second code protège la couche de base et la première couche d'amélioration, et ainsi de suite... Nous considérons ensuite que si le premier code décode, la couche de base est récupérée et donc, toute la redondance du second code s'applique uniquement à la couche d'amélioration.

Ces codes étant des codes MDS, on peut modéliser leur performance indépendamment. En fixant les règles de dépendance entre les codes, on peut calculer une approximation de la performance globale du code. La Figure 3.3(a) présente les résultats théoriques et pratiques d'un code à protection inégale et le compare avec un code MDS ayant le même niveau de redondance. A partir de cette évaluation théorique, nous avons formalisé le problème de la distribution de la redondance entre les différentes couches en tant que problème d'optimisation. Les entrées de ce problème sont le taux de perte de paquets, la taille des différentes couches, la quantité totale de redondance à distribuer et la fonction débit/distortion.

Afin de comparer cette solution avec PET, nous avons évalué théoriquement PET dans le



(a) Comparaison des performances théoriques et pratiques d'un code à protection inégale ainsi que d'un code à protection égale (EEP) MDS. La quantité totale de redondance est de 25%. Le code à protection inégale consacre 75% de la redondance à la couche 1 et 25% à la couche 2. Les 2 couches ont la même taille.

(b) Proportion de données réellement utilisables par DA-UEP

Fig. 3.3 – Performances de DA-UEP

même contexte. Ceci a été rendu possible par le fait que PET utilise lui aussi des codes MDS dont on peut prédire les performances.

Finalement, nous avons défini une méthode permettant de trouver le code PET correspondant à un code DA-UEP donné. Nous avons pu observer que les performances sont très similaires entre les deux méthodes. Les courbes de DA-EP sont "centrées" sur celles de PET, mais sont moins abruptes. Ceci peut être expliqué par le fait que les codes MDS utilisés dans les deux méthodes ont quasiment le même taux de codage, par contre, leur longueur est plus importante dans PET.

Un autre point important est le fait que l'organisation des données en paquet fait qu'avec DA-UEP, toutes les données reçues sont exploitables alors qu'avec PET, ce n'est pas le cas directement. Ceci est illustré sur la Figure 3.3(b) qui présente la quantité de données réellement utilisable par le décodeur lorsque DA-UEP est utilisé. Ces courbes prennent en compte les paquets qui ont été reçus mais qui ne font pas partie d'un mot de code décodé.

3.3.4 Conclusion

L'ensemble de ce travail sera détaillé dans la thèse d'Amine Bouabdallah (soutenue au printemps 2009) et doit être prochainement soumis dans une revue internationale. Le résultat final est un système dont les performances semblent être supérieures à PET, notamment grâce au fait que les paquets reçus mais ne faisant pas partie d'un mot décodé peuvent être utilisés par la couche supérieure. La simplicité de sa mise en œuvre nous paraissent être aussi un point positif important pour une utilisation dans des futurs applications pratiques.

Il faut noter que le principe de ce travail a été récemment repris par une équipe allemande en remplaçant les codes MDS par des codes Raptor [68]. Cette manipulation permet de réduire la complexité du décodage au détriment de la capacité de correction. Comme les codes utilisés dans ce type d'application sont habituellement courts, le problème de la complexité ne nous semble pas crucial et de plus, l'inefficacité des codes pour ce type de codes

sera vraisemblablement de l'optimalité. Nous attendrons donc d'autres publications avant de juger cette proposition.

3.4 Tetrys

Ce dernier travail sur des constructions particulières de codes à effacement a été réalisé avec Emmanuel Lochin. Au moment de l'écriture de ce mémoire, ce travail récent a fait l'objet d'une publication (sous la forme d'un résumé) dans une conférence [44]. Une version étendue est en cours de soumission à une conférence de plus large audience.

3.4.1 Introduction

Comme indiqué dans l'introduction de ce mémoire, l'un des champs d'application possibles des codes à effacement concerne les applications temps-réel, ou de manière plus large, les applications qui supportent difficilement des délais de transmission supérieurs à un nombre réduit (1 ou 2) de RTT. Les applications de visio ou audio-conférence entrent dans cette catégorie, comme, à un degré moindre, les transmissions en streaming. Les transmissions à très long délais, comme les transmissions par satellite ou les transmissions inter-planétaires font aussi partie de cette catégorie.

Tetrys est davantage un système basé sur le codage plutôt qu'un code à part entière. En effet, il est conçu pour les communications bi-directionnelles et les indications des récepteurs sont des éléments essentiels du système. Il peut être vu comme une étape supplémentaire dans l'évolution des codes à effacement vers les protocoles de communication. En effet, historiquement, après l'utilisation de codes issues directement de la théorie des codes, comme les codes de Reed-Solomon, de nouveaux codes ont été proposés pour répondre aux besoins des applications. Les exemples les plus frappants sont les codes "rateless", dont les représentants les plus connus sont les codes Raptor [69], capables de générer un très grand nombre de paquets de redondance pour un nombre donné de symboles d'information.

Tetrys est aussi capable de générer autant de paquets que l'on veut, mais en plus des codes "rateless", il n'a plus la contrainte de générer les paquets de redondance en fonction d'un bloc de paquets d'information. En effet, le principal concept introduit dans Tetrys est qu'à tout moment, l'ensemble des paquets d'information utilisés par le codeur pour générer les symboles de redondance est l'ensemble des paquets qui n'ont pas encore été acquittés par le récepteur. Au niveau conceptuel, Tetrys peut être vu comme une autre extension des codes MDS (après DA-UEP) car il est défini sur un corps fini quelconque (pas forcément binaire) et ses algorithmes de codage et de décodage le rapprochent de ces codes.

3.4.2 Principe de Tetrys

Le principe de Tetrys est illustré sur la Figure 3.4.

Algorithme coté émetteur

Du coté émetteur, le codeur génère des paquets de redondance en respectant un certain taux de redondance (proportion de paquets de redondance par rapport au nombre total de paquets). L'émission de ces paquets de redondance peut être régulière ou aléatoire. Comme indiqué plus haut, chaque paquet de redondance est généré à partir de l'ensemble des paquets qui n'ont pas été acquittés par le récepteur. Par exemple, sur la Figure 3.4, le paquet de

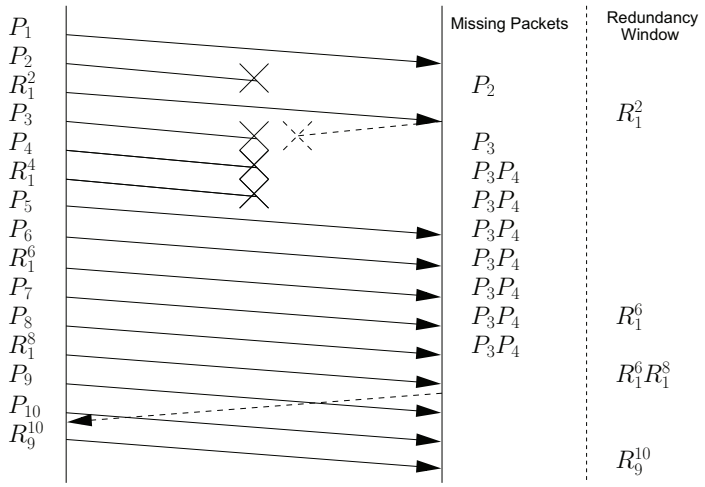


FIG. 3.4 – Principe de base de Tetrys

redondance R_1^2 est codé en fonction de P_1 à P_2, R_1^4 est codé en fonction de P_1 à P_4, et ainsi de suite. Lorsque l'émetteur reçoit un acquittement indiquant que le récepteur a reçu ou décodé tous les paquets jusqu'à un certain niveau, il encode les paquets de redondance suivants en fonction du premier paquet d'information non acquitté et de ses suivants.

Par exemple, sur la Figure 3.4, après le paquet P_{10}, le récepteur reçoit un acquittement indiquant que les paquets d'information ont été reçus ou décodés jusqu'au paquet P_8, le paquet de redondance suivant est alors encodé en fonction des paquets P_9 et P_{10}.

La génération des paquets de redondance se fait par une combinaison linéaire des paquets d'information. Cette combinaison linéaire est réalisée sur un corps fini de la forme $\mathbb{F}_{2^r}$. Les coefficients peuvent être choisis aléatoirement ou suivant une matrice définie à l'avance. L'entête des paquets doit donner une indication sur les coefficients utilisés. Lorsque les coefficients sont choisis aléatoirement, une solution consiste à utiliser un générateur pseudo-aléatoire dont la graine est un des champs de l'entête (qui doit être différent pour tous les paquets). Lorsque la matrice génératrice est définie à l'avance, l'entête doit indiquer le numéro de la colonne de la matrice. Dans les deux cas, on doit aussi indiquer quel est le premier paquet d'information utilisé dans le codage.

Algorithme coté récepteur

Avec les indications données dans les entêtes de chaque paquet de redondance reçu, le récepteur est capable de reconstituer la partie de la matrice génératrice qui a été utilisée pour construire ces paquets de redondance. Il soustrait à ces paquets de redondance les paquets d'information qui ont été reçus.

En supprimant de cette sous-matrice toutes les lignes correspondant aux symboles d'information reçus, il obtient alors une sous-matrice de la matrice génératrice dont les lignes correspondent aux paquets d'information non reçus et dont les colonnes correspondent aux paquets de redondance reçus. Dès que le nombre de paquets de redondance reçus est supérieur ou égal au nombre de paquets d'information perdus, le décodeur tente d'inverser la sous-matrice. Si cette sous-matrice est inversible, la multiplication de cette sous-matrice inversée par les paquets de redondance permet de retrouver les paquets d'information perdus.

Par exemple, sur la Figure 3.4, à la réception du paquet R_1^2, il est possible de récupérer P_2 et avec les paquets R_1^6 et R_1^8, il est possible de récupérer les paquets P_3 et P_4.

Après chaque décodage réussi ou après chaque paquet de redondance reçu inutile (i.e. il n'y a pas de paquets perdus non récupérés), le récepteur envoie un acquittement à l'émetteur pour lui indiquer le numéro du dernier paquet d'information reçu.

Un point très intéressant de ce mécanisme est que la perte d'un acquittement n'a pas de conséquence importante sur le fonctionnement du protocole. Par exemple, sur la Figure 3.4, la la perte de l'acquittement des paquets d'information jusqu'à P_2 a comme seule conséquence que l'émetteur continue à intégrer les paquets P_1 et P_2 dans les nouveaux paquets de redondance.

3.4.3 Analyse de Tetrys

Plusieurs paramètres doivent être évalués dans le cas de Tetrys. Le premier est le *délai de récupération* des paquets perdus. En effet, un des intérêts de ce mécanisme est que les paquets perdus sont récupérés sans que des retransmissions ne soient nécessaires. Les longs RTT n'ont donc pas une influence directe sur les performances du mécanisme. Un autre paramètre important est la *taille des matrices* à inverser pour récupérer les données. Cette taille a une influence sur le temps de calcul et le nombre d'opérations à réaliser par le décodeur. Enfin, le troisième paramètre est la durée observée entre la première perte non récupérée et le moment où le décodage a lieu. Ce paramètre est appelé *temps de récurrence*.

Modélisation

Pour estimer ces trois paramètres, nous avons proposé une modélisation en considérant que les pertes de paquets étaient aléatoires et uniformément distribuées. Cette modélisation représente l'évolution du décodeur comme une marche aléatoire sur des entiers. La variable aléatoire entière correspond au nombre de paquets d'information perdus moins le nombre de paquets de redondance reçus depuis le dernier décodage. Cette modélisation nous a permis de démontrer le résultat suivant :

Théorème 3.4.1 *Si $r > p$, alors tout paquet perdu est récupéré en un temps fini. Si $r = p$, tout paquet perdu est récupéré, mais le délai moyen de récupération est infini.*

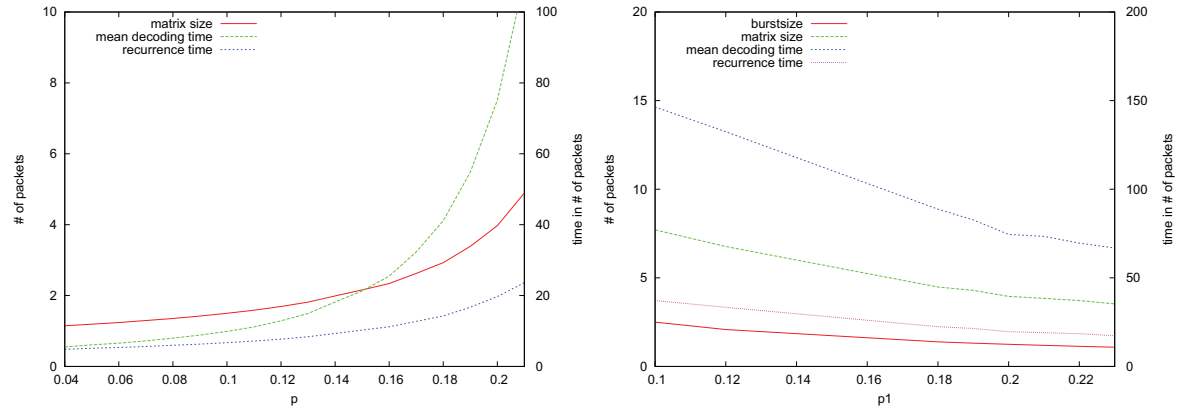
Nous avons aussi pu déterminer la loi du temps de récurrence ainsi qu'une approximation de la loi de probabilité suivie par la taille de la matrice à inverser. Enfin, nous avons obtenu une approximation de la loi du délai de décodage. Ces résultats ainsi que les démonstrations sont détaillés dans [70].

Simulations : résultats et interprétations

Pour compléter cette modélisation, nous avons développé une implémentation du mécanisme. Nous avons ainsi pu observer l'influence des paramètres suivants.

Influence du taux de perte paquet

La Figure 3.5(a) montre les performances en termes de temps de décodage moyen par paquet, de taille moyenne de matrice inversée et de temps de récurrence moyen sur un canal avec des pertes indépendantes. La première observation est que ces 3 courbes augmentent avec le taux de perte des paquets. Ceci s'explique par le fait que lorsque la probabilité d'erreur est faible comparativement au taux de redondance, le décodage est réalisé tôt et donc le temps



(a) Influence du taux de perte paquets

(b) Influence de la taille des bursts de pertes

FIG. 3.5 – Variations de la taille des matrices (en nombre de paquets), du temps de décodage et du temps de récurrence (en unité de temps). L'unité de temps représente la durée entre la transmission de 2 paquets. Le taux de redondance est fixé à 0.25.

de récurrence, le délai de récupération et la taille des matrices sont faibles. Lorsque le taux de perte s'approche du taux de redondance, ces 3 paramètres augmentent. Par exemple, un des résultats théoriques obtenus montre que le temps de récurrence moyen est égal à $1/(r-p)$ et donc est infini lorsque $r = p$.

Influence de la corrélation des pertes

Pour évaluer l'influence des pertes corrélées sur les performances de Tetrys, nous avons généré un canal de Gilbert-Eliott où les pertes sont modélisées par une chaîne de Markov à 2 états. Nous avons fait varier le taux de perte dans le bon état, noté p_1 (qui correspond à la probabilité de passer dans le mauvais état), et dans le mauvais état, noté p_2, de telle sorte à conserver un taux de perte moyen p à 0.2 ($p = p_1/(1 + p_1 - p_2)$). Le taux de redondance est fixé à 0.25. Les résultats sont présentés sur la courbe 3.5(b) où nous avons représenté les mêmes paramètres que sur la Figure 3.5(a) auxquels nous avons rajouté la taille moyenne d'un burst de pertes (égale à $1/(1-p_2)$). La principale information de cette courbe est que les rafales de pertes ont une influence négative sur la performance de Tetrys. On peut en effet observer que lorsque la taille des bursts de pertes varie de 2.5 à 1.25, les autres paramètres sont approximativement divisés par 2.

Influence de la taille du corps fini

La variation de la taille du corps fini nous a permis de constater que le meilleur compromis performance/vitesse de décodage semble être le corps $\mathbb{F}_8$ car pour des tailles de corps supérieures, les performances ne sont pas significativement meilleures. Par contre, l'utilisation des corps finis $\mathbb{F}_4$ et surtout $\mathbb{F}_2$ réduit significativement les performances de Tetrys (multiplication par 4 du délai de décodage pour $\mathbb{F}_2$).

3.4.4 Discussion

La complexité algorithmique de ce système est un point qui doit être évalué précisément. La complexité au niveau du codeur ne semble pas spécialement problématique. Le principal facteur est le temps de récurrence qui a une influence sur le nombre de paquets impliqués dans la génération d'un paquet de redondance. Pour le décodage, deux types d'opérations doivent être réalisés. La première consiste à retirer les paquets d'information reçus des paquets de

redondance reçus. Le nombre de paquets à retirer dépend aussi du temps de récurrence. La seconde opération est l'inversion de la matrice et la multiplication de cette matrice par les paquets de redondance. Pour une matrice de taille $m \times m$, l'inversion est de l'ordre de $O(m^3)$ opérations et la multiplication par les paquets de redondance nécessite $O(sz.m^2)$ opérations, où sz est le nombre de symboles du corps fini dans chaque paquet.

Le fait que Tetrys ne fonctionne pas par bloc de paquets d'information met en relation certains paramètres qui ne sont pas liés dans un schéma de codage classique. En effet, nous avons pu voir que la différence entre le taux de redondance et le taux de perte moyen est un facteur essentiel du mécanisme car elle a une influence directe sur le délai de décodage, sur la taille des matrices à inverser ainsi que le temps de récurrence. Ces 3 paramètres sont directement liés à la quantité de mémoire et au nombre de calculs effectués par le récepteur. Une conséquence importante de ceci est que la différence entre le taux de redondance et le taux de perte peut être vu comme une "mollette" permettant à un terminal avec de fortes contraintes (en mémoire ou en énergie) de supporter le mécanisme. Notons que la quantité de redondance peut être ajustée et modifiée tout au long d'une transmission.

Enfin, un dernier point concerne la façon d'agencer les paquets de redondance par l'émetteur. En effet, nous avons pu observer que lorsque les paquets de redondance sont générés à intervalle régulier, les performances du mécanisme sont largement améliorées. Par exemple, pour un taux de perte de 0.1 avec des erreurs indépendantes, lorsque les paquets de redondance sont émis aléatoirement avec un taux de 0.1667, le délai de décodage moyen est de 33.30. Sur le même canal, si l'émetteur génère un paquet de redondance après exactement 5 paquets d'information, le taux de redondance est identique mais le délai de décodage moyen est réduit à 15.88.

