



HAL
open science

Application du Codage Réseau aux Architectures à Garanties de Qualité de Service (QoS)

Ali Mahmino

► **To cite this version:**

Ali Mahmino. Application du Codage Réseau aux Architectures à Garanties de Qualité de Service (QoS). Informatique et langage [cs.CL]. Institut National Polytechnique de Toulouse - INPT, 2009. Français. NNT : 2009INPT023H . tel-04412471

HAL Id: tel-04412471

<https://theses.hal.science/tel-04412471>

Submitted on 23 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *Institut National Polytechnique de Toulouse (INPT)*

Discipline ou spécialité : *Informatique, Réseaux*

Présentée et soutenue par *Ali Mahmino*

Le 12.11.2009

Titre :

Application du Codage Réseau aux Architectures à Garanties de Qualité de Service (QoS)

JURY

| | |
|--------------------------|--|
| <i>Muriel Médard</i> | <i>Professeur d'Université, MIT, Etats Unis</i> |
| <i>Chadi Barakat</i> | <i>HDR, Chargé de Recherche, INRIA, Sophia Antipolis, France</i> |
| <i>Kavé Salamatian</i> | <i>Professeur à l'Université de Lancaster, Angleterre</i> |
| <i>Francès Fabrice</i> | <i>Enseignant, chercheur à l'ISAE, France</i> |
| <i>Jérôme Lacan</i> | <i>Professeur ISAE, France</i> |
| <i>Christian Fraboul</i> | <i>Professeur INPT, France</i> |

Ecole doctorale : *Mathématiques, Informatique et télécommunications de Toulouse (M.I.T.T.)*

Unité de recherche : *DMIA/ISAE*

Directeur(s) de Thèse : *Jérôme LACAN*
Christian FRABOUL

Remerciements

Je tiens tout d'abord à remercier mes directeurs de thèse, M. le Professeur **Jérôme Lacan**, pour m'avoir guidé, encouragé, supporté et pour la gentillesse, la patience et toute l'attention qu'il a manifesté à mon égard durant cette thèse, et pour sa disponibilité et sa confiance et ses précieux conseils et M. le Professeur **Christian Fraboul**, pour m'avoir encadré et dirigé par ses conseils avisés et suggestions qui ont permis l'accomplissement de ce travail. Je ne sais comment exprimer ma gratitude à ces deux personnes autrement qu'en leur adressant ici les marques de ma reconnaissance et de mon profond respect.

Je remercie M. **Patrick Sénac**, chef du Département de Mathématiques, Informatique et Automatique (DMIA) à l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE) à Toulouse, France, pour m'avoir accueilli au sein de son département et ses groupes.

Je suis très sensible à l'honneur que m'a fait M. **Kavé Salamatian**, professeur à l'université de Lancaster en Angleterre, en acceptant de présider le jury de cette thèse. Je lui exprime toute ma reconnaissance pour l'intérêt porté à ce travail.

J'exprime toute ma gratitude à Mme **Muriel Médard**, professeur au département de génie électrique et informatique et au laboratoire de recherches d'électronique à l'Institut de Technologie du Massachusetts (MIT) à Cambridge, Massachusetts, États-Unis, et M. **Chadi Barakat**, HDR, chargé de recherche, INRIA, Sophia-Antipolis, France, pour l'honneur qu'ils m'ont fait en acceptant d'être les rapporteurs de cette thèse. Ils ont également contribué par leurs nombreuses remarques et suggestions à améliorer la qualité de ce mémoire, et je leur en suis très reconnaissant.

J'adresse ma profonde reconnaissance à Monsieur **Fabrice Francès**, enseignant, chercheur au Département de Mathématiques, Informatique et Automatique (DMIA) à l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE) à Toulouse, France, pour l'honneur qu'il me fait en acceptant de bien vouloir participer à ce jury de soutenance

Je tiens aussi à mentionner le plaisir que j'ai eu à travailler au sein du DMIA, et j'en remercie ici tous les membres qui sans eux, mes conditions de travail auraient sans doute été très différentes et beaucoup moins agréables. Je pense ici en particulier à Mlle. et MM. *A. Mifdaoui, E. Lochin, F. Francès, P. de Saqui-Sannes, T. Pérennou, Y. Caumel, B. Jarlan.*

Je passe ensuite une dédicace spéciale à tous les jeunes gens avec qui j'ai eu le plaisir de travailler et de côtoyer durant ces quelques années à Toulouse, à savoir *Emmanuel, Hervé, Mathieu, Amine, Tarek, Alex, Guodong, Lei, Thomas, Pierre, . . .*

Je remercie chaudement mes parents mes frères et sœurs et le reste de la famille pour leurs encouragements et leur assistance aussi bien matérielle que morale qui m'ont permis de faire cette thèse dans de bonnes conditions.

J'associe à mes remerciements l'Université d'Alep et le laboratoire de Télécommunications Spatiales et Aéronautiques (TéSA- Toulouse) pour leur soutien financier.

Remerciements

Enfin, je voudrais remercier mes amis *Rami* et *Manel*, *A. Kazem* et *Samar*, *Maya*¹, *Sam*, *Hakima*, *Houda*, *Samir* et les autres pour leur aide sympathique tant sur le plan, scientifique qu'humain et pour la bonne humeur dans laquelle ce travail a été accompli.

Pour finir, Je tiens également à exprimer toute ma gratitude envers Mlle. la Dr. *H. Azira* et M. *A. Bouabdallah* pour leur aide et leur soutien et pour leurs relectures avisées du manuscrit.

Ali MAHMINO

Novembre 2009

Département Mathématiques, Informatique, Automatique (DMIA)

Institut supérieur de l'aéronautique et de l'espace (ISAE), Toulouse, France

1. Merci du fond du cœur pour tout

Table des matières

| | |
|--|----------|
| Résumé | 1 |
| 1 Introduction | 3 |
| 1.1 Présentation de la problématique et des objectifs | 4 |
| 1.2 Organisation du document | 6 |
| 2 État de l'art et notions de base | 9 |
| 2.1 Introduction | 10 |
| 2.2 Codage réseau | 10 |
| 2.2.1 Introduction du codage réseau | 10 |
| 2.2.2 Codage réseau et théorie des graphes | 13 |
| 2.2.3 Les principales notions théoriques du codage réseau | 15 |
| 2.2.3.1 Théorie de Min-cut Max-flux | 15 |
| 2.2.3.2 Codage réseau linéaire multicast | 15 |
| 2.2.3.3 Codage réseau linéaire aléatoire | 17 |
| 2.2.3.4 Construction algébrique du codage réseau | 20 |
| 2.3 Bénéfices et intérêts de codage réseau | 28 |
| 2.3.1 Augmentation de la capacité multicast et du débit | 29 |
| 2.3.2 Diminution de l'énergie par bit | 29 |
| 2.3.3 Robustesse et tolérance aux fautes | 31 |
| 2.4 Applications de codage réseau dans différents domaines et types de réseaux | 32 |
| 2.4.1 Internet | 32 |
| 2.4.1.1 Modèle de paquet | 33 |
| 2.4.1.2 Modèle de file d'attente | 35 |
| 2.4.2 Réseaux de recouvrement "Overlay" | 36 |
| 2.4.3 Réseaux ad-hoc et sans fil | 37 |

Table des matières

| | | |
|----------|--|-----------|
| 3 | Codage réseau et réseaux avec des garanties de qualité de service (QoS) | 39 |
| 3.1 | Introduction | 40 |
| 3.2 | Réseaux avec des garanties de qualité de service (QoS) | 40 |
| 3.2.1 | Concepts et exemples de réseaux avec des garanties de QoS | 40 |
| 3.2.1.1 | Internet et Qualité de Service | 40 |
| 3.2.1.2 | ATM et Qualité de Service | 43 |
| 3.2.1.3 | Les réseaux embarqués | 45 |
| 3.3 | Besoins et applications du codage réseau au sein des réseaux avec QoS garantie | 49 |
| 3.3.1 | Intérêts du codage réseau pour les réseaux avec QoS garantie | 49 |
| 3.3.2 | Conditions d'application du codage réseau | 51 |
| 3.4 | Le calcul réseau | 52 |
| 3.4.1 | Intérêts du calcul réseau pour notre problématique | 53 |
| 3.4.2 | Les notions de base du calcul réseau | 53 |
| 4 | Nouvelles stratégies de codage réseau | 59 |
| 4.1 | Introduction et hypothèses | 60 |
| 4.2 | Stratégie Orientée Réseau (NOS : Network-Oriented Strategy) | 62 |
| 4.2.1 | Introduction | 62 |
| 4.2.2 | Définitions et hypothèses | 62 |
| 4.2.3 | Architecture d'un nœud de codage | 63 |
| 4.2.4 | NOS au niveau d'un nœud de codage | 64 |
| 4.2.5 | NOS au niveau d'un réseau | 66 |
| 4.2.6 | Discussion sur le décodage | 71 |
| 4.3 | Stratégie Orientée Flux (FOS : Flow-Oriented Strategy) | 71 |
| 4.3.1 | Introduction | 71 |
| 4.3.2 | Définitions et hypothèses | 71 |
| 4.3.3 | FOS au niveau d'un nœud de codage | 72 |
| 4.3.3.1 | Analyse des délais | 73 |
| 4.3.3.2 | Délai maximum dans un nœud intermédiaire de premier ordre | 74 |
| 4.3.4 | FOS au niveau d'un réseau | 75 |
| 4.3.5 | Discussion | 77 |
| 4.4 | Stratégie de Transfert Rapide (FFS : Fast Forwarding Strategy) | 77 |
| 4.4.1 | Introduction | 77 |

