

# Étude du relayage de codes fontaine sur un lien de transmission multi-sauts

SÉBASTIEN BLANCHARD, SOFIANE RERBAL, ANYA APAVATJRUT, CLAIRE GOURSAUD, JEAN-MARIE GORCE

Laboratoire CITI

INSA Lyon, Bâtiment Claude Chappe, 6 avenue des Arts, 69621 Villeurbanne Cedex, France

sebastien.blanchard@insa-lyon.fr, sofiane.rerbal@insa-lyon.fr,  
anya.apavatjrut@insa-lyon.fr, claire.goursaud@insa-lyon.fr, jean-marie.gorce@insa-lyon.fr

**Résumé** – Dans cet article, nous considérons le relayage de paquets avec codage fontaine sur un lien de transmission multi-saut. Nous cherchons à exploiter les propriétés de codes fontaines afin d’améliorer l’efficacité des relais et de réduire la quantité de paquets émis sur le lien. Nous proposons un nouvel algorithme simple de recréation de paquets qui vérifie au mieux la loi de Soliton. Nous montrons que l’utilisation de cet algorithme dans les relais permet de réduire non seulement la quantité de paquets émis par la source, mais aussi le nombre global de paquets émis sur le lien source-destination sans augmenter le temps de transmission.

**Abstract** – This article proposes a new relaying scheme using rateless property of digital fountain code in multi-hop communication. The proposed algorithm consists in generating encoded packet at each relaying node. These encoded packets have their encoding degrees conform to Soliton distribution. We show that our algorithm permits to reduce not only the quantity of source emitted packet, but also the overall link emitted packet, without increasing the transmission delay.

## 1 Introduction

Important sujet de recherche dans les télécommunications contemporaines, les réseaux de capteurs sont des réseaux sans fils constitués de plusieurs nœuds pouvant communiquer entre eux. Afin d’optimiser la consommation électrique globale du réseau, les nœuds coopèrent en prenant la fonction de relais pour les nœuds plus lointains. Nous considérons dans cet article le cas d’une source transmettant des données au travers d’un ou plusieurs relais sur des canaux à effacement, jusqu’à son destinataire. Cependant, la transmission se faisant en espace libre, le signal émis est sujet à des évanouissements, et certains paquets de données envoyés sont perdus entre les relais. Dans le cas de protocoles classiques, les paquets perdus sont réémis, entraînant une latence supplémentaire dans la réception des données.

Dans une topologie multi-sauts, cela devient problématique pour un grand nombre de sauts. Une des solutions proposées est l’utilisation des codes fontaines [1][2][3][4] qui permet d’alléger le système de contrôle des transmissions tout en assurant un lien fiable vers notre puits, réduisant ainsi ces temps de latence. En effet, les codes fontaines introduisent une redondance aléatoire de l’information, et le récepteur peut décoder à partir du moment où il a reçu un nombre suffisant de paquets codés [1].

En théorie, si on est capable de décoder toute l’information au niveau d’un relais, il est possible de la ré-encoder puis de l’envoyer au relais suivant annulant ainsi la dégradation du flux. Cependant cette méthode entraîne, outre une consommation mémoire importante, une latence évidente puisque chaque relais doit recevoir la totalité des paquets avant de pouvoir ré-encoder l’information pour la transmettre au relais suivant.

Il est donc nécessaire de trouver une procédure simple et peu coûteuse qui permet un remplacement des paquets perdus afin de garantir une utilisation optimale du lien suivant.

Cette voie a déjà été explorée par les « greedy code » [6] mais ceux-ci imposent une complexité de décodage au niveau du receveur de  $O(k^2 \ln k)$  -  $k$  étant le nombre de symboles encodés - alors que dans notre cas, nous respectons la procédure de décodage d’un code LT de complexité  $O(k \ln k)$ .

Nous s’intéressons dans cet article à optimiser la performance de transmission multi-sauts à l’aide des propriétés de codage fontaine. Nous commençons par une étude préliminaire de code fontaines ainsi que leur performance dans plusieurs scénarios de transmission. Ensuite, nous proposons un algorithme de reconstruction de paquets qui agit à chaque relais sur des paquets reçus codés qui tente de remplacer les paquets corrompus par des combinaisons de paquets correctement reçus. Cette méthode est basée sur la distribution « Robust Soliton » qui a été proposée pour le codage fontaine dans [5]. Enfin, nous montrons que notre algorithme assure une performance plus rapide et efficace que le relayage sans code fontaine.