Dans un futur proche, Tetrys devra se positionner par rapport aux travaux de J.K. Sundararajan, D. Shah et M. Médard que nous avons découvert lors de la rédaction de ce mémoire. Ces auteurs proposent dans [71] un système comparable au nôtre dont le but est de réduire les délais de décodage. Par rapport à notre travail, une différence importante est la gestion des informations de retour. La proposition de la méthode "drop when seen" leur permet en particulier de réduire la taille des paquets qu'il est nécessaire de stocker. Notons que ce mécanisme peut être parfaitement intégré à Tetrys. Les application visées ne sont pas exactement les mêmes. Une de leurs propositions [72] consiste à placer ce mécanisme entre les couches IP et TCP pour améliorer le fonctionnement de TCP sur des liens sans fil. Cette différence d'objectifs explique probablement les différences d'analyse, notamment au niveau des temps de récupération.

3.5 Conclusions et perspectives sur les codes à effacement

Les différentes contributions présentées dans ce chapitre traitent de codes non binaires. Cette approche ne va pas dans le sens de la plupart des dernières contributions sur les codes à effacement qui proposent des codes binaires dans le but d'accélérer les vitesses de décodage.

Notre avis est que si certaines applications, comme les transmissions multipoints de gros fichiers, nécessitent des codes avec de grandes longueurs ($k > 1000$), d'autres applications ont des contraintes qui impliquent des codes plus courts. Dans ce dernier cas, comme on peut le voir sur la tableau 3.1, les vitesses de codage et décodage des codes non binaires ne sont plus critiques et les paramètres à optimiser en priorité sont d'une autre nature (dépendances entre les données pour DA-UEP et délai de décodage pour Tetrys). Dans ces conditions, les codes non binaires offrent un niveau de performance supérieur. Pour cette raison, nous pensons que

ce type de code est le plus adapté pour certaines applications et doit continuer à faire l'objet d'études et de propositions.

Chapitre 4

Codage multi-couches

Comme indiqué dans l'introduction de ce mémoire, le fait du pouvoir appliquer des mécanismes de fiabilité (et notamment des codes correcteurs d'erreurs ou d'effacements) sur plusieurs couches a ouvert de nouvelles problématiques de recherche.

Par rapport à cette problématique, nous avons eu des contributions sur les thèmes suivants

- l'optimisation globale de plusieurs codes fonctionnant sur des couches différentes
- la suppression de mécanismes effectuant des opérations redondantes (et donc inutiles) sur certaines couches
- le détournement de certaines redondances pour améliorer la fiabilité

4.1 Utilisation de codes correcteurs d'erreurs sur les couches hautes

Ce travail a été réalisé dans le cadre de la thèse de Fabrice Arnal encadrée par Laurent Dairaine et Gérard Maral. Il a été publié dans [42].

Ce travail a permis d'évaluer l'utilisation de codes correcteurs d'erreurs sur les couches hautes. Le but de cette approche est d'améliorer l'utilisation du lien lors de transmissions multipoints en évitant le gâchis classique consistant à détruire tout paquet de données contenant au moins un bit en erreur. L'idée principale était de laisser les paquets erronés remonter jusqu'aux couches hautes pour :

- soit les corriger avec un code correcteur d'erreurs placé sur la couche transport
- soit laisser l'application gérer elle-même le flux erroné avec des mécanismes de masquage ou de correction d'erreurs

Même si classiquement, la couche liaison est censée délivrer des données sans erreur, l'approche proposée est à situer dans le contexte de plusieurs travaux qui ont considéré ce type d'approche comme envisageable.

La travail le plus connu est probablement UDP-Lite [12] qui permet d'ajuster la zone du paquet UDP couverte par le checksum. Ce protocole permet aux paquets contenant des erreurs hors des entêtes de remonter jusqu'à l'application. Dans le même contexte, le protocole de compression d'entête ROHC-v1 [73] considère que les données issues de la couche liaison peuvent contenir des erreurs.

En pratique, les deux seuls cas que nous ayons rencontrés où la couche liaison peut laisser passer des paquet erronés est la couche RLC de UMTS lorsque ce protocole est utilisé en mode transparent [74] ainsi que le protocole SLIP [75] de transmission des paquets IP sur des lignes séries.

Il faut noter que certaines applications tolèrent des flux d'entrée erronés. Par exemple, le codec de voix de type AMR utilisé dans UMTS supporte un taux d'erreur bit de l'ordre de 10^{-3} sur les bits d'importance moyenne et faible [76]. En ce qui concerne la vidéo, les simulations réalisées par nos collègues de Supélec dans le cadre de [30] montrent que le taux d'erreur bit maximal supporté par un décodeur H.264 classique est de l'ordre de 10^{-5}. Dans le même temps, le taux de perte paquet maximum supporté est de l'ordre de 5.10^{-2}. Le fait de laisser remonter des erreurs bit jusqu'à la couche application n'est donc pas irréaliste.

L'approche que nous avons proposée est principalement constituée de 2 mécanismes :

- MPHP (Multi-Protocol Header Protection) est un code correcteur d'erreurs s'appliquant au niveau liaison pour protéger les entêtes des couches liaison, réseau et éventuellement transport.
- un code correcteur d'erreurs combiné avec des acquittements (hybride ARQ) s'appliquant sur des groupes de paquets et fournissant une fiabilité totale au niveau transport.

Ces deux mécanismes sont évalués dans une pile de protocoles définie pour une transmission par satellite. Les protocoles de base de cette pile sont DVB-S/MPE/IP/UDP. Pour laisser les erreurs remonter les couches, nous avons utilisé le protocole UDP-lite [12] ainsi qu'une version modifiée de MPE réduisant la portée du CRC aux champs choisis. Nous avons appelé ce protocole MPE-lite.

Deux types de paramètres ont été étudiés. Le premier concerne le nombre de symboles erronés observés par la couche application lorsqu'aucun mécanisme de fiabilité totale n'a été utilisé sur les couches intermédiaires. Nous avons considéré que des erreurs sur les entêtes n'étaient pas acceptables. Par conséquent, nous avons défini les approches suivantes qui correspondent aux différentes courbes présentées sur la Figure 4.1 :

- classique : lorsqu'une erreur est détectée sur la totalité du paquet, tout le paquet est effacé. La taille du paquet a donc une influence sur le taux d'erreur symbole observé sur les couches hautes
- MPE-lite : lorsqu'une erreur est détectée sur une des entêtes, le paquet entier est supprimé.
- MPE-lite et MPHP : lorsque le nombre d'erreurs sur l'entête dépasse la capacité de correction du code, le paquet est effacé. Le code utilisé est un Reed-Solomon $[255, 85, 170]$ défini sur $\mathbb{F}_{256}$.

Pour évaluer le taux d'erreur symbole des différentes stratégies, nous avons considéré que les paquets sont des groupes d'octets et qu'un octet effacé est faux avec une probabilité 255/256. Les résultats montrent que le fait que les paquets corrompus soient effacés sur les couches basses augmente de manière très significative le taux d'erreur observé sur les couches hautes (au sens de la définition donnée plus haut). Le meilleur résultat est obtenu par l'utilisation du couple MPE-Lite/MPHP. Ce résultat doit être pondéré par le fait que le taux d'erreur n'est pas forcément le meilleur paramètre pour les applications des couches hautes. Toutefois, si l'on considère un code Reed-Solomon erreur/effacement sur la couche transport (comme dans la partie suivante), on sait qu'il faut 2 fois plus de redondance pour corriger une erreur qu'un effacement. Même dans ces conditions, la solution MPE-Lite/MPHP reste la meilleure.

Le deuxième point étudié est le nombre de paquets devant être transmis pour assurer une fiabilité totale. Pour évaluer ce paramètre, nous avons proposé un système basé sur l'utilisation d'un code de Reed-Solomon fonctionnant en mode erreur ou erreur/effacement sur la couche Transport. La fiabilité totale est assurée par une stratégie de retransmission où les retours des récepteurs sont utilisés pour retransmettre de nouveaux paquets de redondance. Le schéma de codage est le même que celui utilisé pour les codes à effacement de paquet (décrit dans la partie 3.1). La seule différence est que les décodages réalisés utilisent des algorithmes de

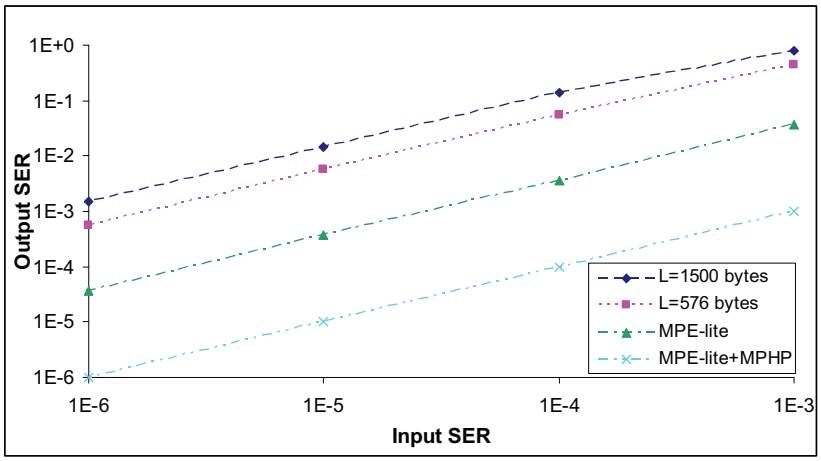


Fig. 4.1 – Taux d'erreur symbole pour les différentes stratégies

correction d'erreurs et d'effacement de type Berlekamp-Massey [16] au lieu des inversions de matrice.

Nous avons considéré les stratégies suivantes :

- classique : chaque paquet erroné est effacé sur la couche liaison. Le code de Reed-Solomon fonctionne en mode effacement. Il retransmet exactement le nombre de paquets manquants.
- classique et estimation du canal : même stratégie, mis à part que le nombre de paquets retransmis tient compte de la probabilité de perte sur le canal. Cette stratégie est légèrement moins efficace que la stratégie précédente, mais elle nécessite moins de phases de retransmissions et de trafic sur la voie retour.
- MPHP : le code fonctionne en mode erreur et effacement. Le paquets de redondance sont transmis par blocs de G paquets.

Ces stratégies sont représentées sur la Figure 4.2. L'axe des abcisses représente la puissance du signal reçu sur la couche physique du plus mauvais des récepteurs du groupe multipoint. Le simulateur, qui a été développé en utilisant la bibliothèque IT++, contenait tous les mécanismes de fiabilité de la couche physique (codec convolutionel et Reed-Solomon) jusqu'à la couche Transport (Hybride-ARQ). Le résultat de ces simulations montre très clairement le gain obtenu avec ce système et le potentiel des solutions multi-couches (cross-layer) autorisant les erreurs à remonter vers les couches hautes.

Il faut toutefois voir que ce système ne fonctionne que si les entêtes peuvent être protégées contre les erreurs. La solution proposée ici (un code de Reed-Solomon protégeant les entêtes de la couche transport à la couche liaison) est très efficace, mais semble difficile à implémenter dans un contexte réel.

Une solution un peu moins efficace mais plus réaliste pourrait être l'utilisation de la redondance déjà présente dans les entêtes pour la protection contre les erreurs comme dans [77].

4.2 Codage multi-couches pour DVB-SH

Ce travail a été réalisé dans le cadre de 2 études CNES [29]-[31] réalisées par les laboratoires TéSA et LSS/Supélec. Les participants étaient Amine Bouabdallah- pour la première étude- et

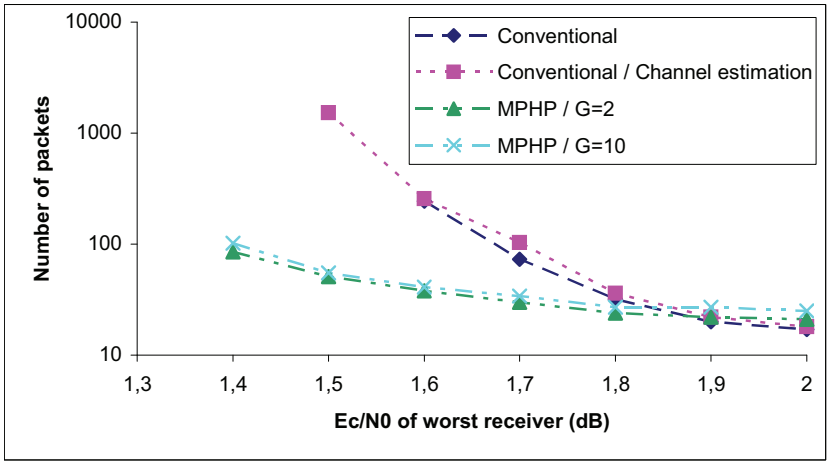


Fig. 4.2 – Nombre de paquets nécessaires pour assurer une fiabilité totale

moi-même pour TéSA et Michel Kieffer et Pierre Duhamel pour LSS/Supélec. Nos collègues de Supélec ont principalement pris en charge la couche application et tous les développements liés à la vidéo et nous nous sommes chargés des couches intermédiaires et basses.

Ce travail reprend une partie des problématiques développées dans la section précédente. Il traite de la répartition du codage sur les différentes couches protocolaires dans le cadre particulier de DVB-SH, qui est le standard DVB de diffusion de télévision mobile par satellite vers des récepteurs mobiles. Les récepteurs peuvent avoir des contraintes variables en termes d'équipement : ces récepteurs peuvent être des bus avec des antennes relativement importantes sur le toit, comme des téléphones mobiles avec des antennes et des ressources très limitées. Même si la diffusion par satellite peut être complétée dans les zones urbaines par des répéteurs terrestres qui retransmettant le signal vers les mobiles dans leur zone de couverture, notre travail a uniquement concerné le lien satellite-mobile.

Du point de vue "cross-layer", ce cas d'étude est particulièrement intéressant car le récepteur contient la pile complète des protocoles utilisés lors de la transmission sans fil. Ceci implique que toutes les techniques cross-layer "verticales" (communications entre les couches au niveau du récepteur) peuvent être envisagées de manière réaliste.

Nous n'entrerons pas dans une présentation détaillée de ce standard relativement complexe. La référence de ce standard est [26] et celle des guidelines est [78]. Toutefois, il est utile de rappeler les grandes lignes du standard.

La pile de protocole utilisée est H.264/UDP/IP/MPE-IFEC/DVB-SH. Les 2 couches spécifiques de ce standard sont :

- MPE-IFEC : couche liaison modifiée à partir de MPE de telle sorte à intégrer un code à effacement avec un entrelacement temporel potentiellement important (un mot de code peut être distribué sur une durée allant jusqu'à 30 secondes). Le code choisi initialement était un code de Reed-Solomon, mais les faibles performances mises en évidence par des simulations récentes [78] pourraient remettre en cause ce choix.
- la couche physique DVB-SH utilise le turbo-code 3GPP2 et peut elle aussi implémenter un entrelaceur temporel. Cette couche physique utilise 2 formes d'ondes (OFDM ou TDM) et 4 modulations différentes. La transmission utilise la bande S.

Dans ce contexte, le canal utilisé est celui décrit dans [79] basé sur une chaîne de Markov à 3 états où chaque état est caractérisé par une loi de Loo particulière [80]. La principale difficulté

de ce canal est son extrême versatilité. Le seul moyen de gérer ces importantes variations est d'entrelacer temporellement les unités de données codées.

La première des solutions envisagées consiste à laisser à la couche physique toute la responsabilité de la fiabilité. Dans ce cas, la totalité de la redondance est allouée à la couche physique et les mots du turbo-code sont distribués au maximum dans le temps (entrelaceur de durée maximale). Cette solution est la plus efficace en termes de capacité de correction, par contre, elle est très coûteuse en termes de quantité de données à stocker. En effet, l'entrelaceur temporel sur plusieurs secondes implique le stockage de plusieurs dizaines de mégabits de données, ces données étant représentées par des LLR (rapport de logarithmes de vraisemblance), qui nécessitent plusieurs bits pour chaque bit reçu.

L'autre solution consiste à utiliser un code moins puissant sur la couche physique et à garder de la redondance pour le code à effacement de la couche liaison. Sur la couche physique, l'entrelaceur est de petite longueur alors que sur la couche liaison, l'entrelaceur peut avoir une grande longueur. Le problème du stockage de données est moins important car l'information à stocker n'est plus sous forme de LLR, mais sous une simple forme binaire. De plus, les paquets contenant des données erronées ont été supprimés.

D'autres solutions de fiabilité peuvent être envisagées. Par exemple, comme dans le chapitre précédent, on peut envisager de laisser remonter les erreurs, éventuellement accompagnées des coefficients soft (les LLR), jusqu'aux couches hautes en leur laissant de soin d'utiliser au mieux ces informations. Une autre solution consiste à utiliser un code à protection inégale, tel que celui présenté dans le chapitre 3.3, sur des données vidéo codées avec une structure en couches hiérarchiques.

Ces différentes solutions ont été étudiées et évaluées en développant un simulateur basé sur IT++ implémentant l'ensemble des couches. Les résultats que nous avons obtenus peuvent être classés en 2 catégories. La première catégorie, parties 4.2.1 et 4.2.2, compare des mécanismes de fiabilité de manière générale ; ceux-ci ont été obtenus avec une version du simulateur pas relativement éloigné du standard DVB-SH (mais reprenant les concepts). Ces résultats ont été publiés dans [30].

La deuxième catégorie de résultats, parties 4.2.3 à 4.2.6, est beaucoup plus proche de ce standard avec des couches physique et liaison respectant les règles générales du standard.

4.2.1 Influence de la vitesse sur la qualité de la vidéo

Pour cette première estimation, le code correcteur implémenté sur la couche physique est le turbo-code 3GPP2 avec un taux de codage de $1/3$ et une dimension de $188 * 8$ bits.

Le canal, fourni par le CNES sur la base de la modélisation proposée dans [79], donne le niveau de réception par unité de longueur. Tous les résultats montrés dans ce chapitre concernent le canal ITS (Intermediate Tree Shadow), qui est le canal intermédiaire parmi les 3 fournis par le CNES.

La Figure 4.3 montre l'influence de la vitesse sur les performances observées en termes de PSNR (Peak Signal Noise Ratio). Nous avons représenté en abcisse le produit vitesse*(longueur de l'entrelaceur) qui définit la dispersion spatiale d'un mot de code. Comme on peut le constater sur la Figure, la vitesse a quasiment la même influence qu'un entrelaceur placé sur la couche physique.