| | | |
|----------|---|-----------|
| 4.4.1.1 | Objectifs et définition de la stratégie de transfert rapide | 77 |
| 4.4.2 | Définitions et hypothèses | 78 |
| 4.4.3 | Présentation de la stratégie FFS | 79 |
| 4.4.4 | FFS au niveau d'un nœud de codage | 81 |
| 4.4.4.1 | Temps d'arrivée à un nœud intermédiaire | 82 |
| 4.4.4.2 | Durée séparant deux blocs consécutifs dans un nœud intermédiaire | 83 |
| 4.4.4.3 | Taille de file d'attente dans un nœud intermédiaire | 83 |
| 4.4.4.4 | Délai maximum dans un nœud intermédiaire | 84 |
| 4.4.5 | FFS au niveau d'un réseau | 85 |
| 4.4.6 | FFS au niveau de récepteurs | 85 |
| 4.4.7 | Discussion et considérations pratiques | 87 |
| 4.4.7.1 | Synchronisation des nœuds | 87 |
| 4.4.7.2 | Délais de transmission | 87 |
| 4.4.7.3 | Mise à jour des paramètres temporels | 87 |
| 4.5 | Stratégie Classique de Routage/Multiplexage (MS : Multiplexing Strategy) | 89 |
| 4.5.1 | Introduction | 89 |
| 4.5.2 | MS au niveau d'un nœud | 89 |
| 4.5.3 | MS au niveau d'un réseau | 90 |
| 4.6 | Conclusion générale | 92 |
| 5 | Évaluation du délai maximum | 93 |
| 5.1 | Introduction | 94 |
| 5.2 | Évaluation de la borne maximale du délai au niveau d'un nœud | 94 |
| 5.2.1 | Rappel des résultats obtenus | 94 |
| 5.2.1.1 | Stratégie orientée flux (FOS) | 95 |
| 5.2.1.2 | Stratégie de transfert rapide (FFS) | 95 |
| 5.2.1.3 | Stratégie de routage/multiplexage | 95 |
| 5.2.1.4 | Conclusion | 96 |
| 5.2.2 | Stratégie de transfert rapide (FFS) vs stratégie orientée flux (FOS) | 96 |
| 5.2.3 | Stratégie de transfert rapide (FFS) vs Stratégie de routage / multiplexage (MS) | 97 |
| 5.2.4 | Stratégie orientée flux (FOS) vs Stratégie de routage / multiplexage (MS) | 99 |
| 5.2.5 | Application numérique | 102 |

Table des matières

| | | |
|----------|--|------------|
| 5.3 | Évaluation de la borne maximale du délai d'un réseau de bout-en-bout | 103 |
| 5.3.1 | Cas d'Étude -1 : Réseau "Butterfly" | 104 |
| 5.3.1.1 | Délai maximum de bout-en-bout - stratégie orientée flux (FOS) | 106 |
| 5.3.1.2 | Délai maximum de bout-en-bout - stratégie de transfert rapide (FFS) | 109 |
| 5.3.1.3 | Délai maximum de bout-en-bout - stratégie de routage / multiplexage (MS) | 111 |
| 5.3.1.4 | Application numérique | 112 |
| 5.3.2 | Cas d'Étude -2- Réseau avec de multiples flux entrants | 114 |
| 5.3.2.1 | Application numérique | 117 |
| 5.3.3 | Cas d'Étude -3- Réseau avec de multiples niveaux du codage / multiplexage | 118 |
| 5.3.3.1 | Application numérique | 121 |
| 5.4 | Discussion | 122 |
| 6 | Évaluation et analyse du délai maximum par simulation | 123 |
| 6.1 | Introduction | 124 |
| 6.2 | Simulation | 124 |
| 6.2.1 | Présentation de simulateur | 124 |
| 6.2.2 | Modélisation des trafics et des délais | 126 |
| 6.3 | Application, analyse et évaluation du délai maximum d'un réseau de bout-en-bout par simulation | 126 |
| 6.4 | Stratégie de transfert rapide (FFS) | 128 |
| 6.5 | Stratégie orientée flux (FOS) | 128 |
| 6.6 | Stratégie de Routage / Multiplexage (MS) | 130 |
| 6.7 | Comparaisons des résultats de simulation obtenus par les trois stratégies | 132 |
| 7 | Conclusion | 135 |
| | Bibliographie | 139 |

Table des figures

| | | |
|------|--|----|
| 1.1 | Exemple de routage classique et de codage réseau | 4 |
| 2.1 | Exemple de graphe d'un réseau | 13 |
| 2.2 | Conservation des flux | 14 |
| 2.3 | Un exemple de codage réseau aléatoire. M_1, M_2, \dots, M_n sont les processus de la source émis en multicast vers les récepteurs. Les coefficients g_i et h_i sont des éléments choisis aléatoirement dans un corps fini. | 18 |
| 2.4 | Une connexion entre une source et un récepteur | 21 |
| 2.5 | Les processus dans un nœud intermédiaire | 21 |
| 2.6 | Graphe de nœuds et de liens | 24 |
| 2.7 | Graphe d'étiquette des liens | 25 |
| 2.8 | Le délai de réception des nœuds récepteurs | 30 |
| 2.9 | Économie d'énergie grâce au codage réseau sur un exemple de réseau ad hoc sans fil | 30 |
| 2.10 | Exemple de l'intérêt du codage réseau | 31 |
| 2.11 | Symboles reçus par un récepteur | 33 |
| 2.12 | Chaque récepteur reçoit une combinaison de vecteurs de la source | 34 |
| 2.13 | Principe du buffering | 36 |
| 3.1 | Le concept de réseau redondant | 46 |
| 3.2 | Un lien virtuel | 47 |
| 3.3 | (a) : Le BAG dans un VL pour un flux de données maximum sur la largeur de bande. (b) : Le BAG dans un VL pour un flux de données non maximum sur la largeur de bande. | 48 |
| 3.4 | Régulation du flux d'un lien virtuel | 48 |
| 3.5 | Mécanisme de control de flux avec un ordonnanceur | 48 |
| 3.6 | L'effet de la gigue sur un flux de données maximum sur la largeur de bande | 49 |
| 3.7 | L'intérêt du codage réseau en termes de robustesse | 50 |
| 3.8 | L'intérêt du codage réseau en termes de déterminisme | 51 |
| 3.9 | Contrôleur du flux de type "Seau Percé" | 55 |
| 3.10 | Bornes maximales des délais et des tailles des files d'attente | 56 |
| 4.1 | Courbe d'arrivée du flux F_1 en forme d'escalier | 63 |
| 4.2 | Architecture d'un nœud de codage | 64 |

Table des figures

| | | |
|------|--|-----|
| 4.3 | Un réseau acyclique, orienté avec des liens sans délais et avec deux sources qui produisent les flux d'entrée R_1, R_2 avec des courbes d'arrivée respective α_1 et α_2 . Les deux récepteurs sont les nœuds (5) et (6) qui produisent les flux de sortie R_1^*, R_2^* avec des courbes d'arrivée respective α_1^* et α_2^* . Les nœuds (1), (2), (3) et (4) sont des <i>nœuds de codage</i> . Chaque nœud offre une courbe de service donnée à ses flux d'entrée vers ses flux de sortie. | 67 |
| 4.4 | Nœud avec n flux d'entrée | 73 |
| 4.5 | Courbe de sortie | 75 |
| 4.6 | Transmission d'un flux à travers n nœuds. | 76 |
| 4.7 | Les fonctionnements de : (a)- la stratégie orientée flux. (b)- la stratégie de transfert rapide. | 80 |
| 4.8 | Stratégie de transfert rapide | 82 |
| 4.9 | Un exemple de temps d'arrivée à un nœud intermédiaire avec deux sources | 84 |
| 4.10 | FFS au niveau de récepteurs | 86 |
| 4.11 | Réglage d'intervalle de temps d'arrivée avec des décalages des horloges | 89 |
| 4.12 | Transmission d'un flux à travers n nœuds de multiplexage. | 91 |
| 5.1 | Délais maximums dans un nœud | 102 |
| 5.2 | Régions favorables pour les stratégies FOS et MS | 103 |
| 5.3 | Cas d'étude -1- Réseau "Butterfly" | 105 |
| 5.4 | Exemple d'un flux étudié | 105 |
| 5.5 | Délais maximums : réseau butterfly | 115 |
| 5.6 | Cas d'étude -2- Réseau avec multiples flux entrants | 116 |
| 5.7 | Délais maximums : réseau avec multiples flux entrants | 117 |
| 5.8 | Cas d'étude -3- Réseau avec multiples niveaux du codage / multiplexage | 119 |
| 5.9 | Délais maximums : réseau avec multiples niveaux du codage / multiplexage | 120 |
| 6.1 | Paquet codé | 125 |
| 6.2 | Délais maximums théoriques | 127 |
| 6.3 | Stratégie FFS : Délais Maximums | 129 |
| 6.4 | Stratégie FOS : Délais Maximums | 130 |
| 6.5 | Stratégie MS : Délais Maximums | 131 |
| 6.6 | Délais maximums par simulation | 132 |
| 7.1 | Réseau "butterfly"- délais moyens | 137 |