## 2 Codage de type fontaine

Dans une transmission avec codage fontaine, les données à transmettre sont partitionnées en plusieurs fragments de données, qui sont combinés entre eux pour créer un nouveau paquet qui sera ensuite transmis dans le réseau. Le nombre de fragments combinés dans un paquet est une variable aléatoire  $d$  qui dépend de la loi de Robust Soliton qui a été définie dans [5].

La distribution de probabilité Robust soliton,  $\mu$ , est définie à l'aide de  $K$ , nombre de paquets à envoyer et de deux fonctions  $\rho$  et  $\tau$  :

$$\begin{cases} \rho(1) = \frac{1}{K} \\ \rho(d) = \frac{1}{d(d-1)} \end{cases} \quad (1)$$

Avec

$$\tau(d) = \begin{cases} \frac{s}{Kd} \text{ pour } d = 1, 2, \dots, \frac{K}{s} - 1 \\ \frac{s}{K} \log\left(\frac{K}{\delta}\right) \text{ pour } d = \frac{K}{s} \\ 0 \text{ sinon} \end{cases} \quad (2)$$

Donnant :

$$\mu(d) = \frac{\rho(d) + \tau(d)}{Z} \quad (3)$$

Où

$$Z = \sum_{d'=1}^K \rho(d') + \tau(d') \quad (4)$$

Cette distribution de probabilité est utilisée par l'encodeur de code fontaine pour déterminer le nombre de paquets qui seront « xor-és » dans chaque paquet envoyé.

Le décodage utilise un algorithme de propagation de croyance ou « Belief Propagation » décrit en [2,5].

### 3 Modélisation du lien multisauts étudié

Nous proposons d'étudier la performance de notre algorithme dans le cas d'un réseau linéaire, caractérisé par une transmission multi-saut selon la figure 1.



Figure 1 : Topologie du réseau étudié : 1 source qui envoie des données au destinataire à travers 1 ou plusieurs relais.

La source envoie l'information à la destination en passant par un certain nombre de nœuds relais. La qualité du canal de transmission est modélisée par le taux d'erreur par bit ou Bit Error Rate (BER). Pendant la transmission, les paquets contenant au moins un bit erroné sont considérés comme corrompus et seront rejetés. Un mécanisme de redondance doit être mis en œuvre pour garantir la transmission de tous les fragments. Nous supposons que l'acquittement de paquets est un processus sans collision et sans perte.

Nous évaluons dans cette étude les performances du système en fonction du nombre de paquets à transmettre pour garantir une bonne transmission de tous les fragments.

### 4 Fonctionnement avec relais

Nous proposons quatre scénarios d'étude.

#### 4.1 Sans codage

Le premier scénario correspond à une transmission multi-sauts sans codage. Dans ce cas, chaque relais

retransmet directement tout paquet reçu. Un processus d'acquittement de bout en bout est mis en œuvre et impose la retransmission par la source de tout paquet perdu en cours de route.

#### 4.2 Codage Fontaine sans reconstruction

Le deuxième scénario est identique au précédent, sauf que la source utilise un code fontaine. Ainsi, il n'est pas nécessaire d'acquitter chaque paquet, et la source transmet des paquets codés jusqu'à ce que le destinataire ait reçu suffisamment de paquets pour pouvoir reconstruire tous les fragments.

#### 4.3 Codage fontaine avec décodage complet

Dans le troisième scénario, chaque relais décode les paquets reçus, reconstruit tous les segments, et ré-encode les fragments pour les transmettre au nœud suivant. Cette approche introduit une forte latence car chaque relais doit attendre d'avoir reçu tous les fragments pour commencer sa retransmission.

Cependant, ce cas peut être défini comme une borne minimale car il optimise le nombre de paquets nécessaire à envoyer. Cependant, ce cas introduit une grande latence de bout en bout.