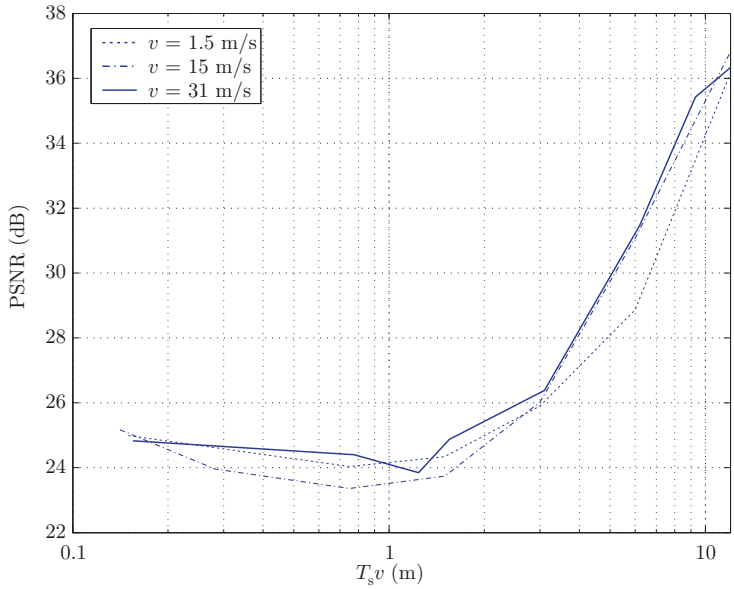


Fig. 4.3 – PSNR observé en fonction du produit vitesse*(longueur de l'entrelaceur).

4.2.2 Analyse de la dispersion du code à effacement

Une des solutions envisagées pour améliorer la fiabilité est l'utilisation d'un code à effacement. Ce code a été placé au niveau transport. Il est caractérisé par son taux de codage et sa dispersion dans le temps, c'est-à dire par sa longueur. Nous avons défini cette longueur par la variable T_{block}, qui correspond à la durée sur laquelle est distribué un mot de code. Le tableau 4.1 montre les performances en termes de qualité vidéo et en taux de perte des codes à effacement. Nous avons progressivement "déplacé" la redondance de la vidéo vers le code à effacement. Le résultat, sans appel, montre clairement que l'utilisation d'un code

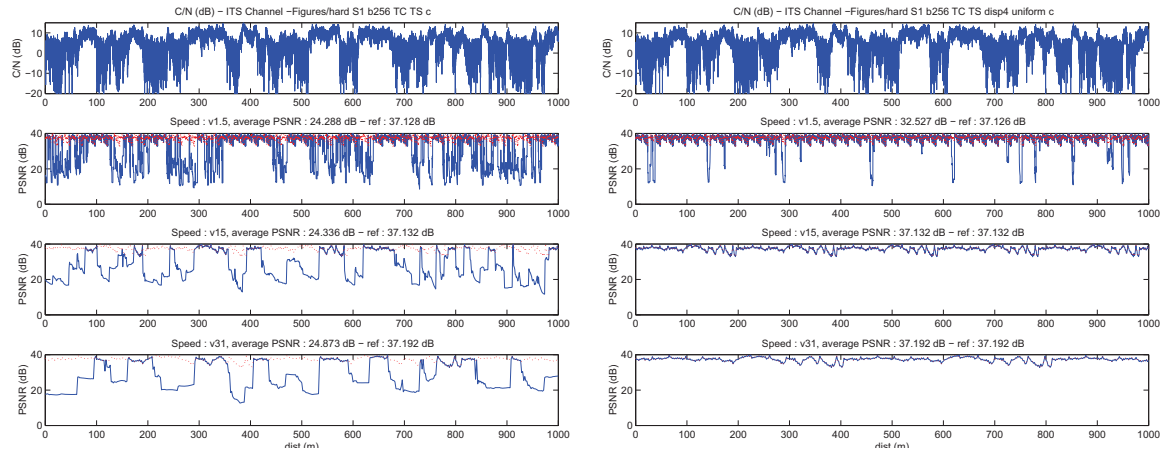
	T_{block} (s)				
R	0	1	2	3	Noise-free
250 kbps	24.5				36.8
200 kbps	24.6	26.2 RS(261, 157)	26.5 RS(521, 313)	27.5 RS(782, 469)	35.8
150 kbps	23.9	28.2 RS(196, 118)	30.1 RS(391, 235)	30.8 RS(586, 252)	34.4
100 kbps	23.1	28.4 RS(131, 79)	30.1 RS(261, 157)	30.1 RS(391, 233)	32.3

Tab. 4.1 – PSNR moyen (PLR) pour $v = 15$ m/s, pour différentes combinaisons de débit de la vidéo et pour plusieurs longueurs de $T_{\text{block}} = 0$.

à effacement est plus efficace. Ce résultat est d'autant meilleur que le code est étalé dans le temps.

4.2.3 Analyse de l'entrelaceur sur la couche physique

Un autre mécanisme de fiabilité testé est l'utilisation d'un entrelaceur sur la couche physique. Chaque mot de code de la couche physique est ainsi étalé sur une durée, ce qui se traduit par un étalement dans l'espace qui est fonction de la vitesse. Nous avons implémenté des entrelaceurs réguliers et non réguliers, ces derniers permettant de réduire le temps de zapping (voir partie 4.2.4). Notons que ces entrelaceurs ne sont pas exactement ceux spécifiés dans DVB-SH. La Figure 4.4 montre la différence de résultats obtenus lorsque le code de la couche physique est dispersé sur une durée de 4 secondes. Le résultat est extrêmement



(a) variation du PSNR sans entrelaceur

(b) variation du PSNR avec un entrelaceur de 4 secondes

Fig. 4.4 – Variation du PSNR en fonction de la vitesse et de l'entrelaceur.

significatif. En effet, on peut observer que la simple dispersion des données dans le temps permet de résoudre le problème de la variabilité du canal.

Néanmoins, cette technique a deux inconvénients. Le premier est qu'elle nécessite le stockage d'une quantité importante de données sur la couche physique (plusieurs Mégabits dans certains cas), ce qui peut poser des problèmes technologiques importants (augmentation du coût des terminaux, dépense d'énergie supplémentaire). L'autre problème est le délai engendré lorsqu'un mobile débute la réception du canal (temps de zapping). L'étude de ce délai est l'objet de la partie suivante.

4.2.4 Analyse du temps de zapping

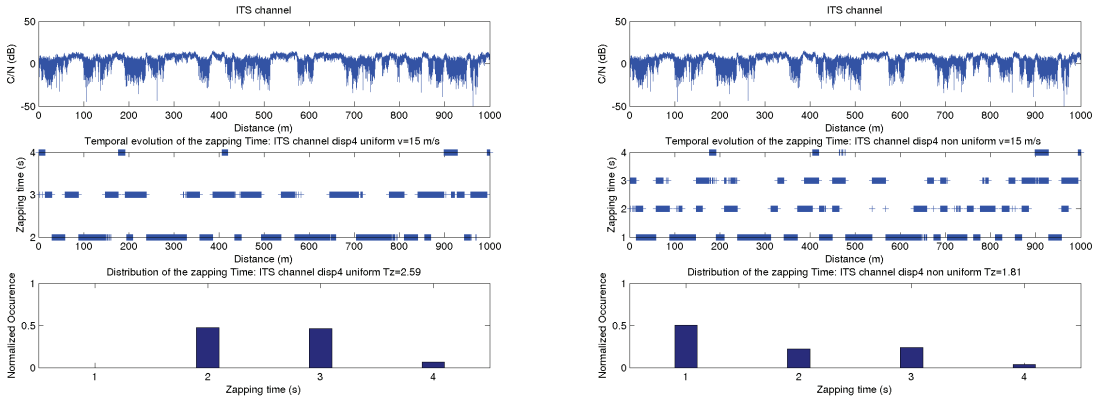
Le temps de zapping est défini comme étant la durée que doit attendre un utilisateur quand il se connecte à un programme, c'est-à-dire quand il commence à recevoir un flux.

Le choix de l'entrelaceur est un problème difficile car il doit concilier la dispersion des données dans le temps tout en maintenant un temps de zapping acceptable. Dans le cas de la couche physique de DVB-SH, le meilleur compromis est un entrelaceur irrégulier où la plus grosse partie d'un mot de code est transmise à la fin de la durée de transmission (voir [78]). Dans de bonnes conditions de réception, un utilisateur qui se connecte peut espérer récupérer des blocs qui ont commencé à être transmis plusieurs secondes auparavant.

Dans le cas du canal ITS et pour une vitesse de 15 m/s, la Figure 4.5 donne les résultats des temps de zapping pour des disperseurs sur 4 secondes. Le code est un turbo-code avec

un taux de codage de 1/3. Le premier entrelaceur uniforme place un quart du mot courant par burst d'une seconde. Le récepteur a donc besoin d'au moins 2 secondes de réception pour décoder le mot. Le second entrelaceur place un sixième du mot courant sur les 3 premières secondes et la moitié du mot sur la quatrième seconde. Le récepteur peut donc décoder avec seulement la dernière seconde de transmission.

On peut observer que l'entrelaceur non uniforme obtient un temps de zapping qui est meilleur que l'entrelaceur uniforme. Toutefois, il faut souligner que dans le cas considéré, les deux entrelaceurs permettent de décoder toutes les données, ce qui n'est pas toujours le cas ; l'entrelaceur non uniforme étant (légèrement) moins efficace en termes de capacité de correction.



(a) Distribution du temps de zapping pour un entrelaceur uniforme de 4s

(b) Distribution du temps de zapping pour un entrelaceur non uniforme de 4s

Fig. 4.5 – Distribution des temps de zapping pour des entrelaceurs sur 4 s

4.2.5 Distribution de la redondance entre la couche physique et la couche liaison

Le problème de la distribution de la redondance entre les couches physique et liaison est probablement le problème le plus important qu'ont à résoudre les concepteurs du système DVB-SH. En effet, la première solution consiste à mettre toute la redondance sur la couche physique et à utiliser l'entrelaceur à ce niveau. L'autre solution consiste à utiliser un entrelaceur court (de l'ordre de 200ms) sur la couche physique et à garder une partie de le redondance pour le code à effacement de la couche liaison qui serait alors dispersé sur plusieurs secondes [78].

Nous avons implémenté les deux solutions dans notre simulateur. Pour le canal ITS, les résultats de la première solution ont déjà été présentés sur la Figure 4.4(b). On peut observer que le niveau de réception est parfait pour les vitesses 15 et 31 m/s. Les résultats de la seconde méthode sont présentés sur la Figure 4.6. On peut constater que les résultats sont nettement plus mauvais que pour la première méthode. Ceci peut s'expliquer par le fait que les données non corrigées par la couche physique se produisent par bursts de longueurs importantes. Malgré l'étalement dans le temps, le code à effacement ne peut corriger ce type d'erreurs. Par exemple, à la vitesse de 31 m/s, le taux de perte de paquets à l'entrée de la couche liaison est de 13.31%. Malgré la quantité de redondance importante du code à effacement (33% de redondance), le code à effacement ne peut faire baisser ce taux de perte

qu'à 9.10%. La faiblesse de cette solution a été récemment confirmée dans le cadre du groupe DVB-SH [78].

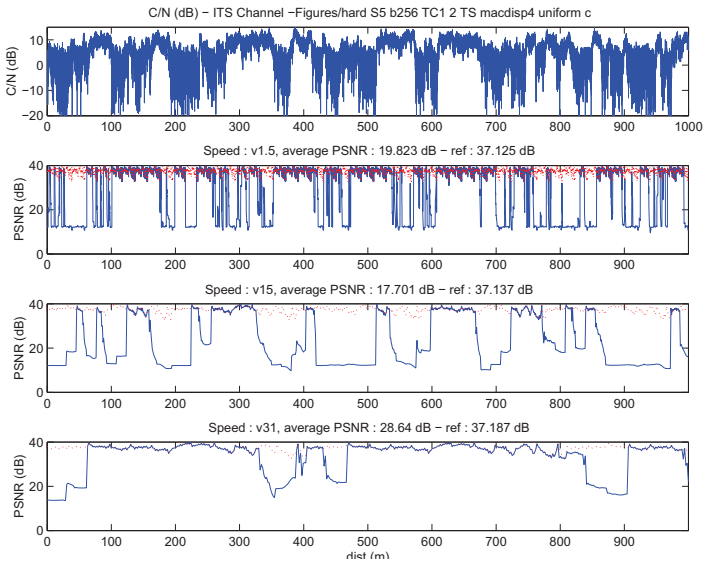


Fig. 4.6 – PSNR avec un Turbo code de taux de codage 1/2 sur la couche physique et un code à effacement de taux de codage 2/3 sur la couche liaison

4.2.6 Autres solutions évaluées

Plusieurs autres solutions ont été évaluées dans le cadre de ces études. Elles sont brièvement présentées ci-dessous.

Décodage à erreur sur la couche liaison

Le principe de cette modification consiste à décoder le code de la couche liaison en mode erreur et non plus effacement. Les unités de données erronées provenant de la couche physique ne sont plus effacées, mais transmises tel quel à la couche liaison. Notons que les données sont entrelacées entre les couches physique et liaison. Le code utilisé étant un code de Reed-Solomon, il peut décoder les erreurs. Les résultats de simulation ont montré que cette technique ne permet pas de récupérer la qualité vidéo qui a été perdue avec l'utilisation de 2 codes, toutefois, le gain n'est pas négligeable car il est de l'ordre d'environ 2 dB en moyenne au niveau de la vidéo dans le cas de la Figure 4.6. Un gain supplémentaire peut être attendu si le code de Reed-Solomon fonctionne avec un décodage soft.

Utilisation d'un code à effacement à protection inégale

Nous avons aussi testé la solution consistant à utiliser un code à effacement à protection inégale. Même si le standard H.264 prévoit un mode "data partitioning", ce mode n'était pas fonctionnel sur le codec de référence au moment des tests. Les "couches" de niveaux d'importances différentes ont été construites en fonction du type d'image I, P, B comme cela a été présenté dans la partie 3.3. Les codes que nous avons utilisés sont DA-UEP, présenté

dans la partie 3.3, et PET [81]. Nous avons "pris" la redondance nécessaire sur le code de la couche physique ainsi que sur la couche vidéo.

Les résultats ont été globalement mitigés. Comme dans la partie 3.3, notre code à protection inégale s'est montré plus efficace que PET. Les meilleurs résultats ont été obtenus en prenant la redondance sur la vidéo. Toutefois, les gains ne sont pas extraordinaires. Encore une fois, ceci s'explique par le fait que les erreurs (et les pertes de paquets) arrivent groupées. Les codes à protection inégales ne sont pas idéaux pour combattre ce type de pertes.

Augmentation de la proportion de données vidéo codées en intra et décodage soft de la vidéo

Nos collègues de Supélec ont testé 2 solutions basées sur des modifications liées au codec vidéo. La première a consisté à augmenter la proportion de données vidéo codées en intra. Les résultats n'ont pas été concluant car, à débit constant, cette augmentation se traduit par une baisse nominale de la qualité vidéo. Les simulations ont montré que cette perte de qualité n'est pas compensée par le gain en fiabilité.

L'autre solution testée a consisté à décoder la vidéo en utilisant les informations soft produites en sortie du turbo-code de la couche physique. Sur un canal gaussien, des gains de 1.2 dB (sur le PSNR) avaient pu être observées. Des résultats de ce niveau n'ont pu être atteints dans le contexte de cette étude, principalement à cause du fait que les erreurs arrivent de manière groupées sur le décodeur vidéo. Une piste de recherche intéressante pourrait consister à placer un entrelaceur avant cette couche.

4.2.7 Conclusions sur DVB-SH

L'étude du contexte DVB-SH est particulièrement intéressante car elle autorise un grand nombre de solutions de codage cross-layer. Il est évident que les résultats obtenus sont directement liés au canal considéré. Parmi l'ensemble des solutions que nous avons testées, la solution la plus efficace consiste à mettre toute la redondance et l'entrelaceur sur la couche physique. S'il s'avère que cette solution n'est pas viable au niveau technologique ou économique, l'utilisation d'un code au niveau de la couche liaison peut être envisagé. Il serait tout de même conseillé de la faire fonctionner en mode erreur plutôt qu'en mode effacement.

4.3 Héracles

4.3.1 Contexte

La dernière contribution au codage cross-layer s'est déroulée dans le cadre de la thèse de Juan Cantillo au laboratoire TéSA financée par Thalès Alenia Space. Coté Thalès Alenia Space, cette thèse a été suivie par Isabelle Buret et Fabrice Arnal, et coté TéSA, par Marie-Laure Boucheret et moi-même.

Le premier résultat de cette thèse, dont le sujet concernait le codage cross-layer pour les communications par satellite, a été la contribution au protocole GSE (Generic Stream Encapsulation) [33] pour lequel nous avons notamment montré que l'utilisation d'un CRC sur la couche liaison n'est pas nécessairement utile si un code en bloc algébrique (Reed-Solomon ou BCH) est utilisé sur la couche inférieure [35][34]. Cette contribution n'est pas détaillée dans ce mémoire.

L'autre contribution de cette thèse est le mécanisme *HERACLES* (Header Redundancy Assisted Cross-Layered Error Suppression). Ce mécanisme a donné lieu au dépôt de 2 brevets [20] et [21]. Il est actuellement soumis pour publication dans un journal.

Ce mécanisme est basé sur le constat que les entêtes des protocoles des différentes couches contiennent nécessairement de la redondance. Cette redondance est due au fait que chaque paquet doit pouvoir être transmis indépendamment des autres paquets. Il existe aussi des dépendances directes entre ces entêtes comme par exemple des numéros de séquence incrémentés à chaque paquet. De plus, des mécanismes de détection tels que les CRC constituent aussi une forme de redondance dans les entêtes.

Cette redondance peut être partiellement réduite par des méthodes de compression d'entêtes, toutefois l'utilisation de ces méthodes n'est pas généralisée dans les systèmes réels.

L'idée principale de HERACLES est d'utiliser la redondance présente dans les entêtes pour améliorer le niveau de protection du flux complet, y compris des données hors des entêtes.