Liste des tableaux

| | | |
|------|---|-----|
| 5.1 | Délais maximum au niveau d'un nœud | 96 |
| 5.2 | Délais maximum d'attente : comparaison entre FOS et FFS | 96 |
| 5.3 | Délais maximum 1 : comparaison entre FFS et MS | 97 |
| 5.4 | Délais maximum : comparaison entre les stratégies FOS et MS | 100 |
| 5.5 | Délais maximum de bout-en-bout | 106 |
| 5.6 | Capacité totale : réseau "butterfly" | 112 |
| 5.7 | Délais maximums de bout-en-bout : réseau "butterfly" | 113 |
| 5.8 | Capacité totale : réseau avec multiples flux entrants | 117 |
| 5.9 | Délais maximums de bout-en-bout : réseau avec multiples flux entrants | 118 |
| 5.10 | Capacité totale : réseau avec multiples niveaux du codage / multiplexage | 118 |
| 5.11 | Délais maximums de bout-en-bout : réseau avec multiples niveaux du codage / multiplexage | 121 |

Liste des tableaux

Résumé

L'intérêt du codage réseau (network coding) pour améliorer le débit ou optimiser l'utilisation de la capacité du réseau a été clairement démontré dans différents contextes. Certains travaux ont notamment montré que le codage réseau permet de diminuer le délai (maximal et moyen) de transmission de bout-en-bout d'un paquet. Ceci est dû au fait que le traitement simultané de plusieurs paquets dans un nœud de codage permet de réduire le temps passé par les paquets dans les files d'attente par rapport au routage classique.

Dans cette thèse, nous considérons l'application du codage réseau dans le contexte des réseaux proposant des garanties de qualité de service (QoS).

Notre principale contribution est la proposition de trois stratégies de codage réseau assurant un niveau de QoS garantie exprimé en termes de délai de bout-en-bout.

La première stratégie, appelée "stratégie orientée réseau" est une stratégie de codage aléatoire, en termes de dates d'arrivée des paquets, permettant de réduire au maximum le temps passé par les paquets dans les files d'attente des routeurs. Le point faible de cette approche, comme toute approche aléatoire, est qu'elle n'est pas totalement fiable. Les deux autres stratégies proposées implémentent une stratégie fiable en utilisant le concept de code en bloc. La première, appelée "stratégie orientée flux" est basée sur la définition classique du codage réseau alors que la seconde, appelée "stratégie de transfert rapide", permet de réduire les temps d'attente des paquets dans les files d'attente en les transférant sans attendre tous les paquets du même bloc.

Les délais maximums engendrés par les différentes stratégies ont été évalués au niveau d'un nœud de codage en utilisant le calcul réseau (network calculus). Les bornes de délais de bout-de-bout ont ensuite été calculées pour plusieurs types de réseaux. Dans la plupart des cas, ces bornes sont meilleures que celles obtenues pour le routage classique.

Les stratégies de codage réseau fiables et la stratégie de routage ont été implémentées et évaluées par simulation sur les réseaux étudiés précédemment. Les résultats obtenus montrent que les pires cas de délais de bout-en-bout observés ont les mêmes comportements que les bornes maximales théoriques calculées, validant ainsi les stratégies proposées.

Mots-clés: Codage Réseau, Délai, Stratégie de Codage, Nœud de Codage, Routage, Calcul Réseau.

1

Introduction

Sommaire

| | | |
|------------|--|----------|
| 1.1 | Présentation de la problématique et des objectifs | 4 |
| 1.2 | Organisation du document | 6 |

1.1 Présentation de la problématique et des objectifs

Les évolutions technologiques actuelles de nos sociétés ont pour conséquence de demander toujours plus de services aux réseaux de communications. Pour répondre à ces attentes, une solution consiste à optimiser les techniques développées dans un domaine de recherche. Une autre solution consiste à combiner des techniques issues de plusieurs domaines de recherche.

Un exemple réussi est le rapprochement des domaines de la théorie de l'information et de celle des réseaux qui a notamment débouché sur la définition de codes à effacement très efficaces et surtout qui a permis l'introduction du codage réseau.

L'objectif principal du codage réseau est de définir des techniques permettant d'atteindre la capacité théorique d'un réseau. L'innovation majeure consiste à autoriser les nœuds intermédiaires du réseau à réaliser des opérations de codage sur des flux entrants pour produire des flux sortants.

Nous pouvons illustrer le codage réseau en considérant le réseau de transmission représenté sur la Figure 1.1(a). Dans ce réseau, la source S veut transmettre en multicast deux bits b_1 et b_2 vers les deux nœuds Y et Z . Nous considérons que chaque lien a une capacité de 1 bit/s.

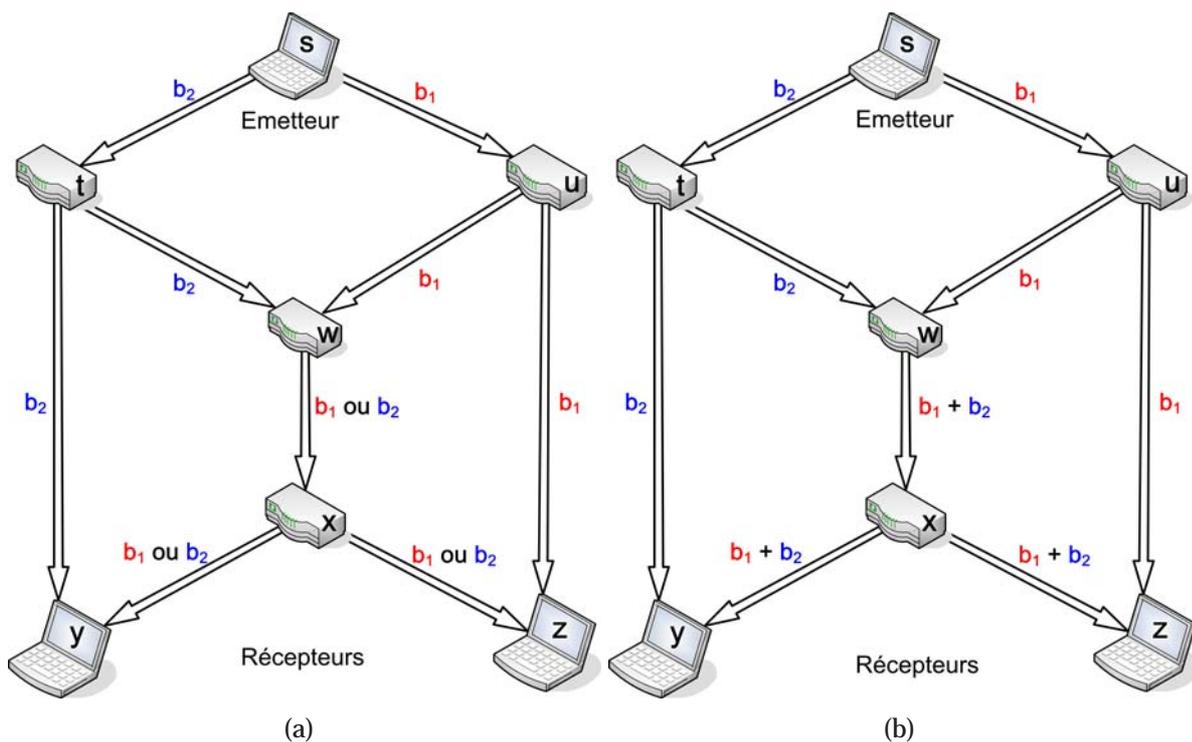


FIGURE 1.1 – Exemple de routage classique et de codage réseau

La solution la plus "simple", qui est utilisée par un routage classique, consiste à transmettre le bit b_1 sur les liens (S, T) , (T, W) , (T, Y) et le bit b_2 sur les liens (S, U) , (U, W) , (U, Z) . Sur le lien (W, X) , on transmet alternativement le bit b_1 , puis le bit b_2 .

1.1. Présentation de la problématique et des objectifs

Lorsque X reçoit b_1 , il le retransmet sur le lien (X, Y) et lorsqu'il reçoit b_2 , il le retransmet sur le lien (X, Z) . Avec un tel schéma, les récepteurs Y et Z reçoivent en moyenne 1,5 bits par unité de temps.

La solution utilisant le codage réseau est représentée sur la Figure 1.1(b). La différence avec le routage classique concerne tout d'abord les liens (W, X) , (X, Y) et (X, Z) . Ces liens transmettent la somme (XOR) $b_1 + b_2$. Le nœud Y reçoit à chaque unité de temps le bit b_1 et la somme $b_1 + b_2$. Il lui est alors facile de retrouver la valeur de b_2 avec l'opération $(b_1 + b_2) - b_1$, qui, au niveau des bits, se traduit par une opération XOR. Le nœud Z , avec le même type d'opération, récupère lui aussi les 2 bits b_1 et b_2 à chaque unité de temps.