#### 4.4 Codage fontaine avec reconstruction

Enfin, le dernier cas consiste à appliquer l'algorithme de reconstruction au niveau des nœuds intermédiaires. Lorsqu'un paquet est perdu, le relais suivant reconstruit un paquet à partir des paquets précédemment reçus et mémorisés dans la mémoire. L'algorithme de reconstruction est défini ainsi :

*POUR* Chaque paquet reçu

(i) SI paquet corrompu ALORS

(i.1) Effacer le paquet

(i.2) Tirer degré\_demandé dans la distribution  
« Robust Soliton »

(i.3) TANT QUE degré (paquet) < degré\_demandé

POUR CHAQUE paquet2 dans le buffer

SI degré (paquet2 XOR paquet)

<= degré\_demandé

paquet = paquet XOR paquet2

FIN SI

FIN POUR CHAQUE

FIN TANT QUE

FIN SI

(ii) Transmettre paquet et attendre paquet suivant

Cet algorithme a pour but d'augmenter la redondance dans le lien suivant en envoyant des combinaisons de type XOR entre les paquets à transmettre, mais en faisant en sorte que les degrés de distribution des paquets créés vérifient la loi de Robust Soliton.

### 5 Résultats

Nous nous intéresserons dans cet article, au nombre de paquets de données transmis, pour évaluer l'apport de la reconstruction de paquets face aux codes fontaines

classiques et face à un système de transmission sans codage.

Nous avons donc considéré le cas de  $K=100$  paquets à envoyer. Le degré des paquets codés suit une loi de distribution de Soliton de paramètres  $c=0.125$  et  $\delta=0.5$  selon scénario 1. Il est à noter que nous avons déterminé qu'un buffer de 10 paquets était optimal pour l'obtention de nos résultats, le gain offert par une augmentation de buffer étant limité.

### 5.1 Evaluation de performance : nombre de paquets émis par la source

Nous évaluons le nombre moyen de paquets nécessaires à envoyer à partir de la source pour décoder l'information au niveau du destinataire, ainsi que les ressources moyennes nécessaires à cette transmission. Le nombre de paquets à envoyer au niveau de la source est représentatif du temps de transmission global de notre information.

Nous avons déterminé théoriquement que le nombre moyen de paquet à envoyer par la source pour assurer la bonne transmission respectivement dans le cas sans codage ( $NB_1$ ), avec codage fontaine sans reconstruction ( $NB_2$ ) et avec codage fontaine avec décodage complète ( $NB_3$ ) peut être modélisé par ces équations :

$$NB_1 = \frac{K}{(1 - BER)^{L \cdot N}} \quad (5)$$

$$NB_2 = \frac{K + \epsilon}{(1 - BER)^{L \cdot N}} \quad (6)$$

$$NB_3 = \frac{K + \epsilon}{(1 - BER)^L} \quad (7)$$

Où  $L$  est la longueur de fragment envoyé et  $N$  le nombre de sauts.

Nous observons dans la fig.2 les résultats obtenus pour les quatre scénarios d'études en simulation et en théorie. Nous constatons que notre algorithme améliore la robustesse des codes fontaines face à une perte de paquets importante. Ceci s'explique par une meilleure utilisation de chaque lien entre relais grâce à la reconstruction de paquets. En effet, l'algorithme de recréation de paquet permet d'augmenter la probabilité que l'information arrivée au nœud courant soit transmise au nœud suivant, et que sans savoir quels paquets ont été perdus.

Ainsi l'écart entre le code classique et ce codage avec reconstruction est d'autant plus important que le nombre de paquets perdus et reconstruits est grand car les slots temporels libérés par les paquets perdus sur un lien sont utilisés pour fiabiliser le lien suivant. Les codes fontaines étant, de par leur nature, plus gourmands en nombre de paquets à envoyer, il est normal de voir l'intérêt de l'algorithme moindre dans les cas où le canal de transmission est de bonne qualité. Cependant, nous avons pu combiner, dans le cas de fortes perturbations, une robustesse face au nombre de relais ainsi qu'une performance meilleure que les systèmes actuels dans des conditions plus « extrêmes ».

On remarque également sur la fig.3 que notre algorithme devient très intéressant dans un environnement avec beaucoup de relais.

En revanche, on pourrait penser que le nombre de paquets nécessaires pour décoder l'information est constant quelque soit le nombre de sauts. Or cela n'est pas le cas. Cela s'explique par le fait que nos paquets reconstruits ne respectent pas exactement une distribution de code fontaines et surtout par le fait que le nombre de paquets disponibles au moment de la reconstruction est limité. Nous introduisons donc une redondance mais ne maîtrisons pas une diversité d'information suffisante pour garantir un nombre de paquets nécessaires à envoyer constant. La solution qui s'en approcherait le plus serait d'attendre la transmission complète des paquets à chaque relais pour décoder selon scénario 3. Or cette méthode introduit une forte latence à chaque relais et demande une puissance de calcul importante à chaque étape de la transmission, elle n'est donc pas souhaitable dans la plupart des cas (dont le nôtre).