4.3.2 Principe

L'hypothèse de base du mécanisme HERACLES est que la couche protocolaire où il est implémenté a une certaine connaissance des protocoles utilisés sur les couches supérieures ainsi que de certains champs de ces protocoles. De cette connaissance, est déduite une séquence de bits constante, pas nécessairement consécutive, mais dont l'espacement entre les bits connus est constant. Pour simplifier la présentation du mécanisme, nous considérerons que cette chaîne est contiguë et que sa longueur est de F octets. Typiquement, cette chaîne, notée SP (static pattern), peut être constituée des champs fixes des entêtes des couches supérieures (RTP, UDP, TCP, IP et/ou MAC) du (ou des) flux transporté(s).

Le mécanisme HERACLES est utilisé uniquement par le récepteur. Il n'implique donc aucune modification des standards. La première étape consiste à faire glisser une fenêtre sur le flux reçu, pouvant contenir des erreurs, pour détecter la présence de la chaîne SP comme indiqué sur la Figure 4.7. La métrique utilisée pour détecter les SP varie en fonction de la

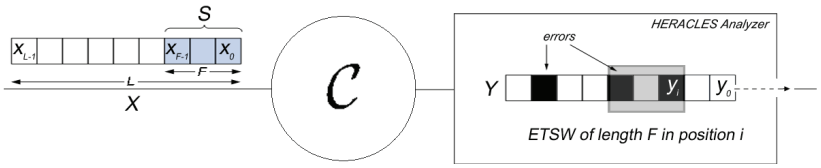


FIG. 4.7 – Principe de base de HERACLES : détection des séquences constantes SP par application d'une fenêtre glissante sur le flux erroné reçu par le récepteur

séquence de symboles sur laquelle est appliqué le mécanisme. A partir de cette métrique, on définit la *Probabilité de Détection Correcte* P_{cd} et la *Probabilité de Fausse Alarme* P_{fa} et on en déduit un seuil à partir duquel on considère que la séquence observée correspond à un SP.

Plusieurs applications de ce mécanisme de détection peuvent être envisagées :

– la détection de l'entête et la connaissance *a priori* de la position de cette entête par rapport au début du paquet permettent la délimitation du paquet. Cette application

peut être utile sur les couches basses lorsque les champs de longueur des paquets sont erroné.
- la correction de l'entête. On remplace les positions de la séquence détectée qui ne correspondent pas avec celles du SP par celles du SP. Le taux d'erreur de l'entête est réduit et la capacité du paquet à remonter dans les couches est renforcée.
- la correction de l'entête sur la couche physique avant le décodage du code correcteur d'erreurs (voir Figure 4.10). HERACLES joue le rôle d'un premier code correcteur d'erreurs concaténé en série avec le code correcteur d'erreur classique. Si la longueur du SP est significative par rapport au taux d'erreur du canal, le premier "nettoyage" du flux par HERACLES a un effet "avalanche" sur le code correcteur d'erreurs.

4.3.3 Utilisation de HERACLES en mode "hard" sur le canal binaire symétrique

Le premier cas d'utilisation de HERACLES est sur un canal binaire. Le flux est une suite de symboles binaires sur lequel la détection des SP se fait bit à bit. Par analogie aux codes correcteurs d'erreurs, on parlera de décodage "hard". Lors du processus de détection, on considère que les SP sont alignés sur les octets, et que donc, il est suffisant de faire glisser la fenêtre de recherche octet par octet.

Comme critère de détection, on considère qu'une séquence correspond à un SP si la distance de Hamming entre ces 2 séquences est inférieure à un seuil que nous noterons η.

Considérons un canal binaire symétrique (BSC) sans mémoire caractérisé une probabilité d'erreur ε. P_{cd} et P_{fa} peuvent alors être déterminées avec des formules combinatoires classiques.

Dans le souci de fournir un critère pratique pour déterminer la valeur du η optimal, nous avons déterminé le critère PSR (Probability of Static Pattern Recovery) qui regroupe les probabilités P_{cd} et P_{fa}. Sous l'hypothèse que la longueur moyenne des paquets est égale à L (c'est-à-dire que la probabilité d'apparition du SP est environ de $1/L$), nous avons défini :

$$PSR = P_{cd} \cdot \left(1 - P_{fa}\right)^{L-1} \tag{4.1}$$

qui correspond au cas où, dans un paquet de taille L, le SP est détecté et les $L-1$ autres positions ne sont pas détectées comme SP. A partir des formules de P_{cd} et P_{fa}, nous obtenons

$$PSR(\eta) = \sum_{k=0}^{\eta} \left(\begin{array}{c} 8F \\ k \end{array} \right) \varepsilon^k \left(1-\varepsilon\right)^{8F-k} \left[1 - \frac{1}{2^{8F}} \sum_{j=0}^{\eta} \left(\begin{array}{c} 8F \\ j \end{array} \right) \right]^{L-1} \tag{4.2}$$

La Figure 4.8 illustre les valeurs de $PSR(\eta)$ pour $\varepsilon = 10^{-1}$, $F = 16$ et $L = 100$. La valeur optimale de η, notée η_{opt} est définie de la manière suivante

$$\eta_{opt} = \arg\max_{\eta > 0} PSR(\eta) \tag{4.3}$$

Sur la Figure 4.8, on peut observer que $\eta_{opt} = 32$ et que pour cette valeur, $PSR(32) = 1 - 10^{-6}$. Notons que cette valeur (très faible) est atteinte pour une valeur de F courante (c'est le nombre d'octets invariants dans une entête TCP/IPv4) et pour un taux d'erreur très fort (10^{-1}). Ceci illustre le potentiel du mécanisme dans un contexte réel.

En faisant varier les valeurs de F et en déterminant pour chaque valeur le seuil $\eta_{opt}(F)$, on peut montrer que, pour un taux d'erreur de 10^{-1}, le mécanisme atteint un niveau de détection

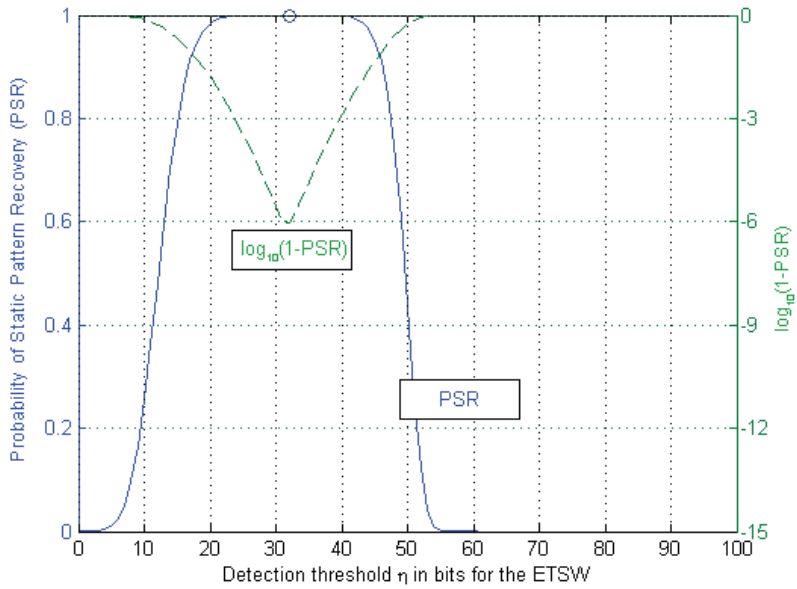


FIG. 4.8 – Valeurs de PSR en fonction de η

(PSR) très satisfaisant à partir de 8 octets tout en maintenant une probabilité de fausse alarme très faible.

En termes de correction d'erreur, le mécanisme (tel que présenté dans cette partie) corrige uniquement les erreurs sur les entêtes. Le taux d'erreur bit BER_H après correction par HERACLES peut être approximé par

$$BER_H \approx \varepsilon \left(1 - PSR\frac{F}{L}\right) \tag{4.4}$$

L'intérêt de cette opération dépend de la proportion d'entête dans le flux. Cette proportion peut être relativement importante pour certaines applications comme la voix sur IP. Un autre intérêt est l'utilisation de cette technique pour consolider la fiabilité des entêtes dans une pile protocolaire où l'on autorise la remontée des paquets erronés vers les couches hautes, comme dans la partie 4.1.

Enfin, utilisé sur la couche physique avant le décodeur d'un code correcteur d'erreur, HERACLES peut réduire le taux d'erreur bit en entrée du décodeur et ainsi contribuer à l'amélioration globale des performances en termes de correction d'erreur du système. Le niveau de cette amélioration n'a pas été étudié dans le cas "hard", mais est largement détaillé dans le cas "soft" (partie 4.3.5) où les gains peuvent être significatifs.

4.3.4 Utilisation de HERACLES en mode "soft" sur le canal gaussien

L'utilisation de HERACLES sur un flux binaire peut être envisagée sur n'importe quelle couche nécessitant une correction d'erreurs ou une délimitation des paquets et dont le flux est potentiellement erroné.

Dans cette partie, nous proposons d'utiliser HERACLES sur la couche physique en l'appliquant sur le flux en sortie du démodulateur. Ce flux est composé de symboles LLR (Log-Likelihood Ratio) indiquant le niveau de confiance associé à la réception de chaque bit.

Par rapport au décodage "hard", la principale différence est la métrique utilisée pour évaluer la distance entre le signal reçu et le SP. En fait, cette métrique est un simple produit scalaire entre la séquence des $8.F$ LLR considérée et celle du SP (où les 0 sont représentés par des 1 et les 1 par des -1).

Dans le cas d'un canal gaussien, on peut montrer que la distance entre le signal reçu et le SP suit deux lois gaussiennes suivant que l'on se trouve dans le cas où un SP a été émis ou pas. A partir de ces lois, on peut déduire les valeurs des probabilités P_{cd} et P_{fa} et en déduire la valeur de PSR. Le seuil optimal η_{opt} peut alors être déterminé.

La Figure 4.9 indique les valeurs des distances observées, notées z, sur un canal gaussien correspondant à un taux d'erreur bit de 10^{-1} pour une modulation QPSK. La valeur de F est égale à 16, L vaut 100. La variable z_0 correspond distance observée lorsqu'un SP a été émis et la variable z_i correspond au cas où une autre séquence a été émise. On peut observer que les approximations théoriques sont satisfaisantes et que le seuil de détection η_{opt} est lui aussi satisfaisant. Comme dans le mode "hard", HERACLES peut être utilisé pour corriger

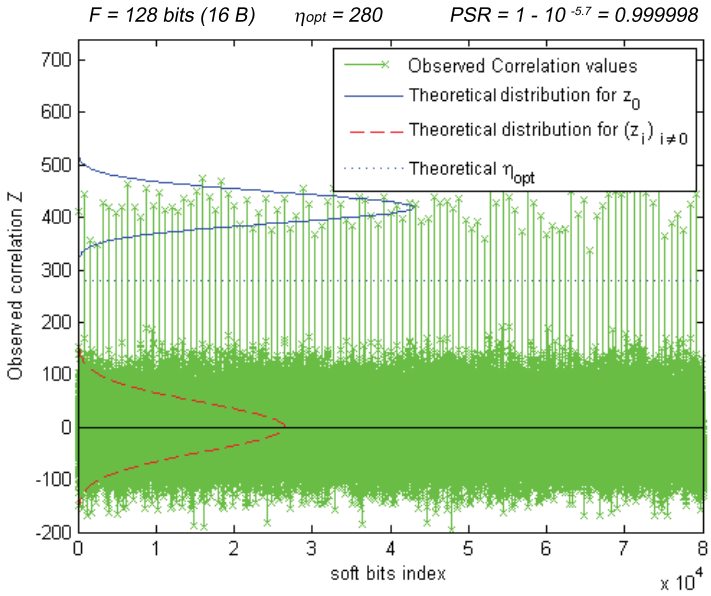


Fig. 4.9 – Observation des valeurs PSR sur un canal gaussien

les erreurs sur les entêtes. Lorsqu'il est appliqué seulement sur les entêtes, la diminution du taux d'erreur est de l'ordre de celui donné dans l'équation 4.4. Le cas le plus intéressant est lorsqu'il est combiné avec un décodeur de code correcteur. Ce cas est analysé dans la partie suivante.

4.3.5 Combinaison de HERACLES avec un code correcteur d'erreurs

Comme indiqué précédemment, l'idée est d'utiliser HERACLES pour "nettoyer" le flux reçu par le récepteur avant le décodage par le code correcteur d'erreurs. Ce système peut être utilisé autant en mode "hard" que "soft", mais il est évidemment plus performant en "soft". La Figure 4.10 présente le schéma d'un récepteur utilisant ce système. Dans ce système, lorsque HERACLES détecte un SP, il modifie les valeurs des coefficients soft de sorte à affecter un niveau de confiance important à ces symboles (voir Figure 4.11). Le décodeur qui suit utilise

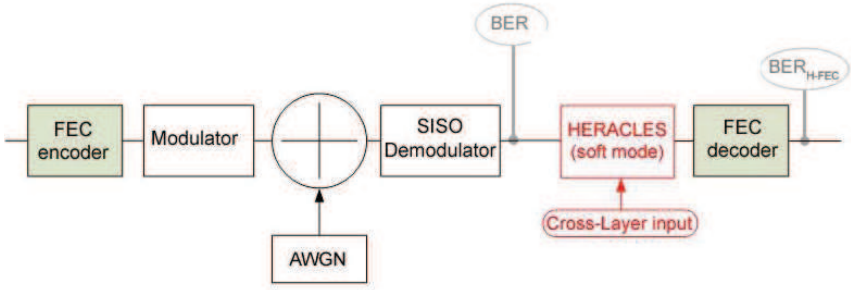


FIG. 4.10 – Concaténation de HERACLES et d'un décodeur soft

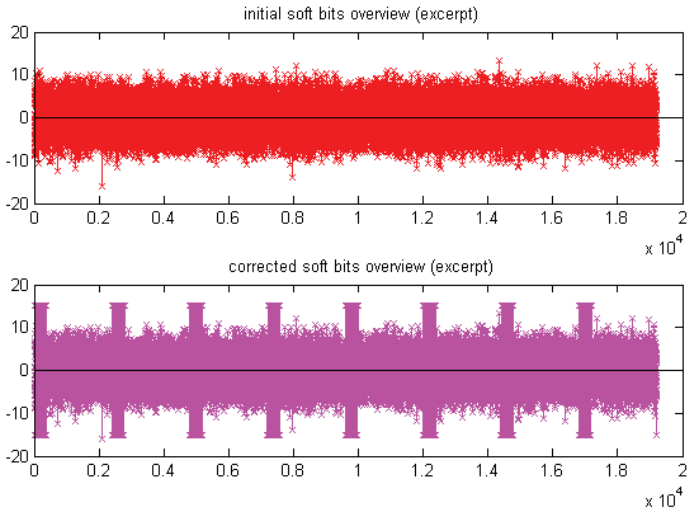
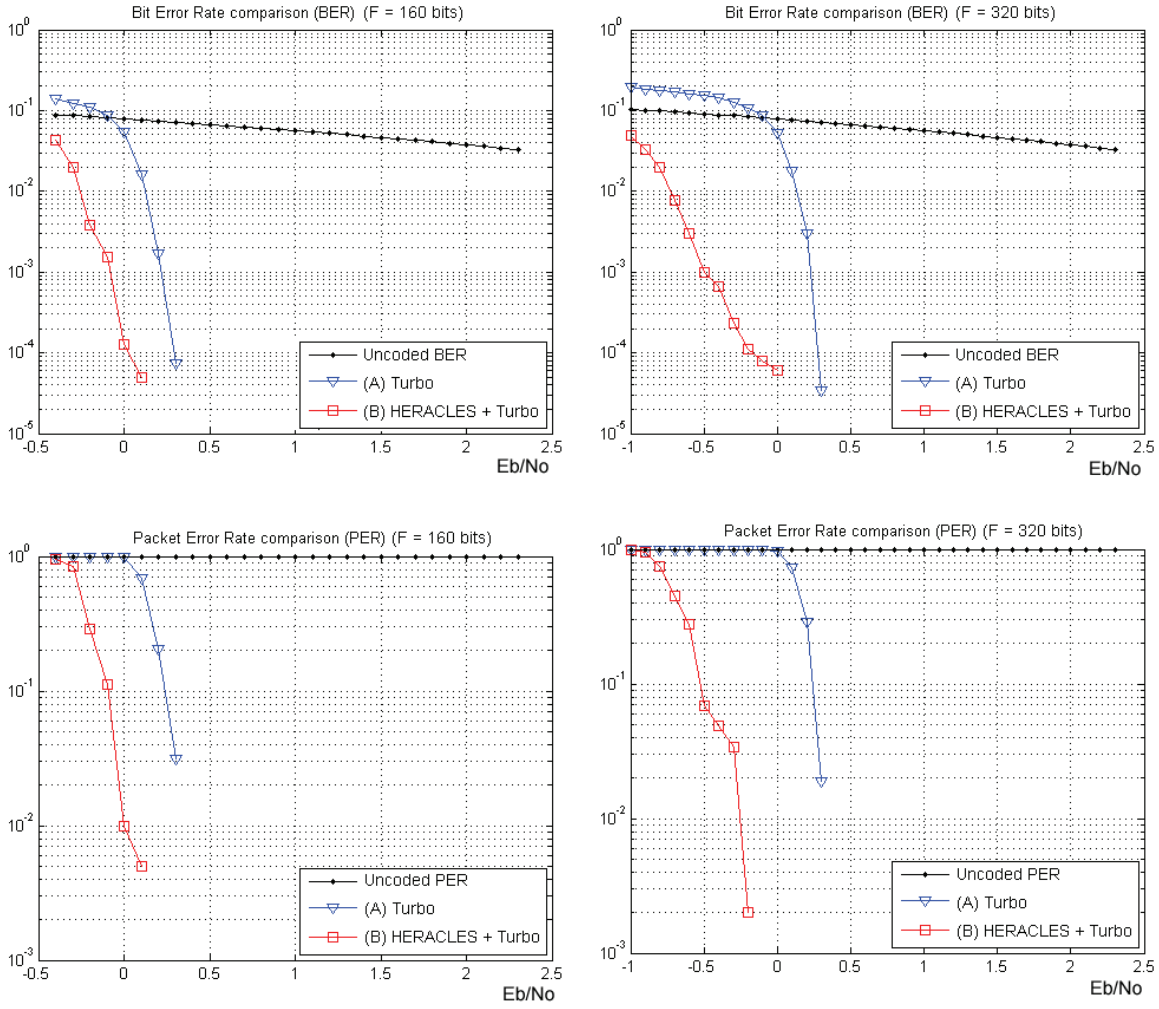


FIG. 4.11 – Modification des informations soft par HERACLES

alors ces symboles dans son décodage global. Si l'on suppose que le décodage de HERACLES est parfait, le décodeur reçoit une partie de ses symboles d'information sans erreur, ce qui est équivalent à faire une opération de raccourcissement ("shortening") du code, le taux de codage du code est ainsi multiplié par $1 - \frac{F}{L}$.