Cet exemple illustre bien l'intérêt du codage réseau car il permet une augmentation du débit de réception de 33% (passage de 1,5 bits/s à 2 bits/s). Notons que sur cet exemple, la capacité multicast du réseau, qui est de 2 bit/s pour chaque récepteur, ne peut être atteinte que par des techniques de codage réseau.

L'intérêt du codage réseau est actuellement clairement établi par de nombreux résultats théoriques et pratiques [1][2][3].

Si, d'un point de vue théorique, le gain principal obtenu avec cette technique concerne la capacité du réseau, ce gain se traduit dans la pratique par des améliorations importantes en termes de débit ou de fiabilité des réseaux réels pour des flux multicast, broadcast ou unicast.

Au niveau pratique, les implémentations du codage réseau doivent être définies en tenant compte des contraintes des réseaux réels : paquets, taille variable des paquets, débit variable, gigue, ... etc. De multiples stratégies et approches ont été proposées pour adapter les nœuds de codage, c'est-à-dire les nœuds effectuant des opérations de codage sur les flux entrants, aux contraintes particulières des différents réseaux. ces réseaux peuvent être des réseaux de recouvrement (*overlays*), des réseaux Internet classiques ou des réseaux sans fil [4] [5][6][7] [8] ou d'autres types de réseaux.

Parmi les avantages du codage réseau, une conséquence de l'optimisation de l'utilisation de la capacité du réseau est l'amélioration des délais de transmission (maximal et moyen) de bout-en-bout. Pour un fichier complet, l'évaluation des gains de performance et de délai de plusieurs stratégies de codage réseau a été étudiée dans le cas d'une transmission des fichiers d'une station de base vers de multiples récepteurs [9].

La problématique considérée dans notre travail concerne également le délai de bout-en-bout, mais les délais sont évalués au niveau des paquets et non pas des fichiers complets. Pour certaines applications temps-réel, le délai de chaque paquet est extrêmement important et une borne théorique doit pouvoir être calculée. Un bon exemple de ces applications sont les réseaux avioniques (par exemple AFDX, sur l'Airbus A380) où, pour pouvoir être normalisés, ces réseaux doivent pouvoir fournir une borne supérieure du délai de transmission de chaque paquet.

Le point de départ de notre travail est la supposition que le codage réseau peut permettre de diminuer les bornes maximales de bout-en-bout car une partie importante des délais est due aux temps d'attente dans les routeurs. En effet, avec une stratégie de codage réseau, un nœud de codage traite plusieurs paquets simultanément, ce qui peut réduire le temps passé par les paquets dans les files d'attente.

Une première contribution de cette thèse a consisté à définir des stratégies de co-

Chapitre 1. Introduction

dage permettant de réduire les *bornes de délais de bout-en-bout*.

Une des principales méthodes pour calculer les bornes de délais en intégrant les différentes garanties et contraintes caractérisant le réseau et les flux est le calcul réseau. Le calcul réseau [10] est un cadre mathématique basé sur l'algèbre Min-Plus permettant d'obtenir des limites déterministes (bornes maximales et minimales) des paramètres de qualité de service (QoS) tels que le délai de bout-en-bout ou les tailles des files d'attente.

Cette théorie a été présentée et développée dans [10] en généralisant les travaux précédents comme [11][12]. Dans cette approche, les contraintes de la régularité et du débit des flux sont exprimées en terme de *courbe d'arrivée*. Les nœuds fournissent des services caractérisés par une *courbe de service* à chaque flux d'entrée vers chaque flux à la sortie. Cette théorie permet de combiner ces notions de courbes d'arrivée et de courbes de service.

La deuxième contribution de notre travail a été de définir les méthodes permettant d'appliquer cette théorie à des nœuds réalisant des opérations de codage réseau.

Les principaux objectifs adressés dans ce travail ont été les suivants. En considérant :

- les contraintes sur les flux d'entrée.
- les garanties de services des éléments du réseau.
- que les nœuds sont capables d'appliquer le mécanisme de codage réseau.

Pouvons-nous :

1. définir les stratégies du codage dans ces nœuds ?
2. améliorer les niveaux des garanties offerts aux flux (délai maximal de bout-en-bout, taille des files d'attente, ...) avec le codage réseau ?

1.2 Organisation du document

Dans le Chapitre 2, nous présenterons les principaux concepts théoriques du codage réseau. Puis, nous détaillerons les intérêts de l'application du codage réseau dans différents domaines et types de réseaux.

Dans le Chapitre 3, nous présenterons les caractéristiques des réseaux fournissant des garanties de qualité de service (QoS). Quelques exemples de ce type de réseaux seront proposés. Ensuite, nous expliquerons en quoi le codage réseau peut être utile dans ce type de réseau. Enfin, nous introduirons brièvement les notions de base du calcul réseau.

Dans le Chapitre 4, trois nouvelles stratégies de codage réseau seront proposées pour les réseaux fournissant des qualités de service (QoS) garanties. Pour chacune de ces stratégies, nous calculerons des bornes de délai de transmission au niveau d'un nœud réseau, puis au niveau d'un réseau complet.

Dans le Chapitre 5, une évaluation des résultats de ces deux dernières nouvelles stratégies sera effectuée et une comparaison avec une stratégie de routage / multiplexage sera proposée. Ces différentes stratégies seront étudiées pour comparer les bornes de délais dans trois réseaux particuliers.

1.2. Organisation du document

Dans le Chapitre 6, des bornes maximales des délais de bout-en-bout seront évaluées par simulation sur les trois réseaux étudiés précédemment. Une description des caractéristiques du simulateur développé sera tout d'abord présentée. Puis, une comparaison entre les résultats théoriques et les résultats pratiques des délais maximaux de bout-en-bout des trois réseaux sera donnée.

Le dernier chapitre présente les principales conclusions de ce travail ainsi que les travaux futurs.

Chapitre 1. Introduction

2

État de l'art et notions de base

Sommaire

| | | |
|------------|---|-----------|
| 2.1 | Introduction | 10 |
| 2.2 | Codage réseau | 10 |
| 2.2.1 | Introduction du codage réseau | 10 |
| 2.2.2 | Codage réseau et théorie des graphes | 13 |
| 2.2.3 | Les principales notions théoriques du codage réseau | 15 |
| 2.3 | Bénéfices et intérêts de codage réseau | 28 |
| 2.3.1 | Augmentation de la capacité multicast et du débit | 29 |
| 2.3.2 | Diminution de l'énergie par bit | 29 |
| 2.3.3 | Robustesse et tolérance aux fautes | 31 |
| 2.4 | Applications de codage réseau dans différents domaines et types de réseaux | 32 |
| 2.4.1 | Internet | 32 |
| 2.4.2 | Réseaux de recouvrement "Overlay" | 36 |
| 2.4.3 | Réseaux ad-hoc et sans fil | 37 |

2.1 Introduction

Le codage réseau a des applications intéressantes sur plusieurs types de réseaux de communications réels. Le premier exemple est l'Internet, où il peut être utilisé à plusieurs niveaux tels que la couche IP ou la couche application. On peut citer aussi les réseaux ad hoc sans fil. Toutefois, dans chacun de ces réseaux, il y a un écart significatif entre le travail théorique sur le codage réseau et les applications du codage réseau dans des réseaux réels.

Dans la première partie de ce chapitre, nous développerons un état de l'art du **codage réseau** en rapportant les travaux réalisés jusqu'à présent dans ce domaine. Puis, nous développerons les principales notions théoriques liées à la théorie du codage réseau.

Dans la deuxième partie, plusieurs exemples sur les bénéfices et les intérêts de l'utilisation de codage réseau seront présentés.

Dans la troisième partie, nous présentons les applications faites ou déjà proposées de codage réseau dans différents domaines et types de réseaux.

2.2 Codage réseau

Dans cette partie, nous commencerons par une présentation du codage réseau dans laquelle nous exposerons les principales étapes qui ont mené à cette théorie. Ensuite, nous présenterons les principaux axes de recherche dans ce domaine en mettant l'accent sur les travaux dont les stratégies ou les mécanismes sont proches des nôtres.

Pour des informations plus détaillées, une bonne référence est le livre de R. Yeung [13] qui aborde les aspects théoriques de la théorie de l'information du codage réseau. D'autres bonnes références sur ce sujet sont les livres de C. Fragouli et E. Soljanin [14][15]. Une autre source bibliographique est la page d'accueil du codage réseau [16], qui contient une grande collection de références dans ce domaine.

2.2.1 Introduction du codage réseau

Le codage réseau est un concept récemment introduit par R. Ahlswede et al. [1] pour améliorer le débit des transmissions d'information multicast dans un réseau formé d'une ou de plusieurs sources et d'un ou de plusieurs récepteurs.

Le principe de ce mécanisme consiste à autoriser les nœuds intermédiaires (situés sur des chemins entre les sources et les récepteurs) à coder (combiner) les paquets entrants au lieu de les expédier simplement comme dans la stratégie de routage classique.