De cette première étude, on peut affirmer que notre algorithme est plus performant que les codes fontaines classiques et deviennent plus intéressants en terme d'activité de la source qu'une transmission sans codage à partir d'une certaine dégradation du canal de transmission.

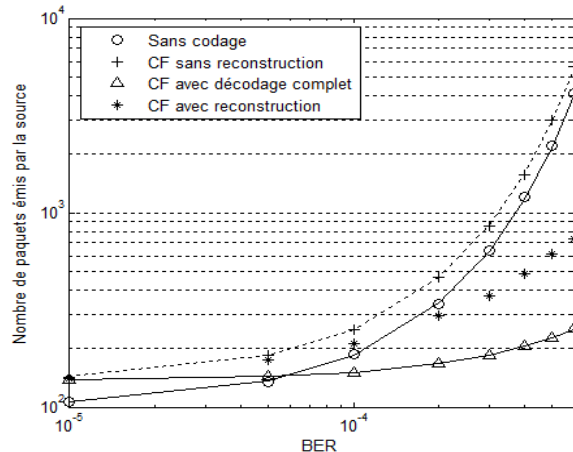


Figure 2 : Nombre moyen de paquets à envoyer pour assurer le décodage ( $K=100$ , 5 relais) suivant le BER

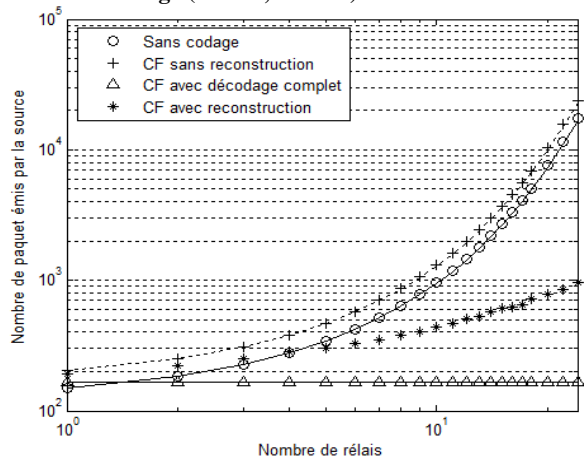


Figure 3 : Nombre moyen de paquets à envoyer pour assurer le décodage ( $K=100$ ,  $BER = 2 \times 10^{-3}$ ) suivant le nombre de relais

## 5.2 L'évaluation de performance : nombre total de sauts

Tout système de transmission radio est limité par l'énergie de ses émetteurs ainsi que par l'occupation de son médium de transmission. Dans le cadre des réseaux de capteurs, la puissance disponible est faible et les ressources radios (paquets pouvant être transmis simultanément par deux relais distants) sont limitées. Afin d'évaluer les performances de notre algorithme dans ce contexte, nous nous intéressons au nombre de sauts effectués pour une transmission complète de 100 fragments ( $K=100$ ). Ce nombre de sauts correspond au nombre de transmissions point-à-point qui se sont produites durant le processus et donc ceci est une indication de la puissance consommée. Cet aspect est intéressant à étudier car dans le cas des transmissions classiques, chaque paquet perdu à un relais n'est pas retransmis et ne réutilise donc pas de ressources radio alors que dans le cas de notre algorithme, chaque paquet perdu est reconstruit puis retransmis. Ainsi, il est intéressant d'évaluer si l'apport du codage justifie la transmission de paquets supplémentaires.

Nous constatons dans la fig.4 que notre algorithme offre une meilleure utilisation des ressources que le code classique. L'impact de la transmission de paquets reconstruits est donc plus faible que le gain qu'il apporte au niveau du nombre de paquets nécessaires à envoyer.

Les codes fontaines utilisent plus de ressources radio qu'un système classique pour de bonnes conditions de transmission mais il est clair que sa robustesse aux fortes perturbations est bien plus grande. Combiné avec notre algorithme, les codes fontaines permettent d'améliorer considérablement les ressources nécessaires à une transmission dans de mauvaises conditions.

Il est légitime de penser que l'écart se creuse lorsque le nombre de paquets reconstruits devient plus important. Nous avons donc augmenté le nombre de relais afin de mettre à l'épreuve, selon la fig.5, nos premières constatations.