La Figure 4.12 présente un exemple des gains obtenus en termes de taux d'erreur bit résiduel (BER) et de taux d'erreur paquets résiduel (PER) lorsque le code utilisé est le turbo-code 3GPP2 [1] avec un taux de codage 1/3 et avec des blocs d'information de 12282 bits. Ces courbes montrent des gains significatifs. Pour des valeurs de $F = 20$ et $L = 100$, le gain



(a) $F = 20$ octets et $L = 100$ octets

(b) $F = 40$ octets et $L = 100$ octets

Fig. 4.12 – Taux d'erreur bit et taux d'erreur paquet pour $F = 20$ et $F = 40$ octets et $L = 100$. Le code utilisé est le turbo-code 3GPP2 [1] aussi utilisé dans DVB-SH

obtenu par le système HERACLES+Turbo sur le Turbo-code seul est de l'ordre de 0.25 dB à $BER = 10^{-3}$. Pour des valeurs de $F = 40$ et $L = 100$, le gain est de l'ordre de 0.75 dB à $BER = 10^{-3}$.

4.3.6 Discussion

Bien que nous n'ayons pas trouvé de travaux reprenant la même idée dans la littérature, plusieurs propositions ont des points communs avec HERACLES.

Par rapport à la correction des erreurs sur les entêtes, [77] propose des mécanismes de correction d'entêtes utilisant les différentes formes de redondance présentes dans les entêtes comme les champs répétés d'une paquet à l'autre ou CRC. La combinaison des techniques de correction utilisées dans cette approche avec HERACLES, notamment lorsque ce dernier est utilisé avant le décodeur FEC, nous semble un point intéressant à étudier.

Un autre ensemble de publications et de brevets relatifs à ce travail est le système DUDE [82] dont le but est d'identifier la redondance dans un flux quelconque. Le fait de placer ce mécanisme avant le décodeur sur la couche physique a été proposé. Toutefois, à notre connaissance, aucune performance du type de celle montrée sur les Figures 4.12 n'a été proposée.

Ceci pourrait être dû au fait que, pour être efficace dans ce contexte (avant le décodeur), le pouvoir de correction du mécanisme doit être supérieur ou au moins comparable à celui du code. Les points relatifs à l'implémentation de notre mécanisme dans des systèmes réels sont discutés dans [83]

4.4 Conclusions et perspectives sur le codage multi-couches

Le codage multi-couches est assurément un thème de recherche encore très ouvert. Le grand nombre de mécanismes de fiabilité utilisables sur les différentes couches ainsi que leurs interactions potentielles génèrent un nombre de solutions importantes.

Les récentes évolutions de la communauté réseau, qui commence désormais à envisager dans certains cas l'utilisation de mécanismes cross-layer, ouvrent de nouvelles perspectives de recherche. Au moment où les codes correcteurs d'erreurs atteignent leur limites sur la couche physique, les mécanismes cross-layer peuvent constituer le moyen d'améliorer encore les performances générale du système. Un bon exemple d'amélioration potentielle est HERACLES qui, combiné avec le turbo-code, améliore de manière très significative les performances du turbo code seul.

Chapitre 5

Applications des codes correcteurs dans les réseaux

Les évolutions successives des différentes technologies de communication ouvrent régulièrement de nouvelles opportunités pour l'utilisation de techniques de codage sur les différentes couches réseaux.

L'événement le plus marquant de ces dernières années dans ce domaine est probablement l'introduction du codage réseau (network coding) [84] qui permet aux routeurs d'effectuer des opérations de codage sur les paquets reçus au lieu de simplement les transférer.

Les trois applications particulières des techniques de codage dans les réseaux que nous présentons dans ce chapitre sont (plus ou moins) liées à ce domaine. Toutefois, ces travaux se situent dans des contextes différents :

- réduction du nombre de retransmissions lorsque plusieurs transmissions point-à-point partagent le même canal de diffusion
- amélioration du temps de téléchargement de fichiers dans des réseaux pair-à-pair
- réduction des bornes du temps de traversée d'un réseau à qualité de service par un paquet

5.1 Réduction du nombre de retransmissions sur un canal à diffusion

Ce travail, réalisé fin 2004 avec Tanguy Pérennou, est la suite d'une étude que nous avions faite sur l'opportunité d'utiliser des codes à effacement sur la couche MAC de 802.11b [85]-[27]. Une première version a été publiée en 2005 [86], puis a été adaptée aux transmissions par satellite dans [28]. Une version complète doit être soumise très prochainement à un journal.

Le contexte de cette application est celui où plusieurs transmissions point-à-point partagent un même canal à diffusion. Ceci se produit, par exemple, lorsque plusieurs utilisateurs accèdent à l'Internet via une même borne Wifi ou via un même satellite.

L'idée de ce mécanisme est d'utiliser la propriété de diffusion du canal sur des transmissions point-à-point parallèles. Dans ce sens, ce travail peut être relié à l'architecture COPE présentée dans [87], non publiée à ce moment-là. Un autre travail publié très récemment, [88], a encore plus de points communs avec notre étude car sa proposition consiste à étendre COPE pour réduire le nombre de retransmissions.

Une autre façon de présenter ce mécanisme est de le voir comme un mécanisme H-ARQ

opportuniste.

5.1.1 Principe

La Figure 5.1 présente le principe de ce mécanisme. Supposons qu'un émetteur E trans-

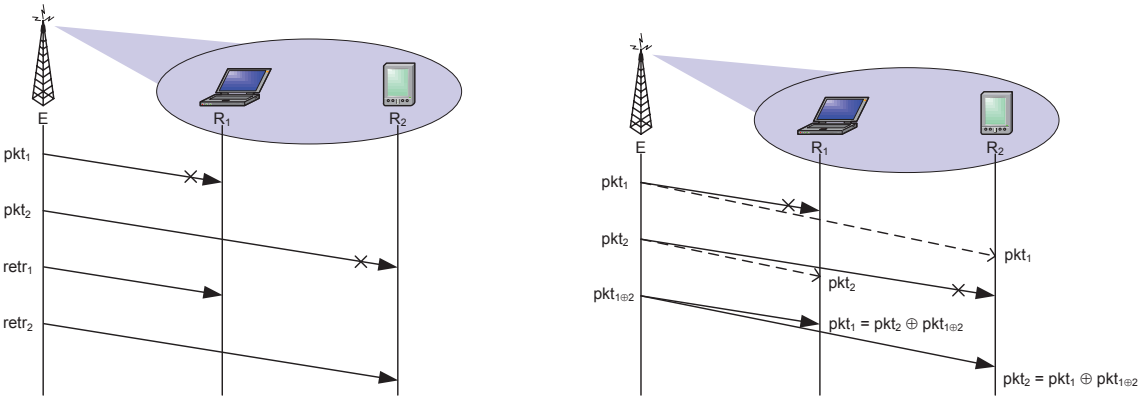


Fig. 5.1 – Retransmissions vs. décodage

mette sur le canal à diffusion un paquet pkt_1 à un récepteur R_1 et un paquet pkt_2 à un récepteur R_2. R_1 ne reçoit pas le paquet pkt_1 qui lui était destiné, mais entend pkt_2. Réciproquement, R_2 ne reçoit pas pkt_2 mais entend pkt_1.

Un mécanisme de retransmission classique imposerait à E de réémettre pkt_1 et pkt_2, ce qui conduit à une occupation du canal (par les paquets d'information) de 50%. Or, si E transmet un paquet $\text{pkt}_{1\oplus 2}$ calculé en faisant le `XOR` bit à bit de pkt_1 et de pkt_2, R_1 et R_2 peuvent retrouver les paquets qui leur étaient destinés à partir des paquets simplement entendus en effectuant les calculs de décodage suivants :

$$\text{pkt}_1 = \text{pkt}_2 \oplus \text{pkt}_{1\oplus 2}$$
$$\text{pkt}_2 = \text{pkt}_1 \oplus \text{pkt}_{1\oplus 2}$$

L'utilisation de l'opérateur `XOR` permet donc d'économiser un paquet de retransmission et de passer à un taux d'occupation du canal par les paquets d'information de 66%.

Comme le nombre de récepteurs est potentiellement supérieur à 2 et que le nombre de paquets de redondance nécessaires peut être important, nous avons généralisé le mécanisme en utilisant des codes de Reed-Solomon qui ont un capacité de correction supérieure à celle des codes binaires.

Afin d'illustrer notre proposition, nous faisons l'hypothèse d'un multiplexage temporel statique : si R utilisateurs reçoivent simultanément une transmission distincte de N paquets d'information chacun au niveau liaison. Si on note $\text{pkt}(i,j)$ la diffusion sur le canal du $i^{\text{ème}}$ paquet de données destiné au $j^{\text{ème}}$ récepteur, l'ordre de passage des paquets de données sur le canal avec un entrelacement dynamique est alors :

$$\text{pkt}(1,1),\ldots,\text{pkt}(1,R),\text{pkt}(2,1),\ldots,\text{pkt}(N,R).$$

Dans ces conditions, l'algorithme général de l'émetteur consiste à émettre un bloc B_n de $N \times R$ paquets d'information, à déterminer la liste complète L_n des paquets perdus par leur destinataire, à diffuser cette liste à tous les récepteurs, à considérer ces paquets comme les

k paquets d'information d'un nouveau code Reed-Solomon et à transmettre des paquets de redondance de ce nouveau code jusqu'à ce que tous les récepteurs aient pu décoder tous les paquets d'information du bloc B_n d'origine.

Nous avons aussi étendu ce mécanisme dans le cas d'un canal à erreurs (et non plus à effacement). Dans ce contexte, on considère que chaque paquet d'information est constitué de G blocs de b symboles. Avec les premiers blocs de chaque paquets, on constitue un bloc d'information (de longueur kb) que l'on code avec un code de Reed-Solomon systématique de paramètres $[nb, kb]$. Les $nb - kb$ symboles de redondance sont découpés en $n - k$ blocs de b symboles. Chacun de ces blocs forme alors le premier bloc d'un paquet de redondance. On réitère cette opération avec les deuxièmes blocs des paquets d'information pour générer les deuxièmes blocs des paquets de redondance et ainsi de suite.

Le bloc des paquets d'information et de redondance est alors constitué de G mots d'un code de Reed-Solomon de paramètres $[nb, kb]$.

5.1.2 Evaluation théorique

Ce mécanisme a été évalué théoriquement pour le canal à effacement ainsi que pour le canal à erreur. Le paramètre que nous avons mesuré est le nombre moyen de retransmissions.

Pour le canal à effacement, nous avons considéré le cas où les pertes apparaissent indépendamment avec une probabilité p. Pour le canal à erreur, nous avons considéré que les paquets erronés apparaissent indépendamment avec une probabilité p et que, dans ces paquets erronés, les erreurs apparaissent indépendamment avec un probabilité q.

Nous avons choisi de prendre comme point de repère le nombre de retransmissions faites par un mécanisme de retransmission classique ARQ pour un bloc de $N \times R$ paquets sur le canal à effacement. On peut montrer que ce nombre, noté O_{ARQ}, est égal à

$$O_{\text{ARQ}} = \frac{N \times R \times p}{1 - p} \tag{5.1}$$

Si O_{HARQ} représente le nombre de retransmissions avec notre mécanisme (sur le canal à effacement ou à erreur), le gain est défini par

$$\text{Gain} = 100 \times \frac{O_{\text{ARQ}} - O_{\text{HARQ}}}{O_{\text{ARQ}}} \tag{5.2}$$

Sous ces hypothèses, nous avons pu déterminer combinatoirement les gains obtenus avec cette méthode sur les différents canaux. Les démonstrations sont détaillées dans les papiers pré-cités.

5.1.3 Résultats

Les résultats ont montré que le gain obtenu par notre mécanisme était généralement très significatif. Par exemple, la Figure 5.2 montre les gains obtenus sur le canal à effacement pour différentes valeurs de N, R et p. Sans surprise, ces gains augmentent avec la longueur des codes (c'est-à-dire avec le produit NR). Ils peuvent dépasser 60% pour des valeurs de $p = 0.01$ et $p = 0.15$. Un point intéressant est que des gains significatifs sont aussi obtenus pour des petites valeurs de N et R, ce qui est un avantage dans la perspective d'une implémentation réelle. Par exemple, un gain d'environ 31% est obtenu dans les 2 cas pour $N = R = 4$. Sur le canal à erreur, les gains sont supérieurs. Ils peuvent atteindre 99% lorsque le taux de paquets

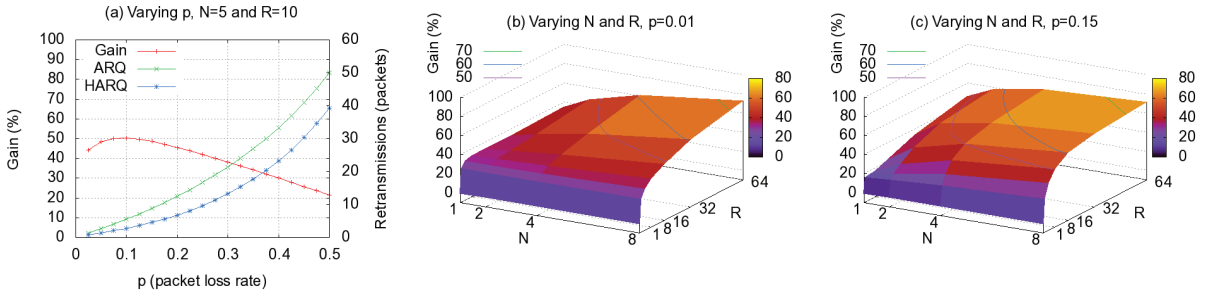


Fig. 5.2 – canal à effacement : Influence de N et R sur le gain

erronés est "important" ($p = 0.15$) et lorsque le taux d'erreur dans ces paquets est faible ($q = 0.01$). A contrario, dans le cas d'un faible taux de paquets erronés ($p = 0.01$) et d'un fort taux d'erreur dans ces paquets (0.3), le gain est plus faible. Dans ce cas-là, pour la même valeur de p, les résultats sont identiques (parfois meilleurs) sur le canal à effacement. Il faut

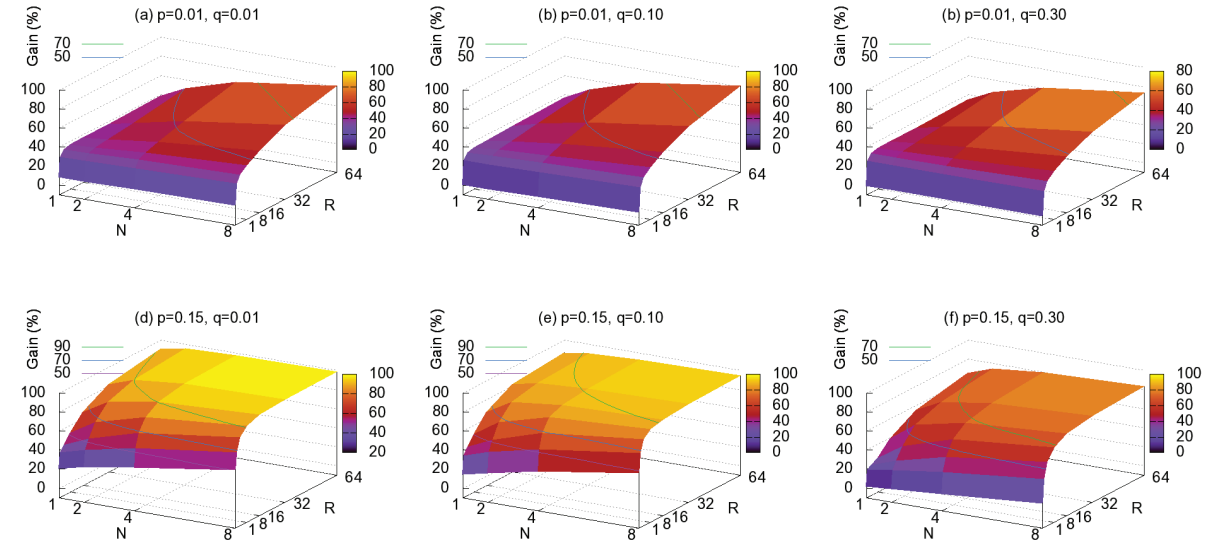


Fig. 5.3 – canal à erreurs : Influence de N et R sur le gain

noter que les résultats sur le canal à erreur dépendent aussi des valeurs de G et de b [28]. Clairement, les gains augmentent avec la longueur des mots, c'est-à-dire avec la valeur de b. Pour des paquets de taille fixée, il est préférable de prendre des valeurs de G le plus petite possible, ce qui augmentera la valeur de b, les performances du mécanisme, mais aussi la complexité du décodage.

5.1.4 Conclusion

Les différents contraintes liées à l'implémentation sont discutés dans [28]. Il est en particulier rappelé que plusieurs des couches physique de dernière génération, comme 802.16e 2005 "mobile Wimax" [18] ainsi que 3GPP High-Speed Downlink Packet Access (HSDPA) [17]

utilisent un mécanisme H-ARQ mettant en place tous les outils nécessaires à l'implémentation de notre mécanisme.

5.2 PeerFect

Si les techniques de codage correcteurs ont été principalement utilisées pour augmenter la fiabilité (de manière générale), le fait d'augmenter la diversité des données transmises ou stockées a aussi des intérêts pour d'autres problématiques.

Le système PeerFect, que nous avons proposé en 2002 [45] [36], a été, à notre connaissance, le premier à montrer que les codes à effacement pouvaient réduire les temps de téléchargement dans des réseaux pair-à-pair. En effet, le travail proposé dans [89] utilise des codes à effacement dans les réseaux pair-à-pair, mais uniquement dans le but d'augmenter la durabilité des données dans le réseau. L'augmentation de la diversité dans le réseau, qui se traduit par une diminution des durées de téléchargement, a depuis été améliorée par l'introduction du codage réseau dans les réseaux pair-à-pair [90] [91].

PeerFect a été construit dans le cadre de la thèse de Laurent Lancérica encadrée par Laurent Dairaine et Christian Fraboul.