Les auteurs de [1] ont prouvé que si le codage réseau est utilisé par les nœuds intermédiaires pour coder les différents flux, le débit multicast maximum (capacité multicast) auquel une source peut envoyer l'information à un ensemble de récepteurs, est donné par la plus petite coupe entre la source et n'importe lequel récepteur quand la taille de corps tend vers l'infini (voir Section 2.2.3.1).

S-Y. R. Li, R. W. Yeung et N. Cai [17] ont prouvé que l'utilisation du codage réseau linéaire avec des symboles de taille finie permet d'atteindre la capacité multicast maxi-

mum. Toutes les opérations de codage linéaire sont définies en tant que combinaisons linéaires sur un corps fini.

R. Koetter et M. Médard [2][3] ont montré que le codage réseau peut améliorer le débit pour des transmissions multicast ou unicast. Une borne maximum pour le débit multicast est identifiée aussi comme la plus petit coupe entre la source et n'importe quel récepteur. Ils ont prouvé l'existence de codes linéaires qui atteignent la capacité multicast avec un corps fini de taille $(h \cdot |T|)$ où h est le débit de la source et $|T|$ le nombre de récepteurs.

Pour les réseaux acycliques, S. Jaggi, P. Sanders, et al. [18][19] ont montré comment trouver des coefficients de codage et de décodage en temps polynomial. Deux versions d'algorithmes ont été proposées : une version aléatoire, qui est la plus rapide en moyenne, et une version déterministe.

Des propositions d'application du codage réseau ont été proposées pour un grand nombre de réseaux et pour un grand nombre de contextes.

Un premier exemple est l'Internet, où le codage réseau peut être employé à 2 niveaux : sur la couche IP (routeurs) et au niveau de la couche d'application (réseaux peer-to-peer, réseaux de distribution de contenu, réseaux overlay).

D'autres exemples sont les réseaux ATM, les réseaux ad hoc et sans fil, et généralement tous les réseaux qui emploient des transmissions par paquets. Cependant, dans chacun de ces réseaux, il y a une différence entre le travail théorique sur le codage réseau et les applications pratiques dans ces réseaux réels.

En effet, les premiers travaux théoriques antérieurs sur le codage réseau ont supposé que le flux de symboles (paquets) est synchronisé dans le réseau et que les liens ont des capacités semblables. Par ailleurs, théoriquement, on suppose souvent avoir au moins une connaissance de la topologie du réseau afin de construire le code. En outre, les graphes sont souvent supposés sans cycles. Enfin, le problème de l'hétérogénéité des récepteurs et les problèmes de la congestion et des pertes sur les liens et dans les routeurs sont souvent ignorés. D'autres problèmes apparaissant dans les réseaux réels compliquent l'application du codage réseau dans des réseaux réels. Par exemple, l'information est transmise d'une manière asynchrone en paquets, les paquets sont sujets à des délais et des pertes aléatoires sur chaque lien et nœud. Les liens ont, dans la majorité des cas, des capacités inconnues. De plus, il est extrêmement difficile d'avoir une connaissance totale du réseau. Enfin, il faut noter que la topologie du réseau peut varier dans certains types de réseaux tels que les réseaux ad-hoc ou les réseaux pair-à-pair.

Un autre type de problèmes dans les réseaux réels concerne les latences et les variations de délai. Les variations de trafic engendrent des problèmes de congestions qui peuvent générer des latences. Pour contrôler les variations de la capacité des liens, des files d'attente sont employées dans les routeurs. La limitation des tailles de ces files d'attente peut produire des pertes.

Pour appliquer le codage réseau à des réseaux réels, plusieurs approches ou stratégies de politiques de codage dans les nœuds intermédiaires ont été proposées pour intégrer des contraintes pratiques.

Un premier ensemble d'approches, présenté dans [20] [5] [21], consiste à effectuer des combinaisons linéaires aléatoires des paquets d'entrée. Dans [20], il est montré que

la probabilité qu'un récepteur puisse décoder avec succès les données reçues peut être majorée par une borne qui est fonction de la taille du corps fini utilisé.

Dans [21][20], [22] ou [23], un codage aléatoire est appliqué aux réseaux de pair-à-pair et aux réseaux sans fil.

Dans différents contextes, [19] ou [5] présentent des solutions assurant le codage de tous les paquets dans des nœuds de codage. Dans ces deux derniers travaux, les implémentations sont basées sur des stratégies particulières de nœuds de réseau permettant de gérer, entre autres, les arrivées asynchrones de données. Ces stratégies impliquent un stockage des informations dans les nœuds de codage afin d'attendre l'arrivée des informations de la même génération (voir le concept de génération dans la partie 2.4.1.2) provenant d'autres liens pour réaliser le codage et les coder avec l'autre information entrante.

Une proposition alternative pour les réseaux contenant des cycles est d'adopter une approche de codage continue [2] [6] où l'information des différentes périodes de temps est combinée et des mémoires sont nécessaires aux récepteurs pour le décodage.

D'autres travaux, comme [4] ou [24], ont utilisé une approche classique pour combiner les paquets selon un code réseau déterminé *a priori*. Cette deuxième approche, qui est théoriquement la plus efficace, peut entraîner un stockage des paquets dans des files d'attente dans des nœuds de codage et par conséquent, une augmentation des délais de bout-en-bout.

L'approche classique (non-aléatoire), qui peut atteindre théoriquement la meilleure performance, a été seulement appliquée dans quelques réseaux particuliers. Pour cette approche, la taille moyenne des files d'attente et le délai ont été étudiés dans [25] avec l'hypothèse que les flux d'entrée sont des flux de Poisson indépendants.

Récemment, une approche mixte qualifiée d'"opportuniste" a été présentée dans [26] et développé dans [27]. Le fait que le canal soit naturellement à diffusion sur un lien sans fil est exploitée en effectuant des combinaisons linéaires des paquets d'entrée selon la connaissance du contexte et quand l'occasion est favorable.

L'intérêt de codage réseau dans les réseaux relais est montré dans [7]. Plusieurs stratégies pour implémenter le codage réseau au niveau physique de ce type réseaux sont présentées.

Dans les réseaux relais, les transmissions simultanées sont habituellement considérées comme des interférences qui ont des effets négatifs.

La stratégie du codage réseau dans la couche physique, proposé dans [28], introduit l'idée de considérer les interférences comme des somme de signaux (et donc un codage). Ce principe peut être appliqué dans les réseaux ad-hoc sans fil et peut permettre une augmentation du débit [28]. Cette stratégie suppose une synchronisation totale des phases des signaux envoyés par les deux sources.

Dans [29], les auteurs exploitent le fait que les nœuds de relais n'ont pas besoin de décoder les informations des sources. Cette stratégie suppose que le nœud relais ne s'occupe pas de décoder le signal mixte reçu mais tout simplement il l'amplifie et le transfère - comme une somme réels des signaux reçus - vers les destinations. Chaque récepteur connaît comment extraire les paquets qui lui manquent car il connaît ses paquets originaux. Une nouvelle méthode pour extraire les paquets désirés est présentée aussi dans [29].

Une autre stratégie basé sur THP "Tomlinson-Harashima precoding" est présentée dans [30]. Elle propose d'utiliser des opérations modulo dans les nœuds de relais et le récepteurs pour réduire la consommation d'énergie nécessaire aux les nœuds relais avec uniquement un léger impact sur les performances. Il est prouvé que l'utilisation de ces approches de codage réseau sur la couche physique améliore la performance.

Nous ne détaillons plus ces deux dernières approches ("opportuniste" et couche physique) car nous estimons qu'elles ne sont pas liée directement à notre travail.

Récemment, une analyse des performances de délai et de files d'attente avec codage réseau dans des nœuds intermédiaires d'un réseau de "stockages et transferts" a été présentée dans [31]. Ces nœuds encoder l'ensemble des flux qui passent par eux. Deux stratégie de codage sont présentées : un codage synchrone (SNC) et un codage partiellement asynchrone (APNC).

Généralement, ils sont trouvés que (SNC) n'est pas adapté à la mise en œuvre pratique sauf si il est utilisée dans un cadre entièrement synchronisés ou dans un réseau hautement saturées. et proposé que les nœuds de réseau le utilise de manière adaptative chaque fois que les débits de flux d'entrée saturent le réseau. Les résultats ont aussi montré que le codage réseau peut améliorer la performance globale de manière significative à condition qu'il soit utilisé de manière opportuniste (APNC).

2.2.2 Codage réseau et théorie des graphes

Dans ce travail, un "graphe" représente une collection de nœuds et une collection de liens qui relie des paires de nœuds.