Ici encore le gain semble quasi-linéaire. Une fois de plus on constate une performance croissante en fonction du nombre de paquets à reconstruire. Il est intéressant de noter que quelques soient les paramètres de simulation, notre algorithme est soit équivalent, soit meilleur qu'un code fontaine classique.

## 6 Conclusions

Nous avons vu au cours de notre étude que les performances de notre algorithme en terme de vitesse de transmission et de ressources radio utilisées sont supérieures au codes fontaines classiques. De plus la suppression d'un canal retour d'acquittement pour chaque paquet est un atout non négligeable car cela simplifie grandement le design de chaque élément de la chaîne de transmission.

Pour le moment, ces résultats sont limités du cas d'une transmission multi-sauts qui utilise un seul chemin de transmission entre une source et un destinataire.

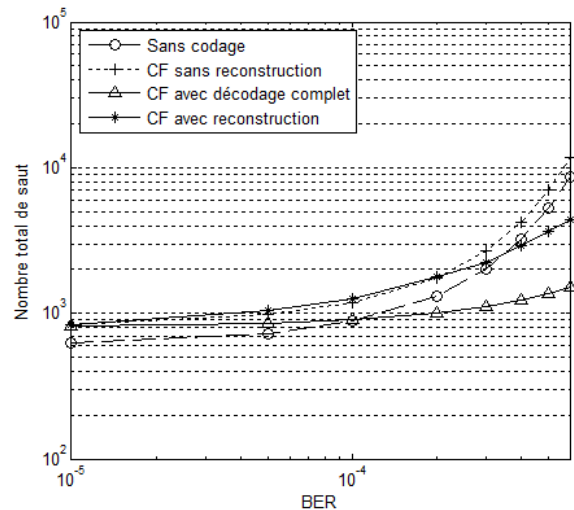


Figure 4 : Nombre moyen de sauts nécessaires pour la transmission ( $K=100$  suivant le BER).

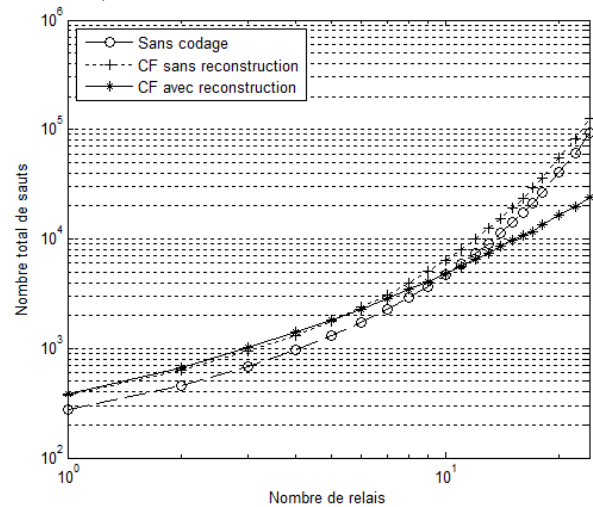


Figure 5 : Nombre moyen de paquets à envoyer pour assurer le décodage ( $K=100$ ,  $BER = 2 \times 10^{-3}$ ) suivant le nombre de relais.

Le principal intérêt de cette étude est d'avoir considéré un acquittement uniquement global, et de bout en bout. Cette étude peut donc facilement être étendue aux réseaux basés sur le relayage opportuniste, ce qui fera l'objet de nos futurs travaux.

## 7 Références

- [1] M. Mitzenmacher, « Digital Fountains: A Survey and Look Forward », Information Theory Workshop, 2004, IEEE, pp. 271- 276
- [2] M. Luby, « LT codes », Proc. Of the 43rd Annual IEEE Symposium on the Foundations of Computer Sciences (STOC), pp. 270-280, 2002.
- [3] S. Puducheri, J.Kliwer, T.E.Fuja , « The Design and Performance of Distributed LT Codes », IEEE Transactions on Information Theory, Vol.53, pp.3740-3754
- [4] S.Puducheri, J.Kliwer, T.E. Fuja, « Distributed LT Codes », IEEE International Symposium on Information Theory 2006, pp.987-991
- [5] D. Mackay, « Information Theory, Interference and Learning Algorithms », Cambridge University Press
- [6] P. Pakzad, C. Fragouli, A. Shokrollahi, « Coding Schemes for Line Networks », IEEE International Symposium on Information Theory (ISIT'05), 2005