5.2.1 Présentation de PeerFect

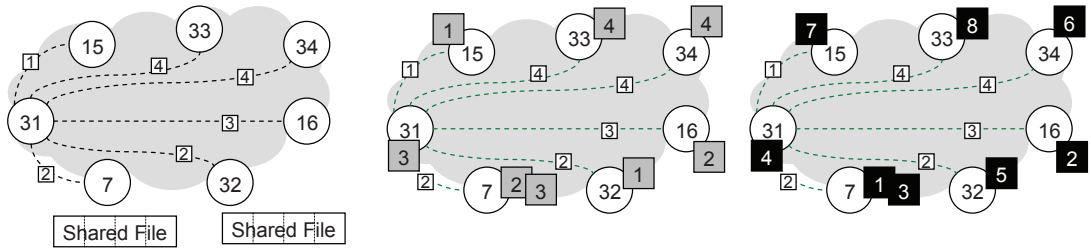
L'objectif de ce travail est de réduire le temps de téléchargement dans un réseau pair-à-pair. Nous supposons que ce réseau contient une ou plusieurs copies d'un fichier donné. Ces copies peuvent être scindées en blocs (codés ou non) pouvant être distribués sur des machines distantes du réseau. Nous considérons aussi qu'une fonction de recherche nous permet de connaître toutes les positions de ce fichier ou des blocs de fichiers. Enfin, nous considérons que des outils permettent de mesurer la bande passante entre 2 machines du réseau.

Notre proposition consiste à scinder le fichier en k blocs qui sont considérés comme l'information d'un code à effacement de paramètres $[n, k]$. Ce code permet de générer $n - k$ blocs de redondance. L'ensemble de ces blocs est alors distribué sur différentes machines du réseau. Nous supposons que le code est MDS, c'est-à-dire que tout ensemble de k blocs permet de retrouver le fichier.

Un utilisateur voulant télécharger ce fichier le plus rapidement possible doit localiser les k blocs accessibles avec la plus grande bande passante, les télécharger, puis les décoder pour récupérer le fichier initial.

Nous avons cherché à évaluer la différence de temps de téléchargement d'un fichier par rapport à des stratégies basées sur la réplication des fichiers ou des blocs de fichier. Un exemple est présenté sur la Figure 5.4. Dans ce schéma, l'équivalent de 2 fois la taille du fichier initial est distribué sur le réseau. La première stratégie réplique simplement le fois sur 2 nœuds du réseau. La seconde découpe le fichier en 4 blocs et distribue 2 copies de chaque bloc sur le réseau. La troisième stratégie découpe le fichier en 4 blocs puis génère 8 blocs codés (avec un code MDS) qui sont distribués sur le réseau. Notons que si le code est systématique, les blocs d'information font partie des blocs codés.

On suppose que le nœud 31 veut récupérer le fichier. Avec la première stratégie, le fichier est téléchargé à partir du nœud 7 (ou du nœud 32) pour un temps de téléchargement total de 8 unités de temps. Avec la deuxième stratégie, le meilleure solution consiste à télécharger le bloc 1 pour un coût de 1, le bloc 2 pour une durée de 2, le bloc 3 pour une durée de 0, le bloc 4 pour une durée de 4, ce qui représente au total une durée de 7 unités de temps. Avec



(a) 2 copies du fichier sont stockées dans le réseau

(b) le fichier est scindé en 4 blocs qui sont 2 répliqués 2 fois dans le réseau

(c) 8 blocs codés sont générés à partir des 4 blocs initiaux et sont distribués dans le réseau

Fig. 5.4 – Stratégies de réplication d'un fichier dans un réseau pair-à-pair. La quantité de données équivalente à 2 copies est stockée dans le réseau pour les 3 stratégies. Les indications sur les liens correspondent au temps nécessaire pour télécharger un bloc

la dernière stratégie, il suffit de télécharger les 4 blocs les plus proches, ce qui représente une durée de 0+1+2+2=5 unités de temps.

Notons que ces durées sont calculées avec l'hypothèse que les téléchargements sont séquentiels. En pratique, comme on peut le voir que la Figure 5.5, les débits de téléchargement ne varient pas si l'on télécharge de manière séquentielle ou parallèle. Le seul goulot d'étranglement se situe du coté du nœud qui télécharge. Dans la suite, nous considérerons le cas de téléchargement des blocs en parallèle.

5.2.2 Evaluation des performances

Pour évaluer cette solution de manière aussi réaliste que possible, nous avons tout d'abord modélisé les temps de téléchargement des 3 stratégies présentées ci-dessus en fonction des bandes passantes disponibles dans un réseau pair-à-pair, ces bandes passantes étant considérées comme des variables aléatoires.

Cette modélisation, qui n'est pas détaillée ici, est présentée dans [36].

Nous avons ensuite réalisé des mesures sur le réseau pair-à-pair Gnutella, qui était le principal réseau pair-à-pair à ce moment-là. Ces mesures nous ont permis d'établir la distribution de la bande passante présentée sur la Figure 5.5. Cette Figure confirme que les débits de téléchargement ne varient pas si les téléchargements se font en séquence ou en parallèle. En combinant la modélisation proposée avec la distribution observée, nous avons obtenu les résultats présentés sur les Figures 5.6 et 5.7. Nous avons noté r le nombre de copies du fichier ou des blocs non codés et r' le nombre de copies des blocs codés. Le gain mesuré est la diminution (en pourcentage) du temps de télechargement par rapport à la stratégie de réplication des fichiers. La Figure 5.6(a) montre l'évolution du gain en fonction du nombre de blocs k. La stratégie utilisant le codage est toujours meilleure que celles sans codage. Elle atteint un peu plus de 60% pour des valeurs de $k \geq 10$. Cette valeur de $k = 10$ semble le meilleur compromis entre les performances en termes de gain et la complexité du décodage.

Nous avons aussi évalué l'influence du facteur de redondance sur la gain. Il est intéressant de noter qu'un facteur de l'ordre de 2 ou 3 permet d'obtenir un gain proche de l'optimal. La Figure 5.7(a) montre que PeerFect est surtout intéressant lorsque le nombre total de blocs dans le réseau est limité. En effet, à partir d'un certain niveau de réplication, la disponibilité du fichier augmente naturellement et les gains relatifs sont donc inférieurs.

Enfin, la dernière Figure 5.7(b) montre que, pour des téléchargements en parallèle, les

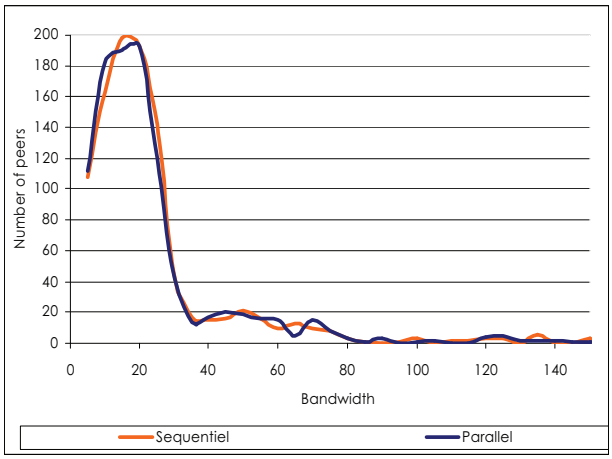
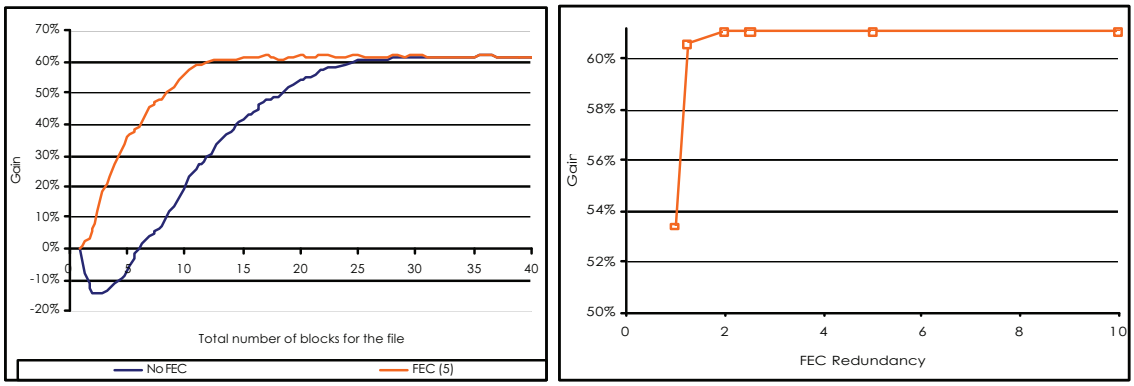


Fig. 5.5 – Distribution des bandes passantes disponibles sur le réseau Gnutella en janvier 2003



(a) Influence du nombre de blocs k par fichier. $n = 5.k$, $r = 10$ et $r' = 2$.

(b) Influence du facteur de redondance n/k . $k = 20$, $r = 10$ et $r' = r.k/n$

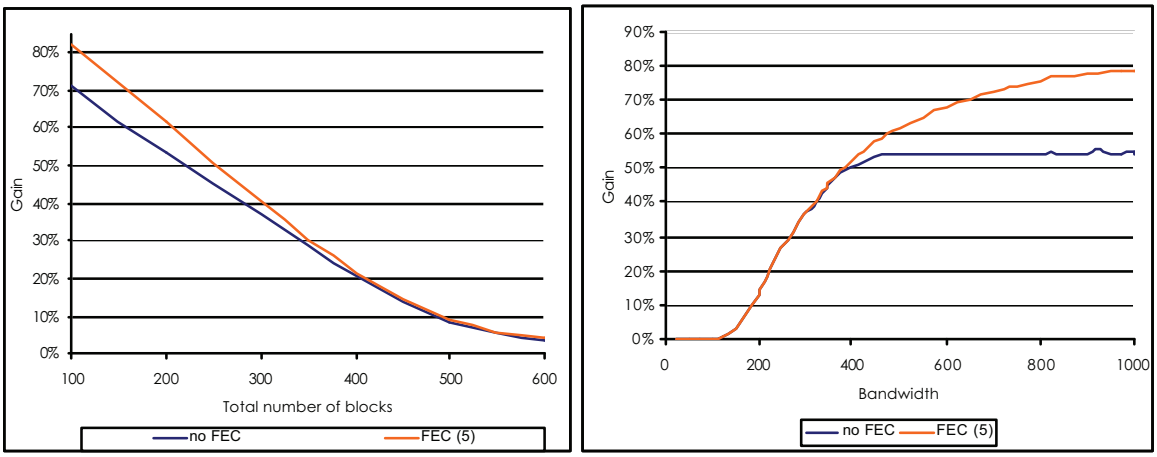
Fig. 5.6 – Influence du nombre de blocs par fichier et du facteur de redondance

gains ne sont réels que si le débit du réseau d'accès est suffisant. Avec les données étudiées, ce seuil se situe autour de 350 Kb/s.

5.2.3 Discussion et Conclusion

Une application de ces résultats à la vidéo a été proposée dans [92] où nous avons évalué le taux de réplication des différentes parties d'une vidéo dans un réseau pair-à-pair pour que celle-ci puisse être visualisée en streaming par un utilisateur. Ce taux de réplication est évidemment réduit grâce à l'utilisation des codes à effacement.

Un autre résultat de ces travaux est le choix du code. Notre position est que les codes les plus adaptés à cette application sont les codes MDS. L'argument principal est qu'avec un découpage approprié des blocs de données, le décodage peut se faire en même temps que le téléchargement des blocs en parallèle. Or les résultats de simulations présentés dans le tableau 3.1 montrent que l'on peut facilement choisir un code MDS dont la dimension correspond au nombre de blocs téléchargés en parallèle et dont la vitesse de décodage est très largement supérieure à la vitesse de téléchargement globale. A contrario, le choix d'un code non MDS



(a) Influence du nombre de blocs total $k.r$ dans le réseau. $k = 20$, $n/k = 5$ et $r' = r.k/n$

(b) Influence de la bande passante au niveau du nœud téléchargeant le fichier. $k = 20$, $n/k = 5$ et $r' = r.k/n$

Fig. 5.7 – Influence du nombre de blocs dans le réseau et du débit disponible

pourra générer de l'inefficacité dans le choix des blocs à télécharger.

5.3 Réduction des bornes des temps de traversée d'un réseau à garantie de qualité de service avec le codage réseau

5.3.1 Introduction

Le résultats présentés dans cette section ont été obtenus dans le cadre de la thèse de Ali Mahmino, co-encadrée avec Christian Fraboul. Ces résultats sont été publiés dans [93, 94, 95] et un papier de revue est en cours de soumission.

Ce travail concerne une application particulière du codage réseau. Le but de ce travail est d'évaluer si le codage réseau peut permettre de réduire la borne théorique du temps de traversée d'un réseau à garantie de qualité de service par un paquet. L'application d'origine de ce travail concerne les réseaux embarqués avioniques, et notamment l'AFDX. Dans ce contexte, les règles de certification imposent des bornes strictes sur les temps de traversée du réseau par un paquet. Comme le temps de traversée en pire cas est étroitement lié aux congestions pouvant se produire dans le réseau, nous avons pensé que le codage réseau, dont les noeuds de codage traitent simultanément des paquets de plusieurs flux, pouvait réduire ces bornes de temps de traversée.

Pour calculer des bornes des temps de traversée des réseaux à qualité de service, il est classique d'utiliser la théorie du calcul réseau (Network Calculus) introduite par J. Y. Le Boudec et P. Thiram [96] à la suite de plusieurs autres travaux dont ceux de R. L. Cruz [97][98]. Pour déterminer ces bornes, nous avons dû combiner codage réseau et calcul réseau[1].

Les hypothèses considérées dans ce travail sont les suivantes :

– Le réseau est représenté par un graphe acyclique (G, V) où G est l'ensemble des nœuds et V est l'ensemble des liens. La topologie de ce réseau est fixée.
– Chaque lien a une capacité donnée.

[1] nous considérons que le lecteur est familier avec les notions du calcul réseau (voir [96])

- Chaque élément réseau (lien, nœud) garantit un service caractérisé par un courbe de service connue.
- Les sources génèrent des flux de paquets de longueur L. Elles peuvent être temporairement inactives. Ces sources ne sont pas synchronisées, par contre, elles partagent la même horloge.
- Les flux vérifient des contraintes caractérisées par des courbes d'arrivée.
- Le code réseau est déterminé *a priori*. Ceci implique que chaque nœud de codage combine de manière fixée les différents flux entrants pour produire les flux sortant.

Nous avons proposé 3 stratégies permettant d'appliquer le codage réseau dans des réseaux à qualité de service.

5.3.2 Stratégie orientée réseau

Cette première stratégie a été définie pour réduire au maximum les quantités de données dans les files d'attente des nœuds de codage. Dans cette optique, nous avons proposé de réaliser un codage "à la volée" des paquets présents dans un nœud à un instant donné. Les paquets sont ainsi combinés de manière aléatoire comme dans [99], même si la topologie du réseau est fixée. Cette stratégie, qui évite les synchronisations entre des paquets de la même génération, peut être analysée avec des outils de calcul réseau classique.

Analyse au niveau d'un nœud de codage

Nous considérons que les n flux R_i, pour $i = 1, 2 \ldots, n$ entrant dans un nœud de codage sont respectivement contraints par les courbes d'arrivée $\alpha_i(t) = \sigma_i + L * v_{L/\rho_i, -L/\rho_i}(t)$ si $t > 0$ et 0 sinon. Soit $\rho = max_{i=1,\ldots,n}(\rho_i)$ et $T = L/\rho$. Nous définissons aussi $R^{j,out}$ la fonction cumulative du $j^{\text{ème}}$ flux de sortie du nœud et $R_i^{j,out}$ comme la fonction cumulative du sous-ensemble de données de $R^{j,out}$ contenant des données de R_i. Ces données sont soit combinées avec d'autres données soit simplement multipliées par un coefficient.

Le nœud de codage est composé de n files d'attente (Leaky Bucket Shapers) synchronisées permettant de remodeler les flux entrants de telle sorte à faire sortir un paquet tous les T unités de temps. Chaque file d'attente offre donc le service $L * v_{T,0}(t)$ au flux correspondant. Les n paquets (au maximum) sortant des files d'attente simultanément sont alors multipliés par le coefficient défini par le code réseau et sont sommés pour produire le paquet du flux sortant. On considère que ces opérations ajoutent un délai de codage supplémentaire majoré par T_{lc}, et représenté par une courbe de service $\delta_{T_{lc}}$

Avec ces hypothèses, nous avons le théorème suivant :

Théorème 5.3.1 *La courbe du service offert par le nœud de codage assure que, pour $i = 1, 2, \ldots, n$:*

1. *La courbe de service β_i offerte par le nœud à R_i est égale à $L * v_{T,-T-T_{lc}}(t)$.*
2. *le délai maximum d'un paquet de R_i dans le nœud est $T_{lc} + T(1 + \sigma_i/L)$.*
3. *la quantité de données maximale dans la $i^{ème}$ file d'attente est $\sigma_i + L$.*
4. *$R_i^{j,out}$ est contraint par $\alpha_i \oslash L * v_{T,-T}$.*
5. *$R^{j,out}$ est contraint par $(\alpha_1 \oslash L * v_{T,-T}) \vee (\alpha_2 \oslash L * v_{T,-T}) \vee \ldots \vee (\alpha_n \oslash L * v_{T,-T})$*

Analyse au niveau du réseau

Pour simplifier les notations, nous considérons dans cette partie que le $i^{\text{ème}}$ flux entrant est contraint par la courbe d'arrivée $\alpha_{i,in}$ et que le nœud de codage lui offre le service $\beta_{i,j}$ vers

le flux $R^{j,out}$. D'après la partie précédente, $R^{j,out}$ est contraint par la courbe d'arrivée $\alpha_{j,out}$ où :

$$\alpha_{j,out} = \alpha_{1,in} \oslash \beta_{1,j} \vee \alpha_{2,in} \oslash \beta_{2,j} \vee \ldots \vee \oslash \alpha_{r,in} \oslash \beta_{r,j}$$

Notre but est de déterminer le service offert par le réseau aux flux qu'il transporte. En d'autres termes, nous voulons déterminer une matrice de transfert M, dont les termes sont des courbes de service, telle que

$$[\alpha_1^*, \ldots, \alpha_n^*] = [\alpha_1, \ldots, \alpha_k] \oslash M \tag{5.3}$$

où les $\alpha_i,\ i = 1, \ldots, k$ sont les courbes d'arrivée des flux entrants dans le réseau et les $\alpha_j^*,\ j = 1, \ldots, n$ sont les courbes d'arrivée des flux reçus par les récepteurs.