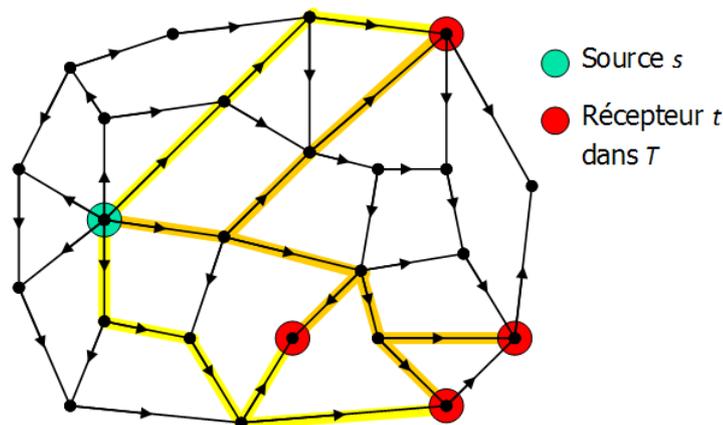


FIGURE 2.1 – Exemple de graphe d'un réseau

Nous définissons un graphe comme $G = (V, E)$, où V est l'ensemble de nœuds et E est l'ensemble de liens. Un graphe, peut être orienté si ses liens sont orientés d'un nœud à l'autre et il peut être non orienté, ce qui indique qu'il n'y a aucune distinction entre les deux nœuds liés à chaque lien. Un lien $e \in E$ où $e = (x, y)$ si $x, y \in V$. Si un lien $e = (x, y)$ est considéré comme orienté de x vers y , alors le nœud x est appelé

Chapitre 2. État de l'art et notions de base

nœud initial et nœud Y est appelé nœud terminal du lien. Un graphe fini est un graphe avec un nombre fini de liens et de nœuds.

Deux liens d'un graphe sont adjacents s'ils partagent un nœud commun. De même, deux nœuds sont dits adjacents s'ils partagent un lien commun, dans ce cas on dit que le lien commun joint les deux nœuds. Un lien sortant de nœud x est un lien incident à x vers l'extérieur et un lien entrant dans un nœud y est un lien incident à y vers l'intérieur.

D'un point de vue théorique, on appelle réseau un graphe orienté où à chaque lien est attribuée une capacité, qui correspond au débit maximum que peut supporter ce lien.

Un flux réseau correspond à une attribution de flux aux liens d'un réseau, tels que la quantité de flux associée à chaque lien n'excède pas sa capacité.

Un flux doit satisfaire la contrainte que la quantité de flux entrants dans un nœud est égale à la quantité de flux sortants de ce nœud. Les seules exceptions à cette règle sont les nœuds sources, qui n'ont pas de liens entrants et les nœuds récepteurs qui n'ont pas de liens sortants.

Soit $G = (V, E)$ un graphe orienté fini dans lequel chaque lien $(u, v) \in E$ a une capacité $c(u, v)$. On considère que si $(u, v) \notin E$ alors $c(u, v) = 0$.

Considérons une source s et un récepteur t . Un flux de réseau est une fonction réelle $f : V \times V \rightarrow \mathbb{R}$ telle que pour tous les nœuds u et v :

- **Contraintes sur la capacité** : le flux sur chaque lien d'un flux de réseau ne peut pas excéder la capacité du lien. Plus formellement, $f(u, v) \leq c(u, v) \forall (u, v) \in E$.
- **Conservation de flux** : Mis à part les sources et les récepteurs, il n'existe pas des nœuds dans un flux de réseau qui peuvent produire ou stocker des flux. Plus formellement, $\sum_{w \in V} f(u, w) = 0$ sauf si $u = s$ ou $u = t$. Les flux entrants sont identiques aux flux sortants, ou en d'autres termes le flux net est nul.

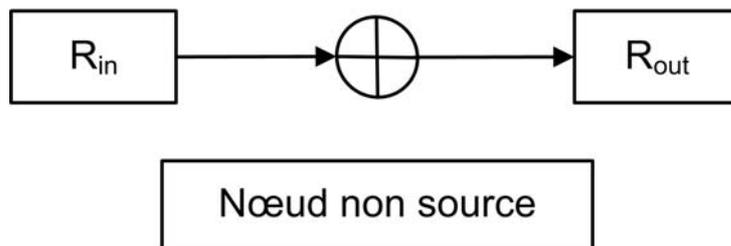


FIGURE 2.2 – Conservation des flux

Si on note $I(x)$ (respectivement $O(x)$) l'ensemble des liens entrants (respectivement sortants) du nœud x , on a :

$$\sum_{e \in I(x)} f(e) = \sum_{e \in O(x)} f(e)$$

$$\sum_{e \in O(s)} f(e) - \sum_{e \in I(s)} f(e) = \sum_{e \in I(t)} f(e) - \sum_{e \in O(t)} f(e)$$

Toutes ces notions sont détaillées dans divers ouvrages, tel que par exemple [32].

2.2.3 Les principales notions théoriques du codage réseau

2.2.3.1 Théorie de Min-cut Max-flux

Si la valeur d'un flux est aussi grande que possible alors le flux s'appelle un flux maximal.

Une coupe (cut) d'un graphe est une division des nœuds V en deux ensembles S et T . Tout lien $(u, v) \in E$ où $u \in S$ et $v \in T$ traverse cette coupe. On dit que ce lien appartient à la coupe.

La taille d'une coupe est le nombre de liens traversant la coupe. Lorsque des poids sont associés à chaque lien, la taille de la coupe est définie la somme de poids des liens traversant la coupe. Pour un flux réseau, la taille d'une coupe est définie comme la somme des capacités des liens traversant la coupe.

Une coupe est minimale si la taille de la coupe est inférieure ou égale à la taille de toutes les autres coupes. Il existe une seule et unique valeur minimale de la coupe et plusieurs coupes minimales possibles.

Une coupe minimale est considérée comme un accès restreint sur la transmission de données entre la source S et le récepteur T .

La **théorie de Min-cut Max-flow** (coupe-minimale flux-maximal) affirme que le débit maximal qui peut être envoyé de S vers T est égal à la valeur minimale de la coupe.

Notons $maxflow(T)$ le débit maximum qui peut être envoyé de S vers T .

Considérons un graphe $G = (V, E)$ où tous les liens ont une capacité de 1. Si la coupe minimale entre un nœud source S et un nœud récepteur T est égale à h , alors le débit maximal peut être envoyé de $s \in S$ vers $t \in T$ est égal à h .

Autrement dit, le débit maximum qui peut être observé au niveau d'un nœud correspond au minimum des valeurs de la coupe entre ce nœud et la source. Par exemple, si l'on considère le nœud Y de la Figure 1.1, toutes les coupes entre ce nœud et S ont une valeur de 2. Il peut donc recevoir un flux de 2 bits par unité de temps. Pour les mêmes raisons, le nœud Z peut lui-aussi recevoir un flux de 2 bits par unité de temps. Le problème pour une transmission multicast est d'arriver à transmettre le maximum à chaque nœud. Sur la Figure 1.1, on observe que ce maximum est atteint avec le codage réseau, mais qu'il n'existe pas de solution avec le routage classique pour atteindre ce maximum.

Il arrive souvent que plusieurs récepteurs aient des $maxflow$ différents. Même dans ce cas, le codage réseau est capable de faire en sorte que chaque récepteur reçoive un flux correspondant à son $maxflow$ individuel [17]. Ce résultat est plus fort que celui de [1] qui indique que le taux d'information de la source à un ensemble de nœuds peut atteindre le minimum des $maxflow$ individuels grâce à l'utilisation du codage réseau.

2.2.3.2 Codage réseau linéaire multicast

Après avoir introduit de manière intuitive le codage réseau, nous sommes maintenant prêts à définir plus précisément le codage réseau linéaire LCM.

Soit h le maximum des $maxflow(T)$ sur tout $T \in V$, l'ensemble des nœuds du réseau. Soit Ω un espace vectoriel de dimension h sur un corps fini suffisamment grand.

Chapitre 2. État de l'art et notions de base

Considérons un symbole sur ce corps fini comme l'unité d'information. Dans ce cas, 1 symbole dans le corps fini peut être transmis sur un lien chaque unité de temps.

(LCM) : Un codage linéaire multicast v sur un réseau de transmission (G, S) est une attribution d'un espace vectoriel $v(X)$ à chaque nœud X et d'un vecteur $v(X, Y)$ à chaque lien (X, Y) tels que :

- $v(S) = \Omega$;
- $v(X, Y) \in v(X)$ pour chaque lien (X, Y) ;
- Pour tout ensemble \wp de nœuds "non source" dans le réseau

$$\langle \{v(T) : T \in \wp\} \rangle = \langle \{v(X, Y) : X \notin \wp, Y \in \wp\} \rangle.$$

Les informations "entrantes" d'un nœud peuvent être considérées comme un vecteur d'information qui appartient au sous-espace vectoriel $v(X)$. La quantité d'information atteignant un nœud X est donnée par la dimension de ce sous-espace vectoriel.

En fait, la loi de flux d'information dans un nœud, qui implique que les informations en sortie d'un nœud X sont des combinaisons des informations "entrantes", implique que le vecteur assigné à un lien sortant de X doit être une combinaison linéaire des vecteurs assignés aux liens entrants de T [17].

En pratique, les données circulant sur un lien (X, Y) sont calculées en faisant le produit du vecteur d'information reçu par le vecteur de colonnes $v(X, Y)$, l'ensemble des vecteurs de colonnes formant une matrice qui caractérise le codage au niveau d'un nœud.

Le LCM sur un réseau de communication est complètement déterminé par les vecteurs qui sont assignés aux différents liens ou, de même, par les matrices assignées aux différents nœuds.

Illustrons la construction d'un LCM sur l'exemple suivant.

Considérons la transmission multicast de deux bits b_1, b_2 de S à Y et Z dans le réseau représenté sur la Figure 1.1(b).