Cet objectif a été atteint en établissant une parallèle entre le codage linéaire des paquets dans le réseau et les opérations correspondantes sur les courbes d'arrivée et les courbes de service.

En effet, on peut observer que lorsque qu'un nœud réalise un combinaison linéaire des paquets entrant sur un corps fini , comme par exemple

$$y = x_1 * b_1 + x_2 * b_2$$

où $x_1,\ x_2$ et y sont des variables aléatoires représentant les flux et b_1 et b_2 sont des éléments d'un corps fini, les opérations réalisées par ce nœud ont les conséquences suivantes sur les courbes d'arrivée des flux :

$$Y = X_1 \oslash \beta_1 \vee X_2 \oslash \beta_2$$

où $X_1,\ X_2$ et Y sont les courbes d'arrivées et β_1 et β_2 sont des courbes de service.

Grâce à ce rapprochement, on peut utiliser la construction de la matrice de transfert M présentée dans [100] pour construire notre matrice de transfert opérant dans l'algèbre du calcul réseau. Cette construction est détaillée dans [93]. En appliquant cette construction au

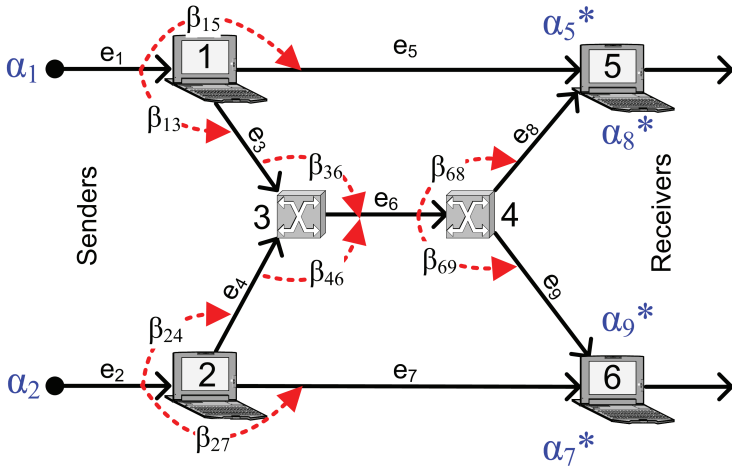


Fig. 5.8 – Exemple du réseau "papillon" où sont représentés les courbes d'arrivées des différents flux ainsi que les courbes de service des différents nœuds.

réseau de la Figure 5.8, on obtient :

$$\begin{bmatrix} \alpha_5^* & \alpha_8^* & \alpha_9^* & \alpha_7^* \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix} \oslash \begin{bmatrix} \beta_{15} & \beta_{13} \otimes \beta_{36} \otimes \beta_{68} & \beta_{24} \otimes \beta_{46} \otimes \beta_{68} & 0 \\ 0 & \beta_{13} \otimes \beta_{36} \otimes \beta_{69} & \beta_{24} \otimes \beta_{46} \otimes \beta_{69} & \beta_{27} \end{bmatrix}$$

D'un point de vue pratique, le principal intérêt de cette stratégie est sa flexibilité car elle est capable d'intégrer les variations du débit des flux dans le réseau. En revanche, comme toute stratégie de codage aléatoire, elle ne fournit pas une fiabilité totale car certaines configurations de combinaisons linéaires ne sont pas décodables par les récepteurs. Même si cette proportion de configurations non décodables peut être aussi réduite que l'on veut [99], ceci n'est pas acceptable pour certains types de réseau. Dans les parties suivantes, nous présentons des stratégies fournissant une fiabilité totale.

5.3.3 Stratégie orientée flux

Comme indiqué précédemment, le but de cette stratégie est d'offrir une fiabilité totale au flux. Dans cette optique, nous avons repris le concept de génération introduit dans [101] et nous l'avons adapté à notre contexte.

Notre proposition consiste à associer chaque génération à un intervalle de temps donné. On suppose que chaque source génère au plus un paquet dans l'intervalle $[t_i, t_i + \Delta]$ et que les paquets générés par les différentes sources appartiennent à la même génération. Comme dans [101], les combinaisons linéaires de paquets ne peuvent se faire qu'entre paquets de la même génération. Lorsqu'une source ne génère pas de paquet pendant l'intervalle correspondant à une génération, on considère qu'elle a généré un paquet nul, ce qui ne perturbe pas le décodage du récepteur. Notons que la contrainte d'un seul paquet par intervalle de temps de longueur Δ implique que le débit maximum des flux est de L/Δ bits/sec.

Nous supposons aussi que chaque lien $e_{i,j}$ a une capacité de $C_{i,j}$ bits/sec, où $C_{i,j} > L/\Delta$. Comme le système offre une garantie de qualité de service, nous considérons, que pour tout lien $e_{i,j}$, le délai de transmission maximum d'un paquet de taille L est égal à $L/C_{i,j} + T_{e_{i,j}} = \omega_{i,j} + T_{e_{i,j}}$.

De plus, contrairement à la partie précédente, nous considérons que le nœud de codage n'a qu'une seule file d'attente qui reçoit les paquets de tous les flux entrants. Avec ces hypothèses, la première stratégie que nous proposons pour un nœud de codage est définie de la manière suivante. La combinaison linéaire de la génération i est réalisée dès qu'une des conditions suivantes est vérifiée pour chacun des flux d'entrée :

- le paquet de la génération i est dans la file
- le paquet de la génération i n'est pas dans la file mais le paquet de la génération $i+1$ est dans la file.
- le paquet de la génération i n'est pas dans la file et la date d'arrivée maximale de la génération i est dépassée.

Notons que la dernière condition implique que chaque nœud connaisse la date d'arrivée maximale de chaque génération. Ceci peut être calculée par chaque nœud à partir des dates de sortie maximale des nœuds en amont.

Pour calculer la borne du temps de sortie d'un paquet, seule la dernière condition doit être prise en compte. La borne dépend donc seulement de la date d'arrivée maximale. En généralisant cette approche à tout le réseau, nous avons pu déterminer le temps de traversée maximum du réseau par un paquet. Ces résultats sont détaillés dans [94].

5.3.4 Stratégie de transmission rapide

Le principal défaut de la stratégie précédente est son mauvais comportement lorsque certains flux sont inactifs. En effet, dans ce cas-là, les autres flux "attendent" ce flux dans les

routeurs jusqu'à la date limite d'arrivée, ce qui pénalise les temps de traversée de tous les paquets.

Pour améliorer ce problème, nous avons proposé une "stratégie de transmission rapide" (Fast forwarding strategy - FFS). Celle-ci reprend les hypothèses de la partie précédente, et ajoute une hypothèse supplémentaire : tout paquet de données contient un champ de contrôle de type somme de contrôle (checksum) vérifiant la propriété suivante : cette fonction ne doit pas être linéaire dans le corps fini ou dans tout sous-corps du corps fini utilisé par le code réseau. Les sommes de contrôle utilisés par UDP ou par IP vérifient cette condition. C'est aussi le cas de la fonction de hachage MD5, utilisée dans certains protocoles. Par contre, les CRC (Cyclic Redundancy Check) ne la vérifient pas car ils peuvent être vus comme des restes de la division polynomiale de polynômes binaires. Ils sont donc linéaires sur le corps $\mathbb{F}_2$ et ne peuvent pas être utilisés pour cette application.

Sous ces conditions, supposons qu'un paquet de la génération X arrive dans le nœud de codage $n+1$ à l'instant t. Le nœud peut alors appliquer la stratégie suivante :

- Si la file d'attente est vide, le paquet est multiplié par le coefficient du corps fini déterminé par le code réseau et est transmis sur le lien de sortie (si ce lien n'est pas en train d'être utilisé pour la transmission d'un autre paquet débutée avant la date t).
- Si la file d'attente n'est pas vide :
  - si la file d'attente ne contient aucun autre paquet de la génération X, le paquet est multiplié par le coefficient du corps fini et est ajouté à la fin de la file d'attente. Par exemple, sur la Figure 5.9, le paquet P_3^1 qui vient du nœud N_1 est ajouté à la fin de la file d'attente.
  - si la file d'attente contient un paquet de la génération X, le paquet est multiplié par le coefficient du corps fini et est ajouté au paquet de sa génération dans la file. Par exemple, sur la Figure 5.9, le paquet P_5^2 qui vient du nœud N_2 est ajouté au paquet P_5^1 déjà présent dans la file.

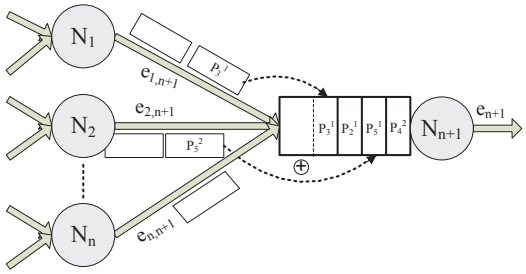


Fig. 5.9 – stratégie de transmission rapide

Au niveau du décodage, dès que le récepteur commence à recevoir un ou des paquets d'un génération, il les multiplie par la matrice inverse (connue car le code réseau est fixé), puis vérifie les sommes de contrôle sur les paquets obtenus. Si ces sommes sont correctes, il y a une très forte probabilité que tous les paquets de cette génération soient reçus. Si ce n'est pas le cas, il attend les autres paquets de cette génération. A la réception de ces nouveaux paquets, il met à jour les paquets obtenus et vérifie de nouveau les sommes de contrôles.

Notons que cette stratégie implique qu'aucun paquet ne soit corrompu ni perdu. Ceci est généralement le cas dans un réseau fournissant des garanties de qualité de service.

Le calcul de délai maximum d'un paquet dans un nœud de codage se base sur le calcul du nombre maximum de paquets présents dans la file d'attente à son arrivée. Ce nombre de

paquets se calcule à partir du délai minimum d'arrivée et du délai maximum d'arrivée d'un paquet d'une génération donnée. Ces délais s'obtiennent récursivement à partir des nœuds précédents.

Il est à noter que cette stratégie peut générer du déséquencement de paquets, comme sur la Figure 5.9. Elle peut aussi générer le cas où plusieurs paquets de la même génération utilisent un même lien lors qu'une stratégie de codage réseau "optimale" en termes d'utilisation du lien aurait fait passer un seul paquet (qui serait la somme de ces 2 paquets) en lieu et place du dernier paquet.

Toutefois, cette stratégie nous paraît être la plus souple. Par exemple, lorsqu'un seul flux est actif, elle ne ralentit pas les paquets dans le réseau et se comporte comme le ferait une stratégie de routage classique. *A contrario*, lorsque tous les flux sont actifs et génèrent un paquet par génération, elle se comporte comme un stratégie de codage optimale. Pour résumer, par construction, les paquets commencent à être combinés lorsque les files d'attentes commencent à se remplir.

5.3.5 Comparaison des bornes de temps de traversée du réseau pour les stratégies FOS et FFS

Tout d'abord, il faut noter qu'il est difficile de comparer ces 3 stratégies car la première n'est pas totalement fiable. Nous nous contenterons donc de comparer les 2 dernières stratégies. Nous intégrerons dans cette comparaison la stratégie de routage classique car ses bornes sont elles aussi calculables avec les outils de calcul réseau développés dans [96].

Il n'est pas possible de déterminer de manière définitive quelle est la meilleure stratégie car ceci varie en fonction des propriétés du réseau considéré. Dans un réseau sur-dimensionné, la stratégie du routage sera probablement la plus intéressante. Dans un réseau où tous les flux sont parfaitement synchronisés et où la longueur des intervalles des générations est petite, la stratégie FOS sera certainement la meilleure. Enfin, dans un réseau où les flux ont un débit variable mais où peuvent apparaître des congestions, la stratégie FFS obtiendra les meilleurs résultats.

A titre d'exemple, nous avons effectué la comparaison de ces trois stratégies sur le réseau présenté sur la Figure 5.10. Dans ce réseau, les 3 sources transmettent leurs flux vers les 3 récepteurs. On peut observer que les stratégies de codage réseau et de routage atteignent toutes la capacité du réseau.

Par contre, en calculant les bornes de traversée du réseau, on peut montrer que la stratégie FFS est meilleure que la stratégie FOS. Entre FFS et la stratégie de routage, on peut montrer que FFS obtient une meilleure borne si et seulement si :

$$T_{lc} \leq \frac{\tau}{3} + T(1 - \frac{3.L}{4.C.\Delta}) + \frac{11.\sigma}{12.C} - \frac{5.L}{12.C} \tag{5.4}$$

où T_{lc} est la durée maximale d'une opération de codage, τ est le temps de service d'un nœud (indépendamment du codage), C est l'unité de capacité des liens, L est la longueur des paquets, Δ est la durée d'un intervalle correspondant à une génération et σ est le coefficient représentant les bursts de données du flux d'entrée.

De manière prévisible, le codage réseau est meilleur quand le temps de codage est faible. Son l'intérêt grandit lorsque les paramètres τ et T augmentent. Enfin, on peut observer avec le paramètre σ que plus le trafic contient des bursts, plus le codage réseau se comporte bien.

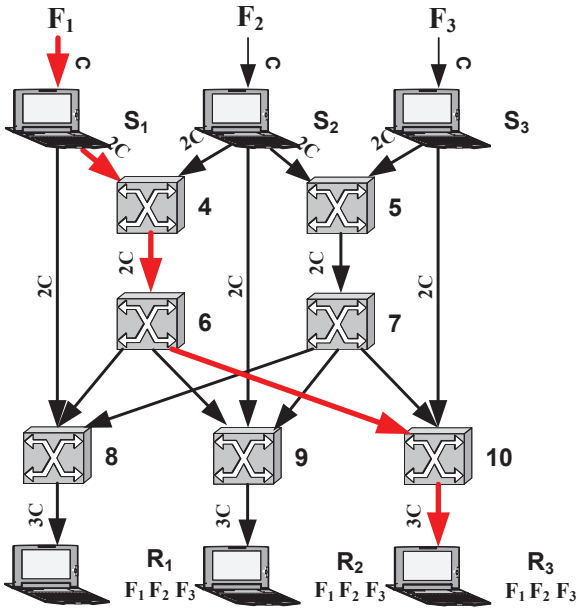


Fig. 5.10 – Exemple de réseau à qualité de service

5.3.6 Conclusion

L'objectif initial de ce travail consistait à évaluer la capacité du codage réseau à diminuer les bornes de temps de traversée des réseaux à qualité de service. Cette objectif a été atteint en définissant des stratégies de codage réseau pour ce type de réseau et en montrant comment évaluer les bornes de temps de traversée.

La principale suite de ce travail est l'analyse du comportement moyen de la stratégie FFS qui assure une fiabilité totale tout en étant capable de s'adapter à la variabilité des flux et de leur débit. Cette stratégie pourrait s'avérer intéressante pour les applications nécessitant une fiabilité totale dans des réseaux à topologie fixe.

5.4 Conclusions et Perspectives pour les applications du codage dans les réseaux

Le nombre de travaux consacrés aux applications du codage (et de la théorie de l'information) dans les grandes conférences liées au réseaux est en augmentation constante. Ces travaux qui, jusqu'à récemment, concernaient principalement des évaluations plus ou moins théoriques de mécanisme de codage, tendent de plus en plus vers des propositions d'intégration du codage dans des protocoles existants.

Même si l'utilisation du codage dans des applications réseaux à grande échelle n'est pas encore tout à fait d'actualité, ceci pourrait être le cas dans un futur proche dans plusieurs domaines tels que les réseaux pair-à-pair ou les réseaux sans fil maillés.

Chapitre 6

Conclusions et perspectives

Comme indiqué dans la conclusion du chapitre précédent, de plus en plus de travaux liés au codage sont publiés dans des conférences et revues liées aux réseaux. De la même manière, les applications du codage en réseau apparaissent dans les conférences de théorie de l'information.

Par rapport à ce constat, les sujets que nous développons nous semblent donc d'actualité. Il est même probable que les communautés du réseau et des communications/télécommunications devraient continuer à se rapprocher.

En effet, la recherche en réseau commence à atteindre un certain niveau de maturité et les améliorations possibles ne peuvent se faire que sur des bases théoriques solides. En ce sens, cette recherche devrait pouvoir tirer profit des outils et des compétences développées depuis 50 ans dans le cadre de la théorie de l'information. Les meilleurs exemples sont probablement les domaines du codage réseau et des codes à effacement, que l'on peut désormais considérer comme parties intégrantes à la fois de la théorie des réseaux et de la théorie de l'information.

De plus, le fait que les limites théoriques soient proches sur la couche physique poussent un certain nombre de chercheurs à orienter leur travaux vers des domaines plus "neufs". Il suffit d'observer l'engouement suscité par le codage réseau, qui a généré en quelques années un nombre considérable de publications.

Cette convergence d'intérêts de ces deux communautés coïncide avec le besoin désormais clairement identifié d'amélioration globale des architectures en couches protocolaires. Bien que la notion de couche protocolaire ne soit pas remise en cause (surtout en ces temps de convergence de technologies), les gains potentiels des approches multi-couches poussent les industriels et, par suite, les organismes de normalisation à intégrer ce type d'approches dans les architectures protocolaires de prochaine génération. Un bon exemple est le standard DVB-SH qui a été défini en tenant compte de la couche application, et qui intègre une véritable approche bi-couches entre la couche physique et la couche liaison.

Une des difficultés de ce thème de recherche est le grand nombre de notions différentes (protocoles, codages, systèmes) qu'il faut intégrer pour produire des résultats significatifs. Les collaborations avec des spécialistes des domaines connexes (réseau ou codage) sont d'ailleurs un élément de réussite essentiel dans ce domaine.

Le nombre d'applications potentielles sur les différentes couches est aussi un facteur qui peut entraîner une certaine dispersion de l'activité de recherche. Toutefois, tous ces sujets sont étroitement liés : le codage réseau et les codes à effacement sont de plus en plus liés, la conception ou le choix d'un code à effacement ne peut se faire sans une connaissance des couches basses utilisée, les couches basses commencent à envisager l'utilisation de codage

réseau... En l'état actuel des choses, il paraît donc difficile de focaliser toute son énergie sur un sujet particulier.

L'étendue des connaissances nécessaires est d'ailleurs un des points intéressants de ce domaine. Il est aussi important de ne pas perdre de vue qu'un des objectifs du métier d'enseignant-chercheur est de maintenir un lien constant entre la recherche et l'enseignement. Dans le cadre d'une école d'ingénieur généraliste, il apparaît que l'objectif d'enseignement se réalise d'autant plus facilement que le spectre des domaines de compétences est large.

Bibliographie

[1] ETSI, "Physical layer standard for cdma2000 spread spectrum systems," 3GPP2 C.S0002-D version 1, Tech. Rep., feb. 2004.

[2] C. E. Shannon, "A mathematical theory of communication." *Bell Systems Technical Journal*, vol. 27, pp. 379–423, 1948.

[3] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding : turbo-codes," *IEEE, Trans. Inform. Theory*, vol. 44, no. 10, pp. 1261–1271, 1996.

[4] R. G. Gallager, *Low-density parity-check codes.* MIT Press, 1963.

[5] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[6] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of Soc. Ind. Appl. Math*, vol. 8, pp. 300–304, 1960.

[7] ISO, "Information Processing Systems - Open Systems Interconnection - Basic Reference Model, American National Standards Institute, ISO/IEC 7498-1," 1984.

[8] P. Karn, C. Bormann, G. Fairhurst, D. Grossman, R. Ludwig, J. Mahdavi, G. Montenegro, J. Touch, and L. Wood, "Advice for Internet Subnetwork Designers," Internet Engineering Task Force, RFC 3819, Jul. 2004.

[9] J. Postel, "Internet Protocol," Internet Engineering Task Force, RFC 0791, Sep. 1981. [Online]. Available : http://www.rfc-editor.org/rfc/rfc791.txt

[10] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," Internet Engineering Task Force, RFC 2460, Dec. 1998. [Online]. Available : http://www.rfc-editor.org/rfc/rfc2460.txt

[11] J. Postel, "User Datagram Protocol," Internet Engineering Task Force, RFC 0768, Aug. 1980. [Online]. Available : http://www.rfc-editor.org/rfc/rfc768.txt

[12] L.-A. Larzon, M. Degermark, S. Pink, L.-E. Jonsson, and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)," RFC 3828 (Proposed Standard), Jul. 2004. [Online]. Available : http://www.ietf.org/rfc/rfc3828.txt

[13] J. Postel, "Transmission Control Protocol," Internet Engineering Task Force, RFC 0793, Sep. 1981. [Online]. Available : http://www.rfc-editor.org/rfc/rfc793.txt

[14] J. Nonnenmacher, E. W. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 349–361, Aug. 1998.

[15] J.-C. Bolot, S. Fosse-Parisis, and D. F. Towsley, "Adaptive FEC-based error control for internet telephony," in *INFOCOM*, 1999, pp. 1453–1460.

[16] S. Lin and D. Costello, *Error Control Coding : Fundamentals and Applications*. Prentice – Hall, Englewood- Cliffs, NJ, 1983.

[17] 3GPP, "Technical Specification Group Radio Access Network ; High Speed Downlink Packet Access (HSDPA), 3GPP TS 25.308-V8.0.0 ," 2007.

[18] IEEE, "IEEE Standard for Local and metropolitan area networks Part 16 : Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2 : Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1," *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (Amendment and Corrigendum to IEEE Std 802.16-2004)*, pp. 1–822, 2006.

[19] J. Cantillo, J. Lacan and M.-L. Boucheret and I. Buret, "Enhancing Delineation and Error Control with Header Redundancy : HERACLES," soumis à une revue en novembre 2008, 2008.

[20] J. Cantillo, J. Lacan, I. Buret and F. Arnal, "Procédé et module de correction d'erreurs de transmission dans un flux de données, système de communication comprenant ledit module," December 2007, french patent application FR0708623.

[21] ——, "Procédé et dispositif de délinéation d'un flux de données et système de communication comprenant ledit dispositif," 2008, french patent application FR0800968.

[22] J. Lacan and E. Delpeyroux, "The q-ary image of some q^m-ary cyclic codes : Permutation group and soft decision decoding algorithm," *IEEE Transactions on Information Theory*, vol. 48, pp. 2069–2078, 2002.

[23] M. L. Boucheret and J. Lacan, "Support etude modem : décodage souple de codes reed-solomon," TéSA/INPT/ISAE, Etude Rockwell-Collins, juillet 2008.

[24] Y. Oster, J. Lacan, and A. Duverdier, "Benchmark of reed-müller codes for short packet transmission," in *25th AIAA International Communications Satellite Systems Conference, Seoul, Corée*, avril 2007.

[25] ETSI, "Digital Video Broadcasting (DVB) ; Transmission System for Handheld Terminals (DVB-H) , ETSI EN 302 304," 2004.

[26] "Digital Video Broadcasting (DVB) ; Framing Structure, channel coding and modulation for Satellite Services to Handheld devices (SH) below 3 GHz," march 2007.

[27] J. Lacan and T. Pérennou, "Evaluation of Error Control Mechanisms for 802.11b Multicast Transmissions ," in *Second International Workshop On Wireless Network Measurement (WinMee 2006)*, April 2006.

[28] J. Lacan and T. Perennou, "Reducing Retransmissions in Point-to-point Satellite Transmissions," in *Proceddings of AIAA's* 25^{th} *International Communications Satellite Systems Conference, San Diego, California*, June 2008.

[29] A. Bouabdallah, M. Kieffer, J. Lacan, G. Sabeva, and P. Duhamel, "Rapport final : Etude cross-layer-application à sdmb," TéSA/ISAE, Tech. Rep., june 2006.

[30] A. Bouabdallah, M. Kieffer, J. Lacan, G. Sabeva, G. Scot, C. Bazile, and P. Duhamel, "Evaluation of Cross-Layer Reliability Mechanisms for Satellite Digital Multimedia Broadcast," *Broadcasting, IEEE Transactions on*, vol. 53, no. 1, pp. 391–404, March 2007.

[31] A. Bouabdallah, M. Kieffer, J. Lacan, G. Sabeva, and P. Duhamel, "Rapport final : Approfondissement cross-layer," TéSA/ISAE/Supelec, Etude CNES, mars 2007.

[32] J. Cantillo, J. Lacan and I. Buret, "Cross-layer enhancement of error control techniques for adaptation layers of DVB satellites," *International Journal of Satellite Communications and Networking*, vol. 24, pp. 579–590, 2006.

[33] DVB, "Generic Stream Encapsulation (GSE) Protocol," DVB BlueBook A116, May 2007.

[34] J. Cantillo, B. Collini-Nocker, U. De Bie, O. Del Rio, G.Fairhurst, A. Jahn and R. Rinaldo, "GSE : A Flexible, yet Efficient, Encapsulation for IP over DVB-S2 Continuous Generic Streams," *International Journal of Satellite Communications and Networking*, vol. To appear, 2008.

[35] J. Cantillo and J. Lacan, "A Design Rationale for Providing IP Services Over DVB-S2 Links," IETF draft, draft-cantillo-ipdvb-s2encaps-04.txt, expired, December 2007.

[36] L. Dairaine, L. Lancérica, J. Lacan, and J. Fimes, "Content-Access QoS in Peer-to-Peer Networks Using a Fast MDS Erasure Code," *Computer Communications*, vol. 28, no. 15, pp. 1778–1790, september 2005.

[37] J. Lacan and J. Fimes, "Systematic mds erasure codes based on vandermonde matrices," *IEEE Communications Letters*, vol. 8, pp. 570– 572, September 2004.

[38] J. Lacan, V. Roca, J. Peltotalo, and S. Peltotalo, *Reed-Solomon Forward Error Correction (FEC), draft-ietf-rmt-bb-fec-rs-05*, 2008, work in progress, draft-ietf-rmt-bb-fec-rs-01, Internet draft, in the RFC queue.

[39] T. Paila, M. Luby, R. Lehtonen, V. Roca, and R. Walsh, "FLUTE - File Delivery over Unidirectional Transport," Internet Engineering Task Force, RFC 3926, octobre 2004. [Online]. Available : http://www.rfc-editor.org/rfc/rfc3926.txt

[40] F. de Belleville, L. Dairaine, C. Fraboul, and J. Lacan., "Une approche hybride satellite/terrestre pour le transport fiable multipoint à grande échelle." in *Colloque Francophone sur l'Ingénierie des Protocoles*, 2003.

[41] F. de Belleville, L. Dairaine, J. Lacan, and C. Fraboul, "Reliable multicast transport by satellite : a hybrid satellite/terrestrial solution with erasure codes," in *IEEE conference on High Speed Networks and Multimedia Communications (HSNMC)*. Springer-Verlag, 2004.

[42] F. Arnal, L. Dairaine, J. Lacan, and G. Maral, "Cross-layer reliability management for multicast over satellite," *Computer Networks*, vol. 48, no. 1, pp. 29–43, May 2005.

[43] A. Bouabdallah and J. Lacan, "Dependency-Aware Unequal Erasure Protection Codes," in *Proceedings of the* 15^{th} *Packet Video Workshop, Hanghzou, China*, April 2006.

[44] J. Lacan and E. Lochin, "Rethinking reliability for long-delay networks," in *International Workshop on Satellite and Space Communications, 2008. IWSSC '08.*, octobre 2008.

[45] J. Lacan, L. Lancérica, and L. Dairaine, "When FEC speed up data access in p2p networks," in *IDMS'02 Conference (Interactive Distributed Multimedia Systems)*, 2002.

[46] J. Lacan and C. Michel, "Analysis of a circular code model," *Journal of Theoretical Biology*, vol. 213, pp. 159–170, 2001.

[47] D. G. Arquès, J. Lacan, and C. J. Michel, "Identification of Protein Coding Genes in Genomes with Statistical Functions Based on the Circular Code ," *Biosystems*, pp. 159–170, 2001.

[48] A. Soro, J. Lacan, E. Chaput, C. Baudoin, and F. Arnal, "Techniques de compression et satellite," TéSA/ISAE, Etude CNES, juin 2007.

[49] A. Soro, J. Lacan, E. Chaput, C. Donny, and C. Baudoin, "Evaluation of a generic unidirectional header compression protocol," in *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*, Sept. 2007, pp. 126–130.

[50] A. Soro, J. Lacan, E. Chaput, and C. Baudoin, "Header Compression Protocols Performance Modeling," soumis à une conférence en août 2009, disponible sur http ://oatao.univ-toulouse.fr/, 2008.

[51] J. Lacan, "Fault-tolerant distributed computing scheme based on erasure codes," in *NOTERE, Toulouse*, Juin 2006.

[52] J. Metzner, "An improved broadcast retransmission protocol," *Communications, IEEE Transactions on [legacy, pre - 1988]*, vol. 32, no. 6, pp. 679–683, Jun 1984.

[53] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *Journal of the ACM*, vol. 36, no. 2, pp. 335–348, 1989.

[54] V. Roca, C. Neumann, and D. Furodet, "Low Density Parity Check (LDPC) Staircase and Triangle Forward Error Correction (FEC) Schemes," Internet Engineering Task Force, RFC 5170, juin 2008.

[55] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "Raptor Forward Error Correction Scheme for Object Delivery," RFC 5053 (Proposed Standard), Oct. 2007. [Online]. Available : http://www.ietf.org/rfc/rfc5053.txt

[56] R.Lidl and H.Niederreiter, *Finite Fields*. Reading, MA : Addison-Wesley,, 1983.

[57] L. Rizzo, "Effective Erasure Codes For Reliable Computer Communication Protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, April 1997.

[58] F. J. McWilliams and N. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands : North Holland,, 1977.

[59] C. Neumann and V. Roca, "Analysis of fec codes for partially reliable media broadcasting schemes," in *2nd International Workshop on Multimedia Interactive Protocols and Systems (MIPS'04), Grenoble, France (BEST YOUNG RESEARCHER PAPER AWARD)*, Nov. 2004.

[60] Nokia, *Simulation results for the performance and complexity of RS Codes for MBMS FEC*, april 2005, 3GPP SA4 PSM 7 meeting Tdoc S4-AHP221, Sophia Antipolis, France.

[61] J. Bloemer, M. Kalfane, M. Karpinski, R. Karp, M. Luby, and D. Zuckerman, "An XOR-Based Erasure-Resilient Coding Scheme," in *Technical Report ICSI TR-95-048*, August 1995.

[62] I. Gohberg and V. Olshevsky, "Fast algorithms with preprocessing for matrix-vector multiplication problems," *Journal of Complexity*, vol. 10, no. 4, pp. 411–427, Dec. 1994.

[63] D. G. Cantor, "On arithmetical algorithms over finite fields," *J. Comb. Theory Series A*, vol. 50, pp. 285–300, 1989.

[64] A.-V. T. W. Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP : A Transport Protocol for Real-Time Applications," RFC 1889 (Proposed Standard), Jan. 1996, obsoleted by RFC 3550. [Online]. Available : http://www.ietf.org/rfc/rfc1889.txt

[65] P. Amer, C. Chassot, T. Connolly, M. Diaz, and P. Conrad, "Partial-order transport service for multimedia and other applications," *IEEE/ACM Trans. Netw.*, vol. 2, no. 5, pp. 440–456, 1994.

[66] C. Leicher, "Hierarchical encoding of MPEG sequences using priority encoding transmission (PET)," International Computer Science Institute, Berkeley, CA, Tech. Rep. TR-94-058, Nov. 1994.

[67] A. Bouadallah and J. Lacan, "Dependency-Aware Unequal Erasure Protection Codes," *Journal of Zhejiang University - Science A*, vol. 7, pp. 27–33, 2006.

[68] C. Hellge, T. Schierl, and T. Wiegand, "Multidimensional layered forward error correction using rateless codes," *Communications, 2008. ICC '08. IEEE International Conference on*, pp. 480–484, May 2008.

[69] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[70] J. Lacan and E. Lochin, "On-the-Fly Coding to Enable Full Reliability Without Retransmission," soumis à une conférence en août 2009, disponible sur http ://arxiv.org/, 2008.

[71] J. Kumar Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pp. 1651–1655, July 2008.

[72] J. Kumar Sundararajan, D. Shah, M. Médard, M. M., and B. J., "Network Coding meets TCP," septembre 2008, disponible sur ArXiv.

[73] C. Bormann and *al.*, "RObust Header Compression (ROHC) : Framework and four profiles : RTP, UDP, ESP, and uncompressed ," RFC 3095 (Proposed Standard), Jul. 2001, updated by RFC 3759. [Online]. Available : http://www.ietf.org/rfc/rfc3095.txt

[74] 3GPP, "Radio Link Control (RLC) Protocol Specification , 3G TS RAN 25.322," 1999.

[75] J. Romkey, "Nonstandard for transmission of IP datagrams over serial lines : SLIP," Internet Engineering Task Force, RFC 1055, Jun. 1988. [Online]. Available : http://www.rfc-editor.org/rfc/rfc1055.txt

[76] 3GPP, "Mandatory speech codec ; AMR speech codec ; Interface to Iu and Uu, 3GPP TS 26.102 V 3.3.0," 2001.

[77] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel, "Robust mac-lite and header recovery based improved permeable protocol layer scheme," *Spread Spectrum Techniques and Applications, 2008. ISSSTA '08. IEEE 10th International Symposium on*, pp. 496–501, Aug. 2008.

[78] DVB, "DVB-SH Implementation Guidelines," DVB BlueBook A120, May 2008.

[79] F. Pérez Fontán, M. Vázquez-Castro, C. E. Cabado, J. P. García, and E. Kubista, "Statistical modelling of the lms channel," *IEEE Transactions On Vehicular Technology*, vol. 50, no. 6, pp. 1549–1567, november 2001.

[80] C. Loo, "A statistical model for land mobile satellite link," *IEEE Trans. Veh. Technol.*, vol. 34, pp. 122–127, august 1985.

[81] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *Information Theory, IEEE Transactions on*, vol. 42, no. 6, pp. 1737–1744, Nov 1996.

[82] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu and M. Weinberger, "Universal Discrete Denoising : Known Channel," *IEEE Transactions on Information Theory*, vol. 51, 2005.

[83] J. Cantillo, "Codage multi-couches pour systèmes de communication par satellites," Ph.D. dissertation, ENST, Toulouse, Mai 2008.

[84] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flows," *IEEE Transactions on Information Theory*, 2000.

[85] T. Pérennou, J. Lacan, and H. Elnatour, "Evaluation de Mécanismes de Contrôle d'Erreur pour des Transmissions Multipoints sur des Réseaux de Mobiles," *Technique et Science Informatiques (TSI) 24/2005*, pp. 865–886, 2005.

[86] J. Lacan and T. Pérennou, "Amélioration de la fiabilité des transmissions point-à-point sur un canal à diffusion," in *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'05)*, mars 2005.

[87] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air : practical wireless network coding." in *SIGCOMM*, 2006, pp. 243–254.

[88] E. Rozner, A. P. Iyer, Y. Mehta, L. Qiu, and M. Jafry, "ER : Efficient Retransmission Scheme for Wireless LANs," in *International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, December 2007.

[89] H. Weatherspoon and J. D. Kubiatowicz, "Erasure coding vs. replication : A quantitative comparison," *Lecture Notes in Computer Science*, vol. 2429, pp. 328–338, 2002.

[90] S. Acedanski, S. Deb, M. Médard, and R. Koetter, "How good is random linear coding based distributed networked storage," in *In NetCod*, 2005.

[91] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *IEEE/INFOCOM, Miami*, March 2005.

[92] L. Lancérica, L. Dairaine, and J. Lacan, "Evaluation of content-acces qos for various dissemination strategies in peer to peer networks," in *11th IEEE International Conference on Networks ICON*, 2003.

[93] A. Mahmino, J. Lacan, and C. Fraboul, "Calculus of service guarantees for network coding," in *Proc. ISITA 2006*, October 2006.

[94] ——, "Enhancing guaranteed delays with network coding," in *International Conferences on Networking - Networking 2007*, 2007, pp. 1229–1232.

[95] ——, "Guaranteed packet delays with network coding," in *First IEEE International Workshop on Wireless Network Coding : WiNC 2008*, June 2008, pp. 1–6.

[96] J.-Y. Le Boudec and P. Thiram, *Network Calculus A Theory of Deterministic Queuing Systems for the Internet*, ser. Series : Lecture Notes in Computer Science, Vol. 2050. Springer Verlag, 2001.

[97] R. L. Cruz, "A calculus for network delay, part i : Network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–131, January 1991.

[98] ——, "A calculus for network delay, part ii : Network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 132 – 141, January 1991.

[99] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *41st Allerton Conf. Communication, Control and Computing*, Oct. 2003.

[100] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, 2003.

[101] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *41st Allerton Conf. Communication, Control and Computing*, Oct. 2003.