On choisit comme corps fini le corps \mathbb{F}_2 (formé des éléments 0 et 1). Sur cette Figure, on peut voir que le maximum des $maxflow(T)$ pour tout nœud T vaut 2. On choisit donc comme espace vectoriel $\Omega = \mathbb{F}_2^2$, c'est-à-dire l'ensemble des couples (b_1, b_2) où b_1 et b_2 sont des éléments de \mathbb{F}_2 .

Pour que $v(S)$ soit égal à Ω , on considère que cette source génère les 2 bits b_1 et b_2 par unité de temps. A partir de là, si on veut envoyer b_1 sur (S, U) et b_2 sur (S, T) , on doit avoir :

$$v(S, U) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$v(S, T) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

car l'information qui passe sur (S, U) (resp. (S, T)) est égale au produit $(b_1, b_2) * \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ (resp. $(b_1, b_2) * \begin{pmatrix} 0 \\ 1 \end{pmatrix}$).

Les nœuds qui ne reçoivent qu'une seule information, comme U ou T sont obligés de la rediriger simplement alors que les nœuds qui reçoivent plusieurs informations (comme W) peuvent combiner ces informations, c'est-à-dire utiliser un vecteur contenant plusieurs termes non nuls. En effet, lorsque tous les vecteurs associés à un code contiennent au plus 1 terme non nul, on est dans le cas du routage classique (si ce terme non nul vaut 1).

Le codage multicast correspondant à la Figure 1.1(b) est donc :

$$v(S, T) = v(T, W) = v(T, Y) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$v(S, U) = v(U, W) = v(U, Z) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

et

$$v(W, X) = v(X, Y) = v(X, Z) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Ce code atteint le *maxflow* de chaque nœud car la dimension de l'espace vectoriel formé par les éléments qui arrivent en Y (resp. Z) est 2 (dimension de $\langle\langle b_1, b_1 + b_2 \rangle\rangle = 2$ (resp. (dimension de $\langle\langle b_1 + b_2, b_2 \rangle\rangle = 2$).

2.2.3.3 Codage réseau linéaire aléatoire

Dans des réseaux ayant des topologies variables et/ou un grand nombre de nœuds, il est très difficile d'utiliser le codage réseau classique où les routeurs réalisent des combinaisons linéaires fixes.

Dans ces conditions, une solution consiste à utiliser du codage réseau aléatoire. Cette approche, proposée dans [6][33], consiste à ce que chaque nœud du réseau effectue des combinaisons linéaires aléatoires dans un corps fini à partir des paquets reçus par leur lien d'entrée. Ils transmettent ensuite les paquets combinés (codés) sur les liens sortants.

Pour décrire plus précisément cette technique, supposons que chaque paquet contienne L bits (si les paquets ne sont pas de la même taille, ils seront complétés avec des zéros). Chaque paquet est composé d'un vecteur de L/m symboles où un symbole dans le corps \mathbb{F}_{2^m} est représenté par m bits consécutifs. Rappelons que la longueur d'un paquet combiné est toujours de L bits après la combinaison de deux paquets de longueur L et que ce paquet contient des informations issues de certains des paquets générés par les sources [34].

Considérons n paquets M_1, \dots, M_n générés par une ou plusieurs sources. Avec un codage réseau linéaire, chaque paquet X dans le réseau est associé à un vecteur de codage qui est un vecteur de coefficients $g = (g_1, \dots, g_n)$ dans $(\mathbb{F}_{2^m})^n$.

Le problème de la construction d'un code réseau consiste à déterminer les combinaisons linéaires que chaque nœud du réseau doit effectuer afin d'assurer que le récepteur reçoive au moins n combinaisons linéaires indépendantes à partir desquelles il peut décoder les paquets sources.

Chapitre 2. État de l'art et notions de base

Avec le codage réseau aléatoire, chaque nœud du réseau choisit ses coefficients de façon indépendante, aléatoire et uniforme dans le corps \mathbb{F}_{2^m} , d'une manière totalement décentralisée. De plus, aucune coordination n'est exigée entre les nœuds dans leur choix des combinaisons linéaires d'entrées et de sorties [6][20].

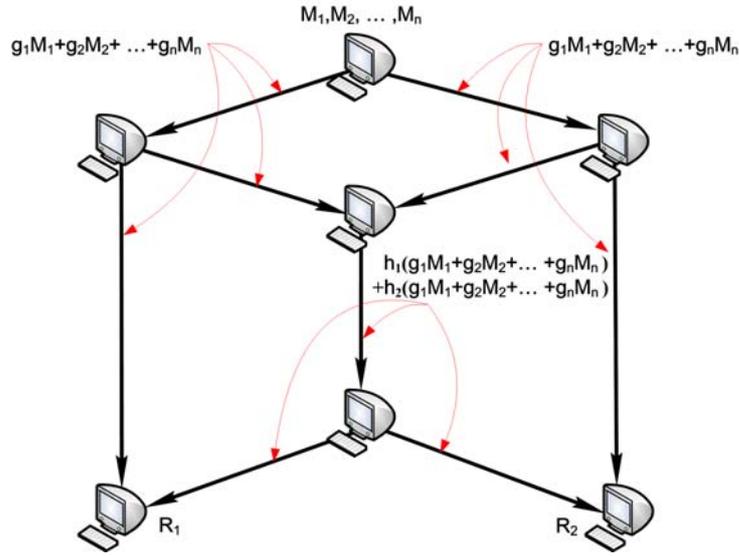


FIGURE 2.3 – Un exemple de codage réseau aléatoire. M_1, M_2, \dots, M_n sont les processus de la source émis en multicast vers les récepteurs. Les coefficients g_i et h_i sont des éléments choisis aléatoirement dans un corps fini.

Dans ce cas, le paquet combiné sera

$$X = \sum_{i=1}^n g_i M_i$$

où g_1, \dots, g_n sont des coefficients choisis de façon aléatoire uniforme dans le corps \mathbb{F}_{2^m} (voir Figure 2.3).

Les paquets étant considérés comme des vecteurs de L/m symboles, la combinaison linéaire doit être réalisée pour chaque position du vecteur c'est-à-dire,

$$X_k = \sum_{i=1}^n g_i M_i^k$$

où $0 \leq k \leq L/m - 1$ et M_i^k et X_k sont respectivement les $k^{\text{ème}}$ symboles de M_i et de X . En plus des données combinées X , le vecteur des coefficients $g = (g_1, \dots, g_n)$ est également stocké dans le paquet combiné. Ces informations supplémentaires ont un coût supplémentaire pour la transmission, mais ce coût diminue proportionnellement avec l'augmentation des longueurs de blocs de données.

Des paquets combinés peuvent également être recombinaés dans des nouveaux nœuds intermédiaires. Supposons qu'un nœud intermédiaire reçoive un ensemble de paquets

combinés reçus $(g^1, X^1), \dots, (g^s, X^s)$, où g^j est le vecteur de codage du $j^{\text{ème}}$ paquet combiné X^j . Après avoir choisi de façon aléatoire uniforme un ensemble de coefficients h_1, \dots, h_s , ce nœud peut générer un nouveau paquet combiné (g', X') en faisant la combinaison linéaire suivante :

$$X' = \sum_{j=1}^s h_j X^j$$

Le vecteur de codage correspondant g' est alors

$$g'_i = \sum_{j=1}^s h_j g_i^j$$

Cette opération peut être répétée dans plusieurs nœuds dans le réseau.

Au niveau du décodage, il est tout d'abord nécessaire que pour chaque paquet reçu, le récepteur connaisse le vecteur des combinaisons linéaires totales de ce vecteur en fonction des paquets sources. Il réalise ensuite la combinaison linéaire inverse (qui correspond à une inversion de matrice) pour retrouver les paquets sources.

Par exemple, supposons qu'un nœud reçoive l'ensemble de paquets $(g^1, X^1), \dots, (g^s, X^s)$. Pour récupérer les paquets sources M_1, \dots, M_n , il doit résoudre le système de s équations linéaires avec n inconnues :

$$\begin{cases} X^1 = \sum_{i=1}^n g_i^1 M_i \\ X^2 = \sum_{i=1}^n g_i^2 M_i \\ \dots \\ X^s = \sum_{i=1}^n g_i^s M_i \end{cases}$$

où les inconnues sont les paquets sources M_i . Si le nombre d'équations linéairement indépendantes est supérieur ou égal au nombre de paquets sources, le système d'équations peut être résolu et les n paquets sources, M_1, \dots, M_n , peuvent être récupérés.

Dans [33][6], il est montré que la probabilité que des équations linéaires soient indépendantes augmente avec la taille du corps fini. Lorsque la taille de corps est suffisamment grande, le codage réseau aléatoire permet d'atteindre le débit maximum entre la source et les récepteurs avec une grande probabilité de succès. Cette probabilité, qui dépend de la longueur du code, tend vers 1 lorsque la taille de corps 2^m tend vers l'infini.

Ces travaux ont donné une borne inférieure de la probabilité du succès pour des sources indépendantes dans un réseau acyclique. Cette borne peut être obtenue de la manière suivante. Considérons un graphe acyclique $G = (V, E)$ avec des sources indépendantes et un ensemble de d récepteurs. Considérons un code réseau aléatoire dans lequel les coefficients sont choisis de façon aléatoire uniforme sur un corps fini \mathbb{F}_q où $q > d$. La probabilité du succès avec laquelle tous les récepteurs peuvent décoder et récupérer les processus des sources est supérieure ou égale à

$$(1 - d/q)^n$$

Chapitre 2. État de l'art et notions de base

où η est le nombre total de liens qui reçoivent des combinaisons des paquets sources avec des coefficients aléatoires [33] [20] [35][6].

Même si il y a une certaine probabilité que les paquets combinés (ou les combinaisons) soient linéairement corrélés, une petite taille de corps (par exemple, $m = 8$) est suffisante pour que cette probabilité devienne négligeable [20] et que le codage aléatoire reste plus performant que le routage [36]. Une borne est également proposée pour des sources corrélées, des réseaux cycliques et des réseaux avec des délais de transmission [6].

Au niveau pratique, il faut noter que l'augmentant du nombre de paquets sources n codés ensemble nécessite une augmentation de la quantité de mémoire au niveau du récepteur. En effet, le récepteur doit stocker tous les paquets reçus jusqu'à ce qu'il reçoive au moins n paquets indépendants. En plus de la mémoire, la taille du bloc codé n a une conséquence sur le délai de décodage des paquets. Afin de réduire les besoins en mémoire, le délai et la complexité de calcul, la taille des matrices de décodage doit être limitée.

Pour plus de détails, les principaux avantages du codage réseau aléatoire comparés au routage classique ainsi que plusieurs exemples sont présentés dans [5][6][9][20][37][38][39].

2.2.3.4 Construction algébrique du codage réseau

Dans cette partie, une méthode de construction algébrique d'un code réseau introduite dans [2], [3] sera présentée. Cette présentation est détaillée car une de nos contributions est directement inspirée de cette construction.

Supposons qu'un réseau soit représenté par un graphe orienté sans cycles $G = (V, E)$ où V est l'ensemble de nœuds et E est l'ensemble des liens $\subseteq V * V$. Un lien orienté est noté par $e = (v, v')$ où v est appelé *init*(e) et v' est la *term*(e).

$\Gamma_I(v)$ et $\Gamma_O(v)$ sont définis comme :

- $\Gamma_I(v)$ est l'ensemble des liens qui entrent dans v .
- $\Gamma_O(v)$ est l'ensemble des liens qui sortent de v .

Et donc :

- $\Gamma_I(v) = \{e \in E : \text{init}(e) = v\}$.
- $\Gamma_O(v) = \{e \in E : \text{term}(e) = v\}$.

Le degré d'entrée $\delta_I(v)$ de v est défini comme : $\delta_I(v) = |\Gamma_I(v)|$. Le degré de sortie $\delta_O(v)$ est défini comme : $\delta_O(v) = |\Gamma_O(v)|$.

Soit $\chi(v) = \{X(v, 1), X(v, 2), \dots, X(v, \mu(v))\}$ un ensemble de $\mu(v)$ variables aléatoires discrètes obtenues en entrée du nœud v . Une connexion c est définie comme un triplet

$$(v, v', \chi(v, v')) \in V * V * \rho_{\chi(v)}$$

où $\rho_{\chi(v)}$ représente l'ensemble des puissances de $\chi(v)$. L'ensemble des connexions dans G est ℓ .

Pour chaque connexion $c = (v, v', \chi(v, v'))$ le débit $R(c)$ est défini comme :

$$R(c) = \sum_{i: X(v, i) \in \chi(v, v')} H(X(v, i))$$

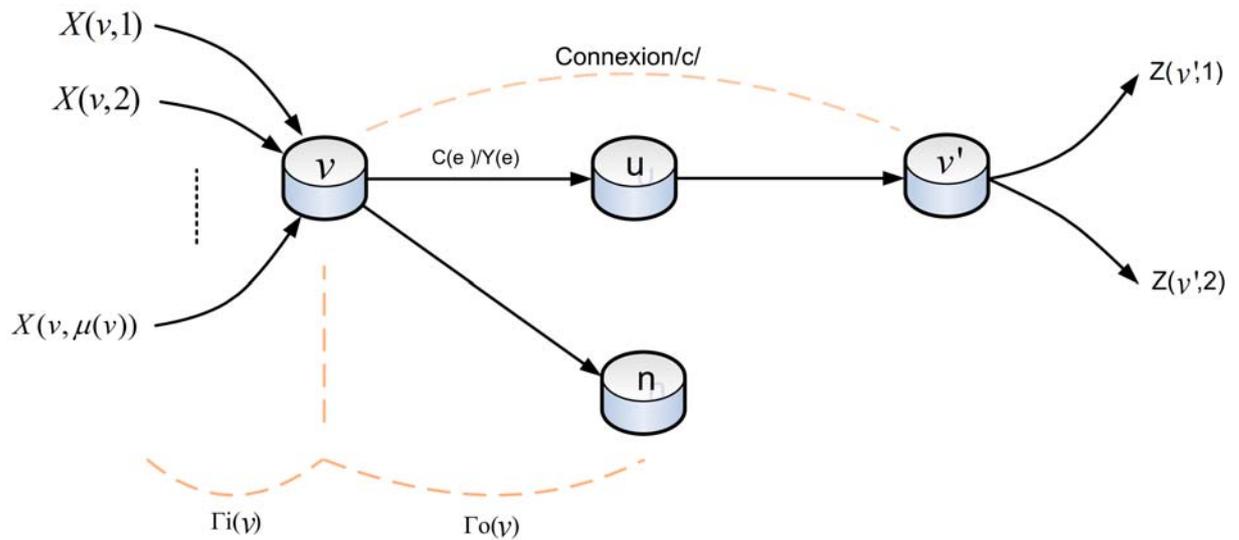


FIGURE 2.4 – Une connexion entre une source et un récepteur

où $H(X)$ est l'entropie de la variable aléatoire X . Pour un connexion $c = (v, v', \chi(v, v'))$: v est la source et v' est le récepteur.

Un nœud v peut émettre une information via un lien $e = (v, v')$ à destination de v' avec un débit maximum $C(e)$ où $C(e)$ est la capacité du lien e .

Les variables aléatoires représentant les informations transmises sur le lien e sont notées $Y(e)$, $e \in \Gamma_O(v)$. Il peut noter que d'autres informations sont transmises vers v , il s'agit de $Y(e')$ tel que $e' \in \Gamma_I(v)$.

En général, $Y(e)$, qui est transmis sur le lien $e = (v, u) \in \Gamma_O(v)$, est une fonction de $\chi(v)$ et $Y(e')$ si $e' \in \Gamma_I(v)$.

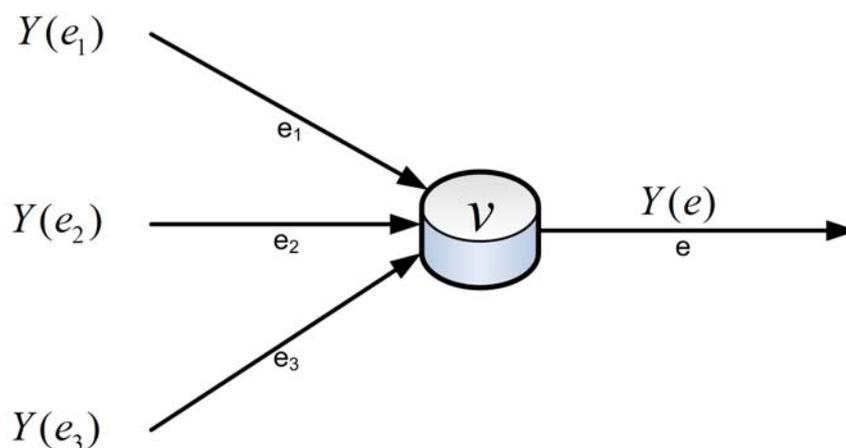


FIGURE 2.5 – Les processus dans un nœud intermédiaire

Si v est un récepteur, ses sorties sont représentées par un ensemble de $\nu(v)$ proces-

Chapitre 2. État de l'art et notions de base

sus aléatoires $Z(v) = \{Z(v, 1), Z(v, 2), \dots, Z(v, \nu(v))\}$.

Dans le cas où v n'est ni une source ni un récepteur, les différents processus aléatoires seront représentés comme indiqué sur la Figure 2.5.

Pour simplifier, les suppositions suivantes seront considérées :

- la capacité de chaque lien de G est constante.
- chaque lien dans le réseau de communication a le même délai de transmission.
- les processus aléatoires $X(v, l)$ où $l \in \{1, 2, \dots, \mu(v)\}$ sont indépendants et ont un taux d'entropie constant (m bits par unité de temps).
- les processus aléatoires $X(v, l)$ sont indépendants pour les différents v (les informations qui sont injectées dans le réseau à des positions différentes sont supposées indépendantes).

De plus, une communication dans le réseau est considérée comme une transmission d'un vecteurs de bits tels que les longueurs de ces vecteurs sont égales dans toutes les transmissions. Tous les liens sont supposés être synchronisés.

Chaque vecteur binaire transmis de longueur m peut être considéré comme un élément d'un corps fini contenant 2^m éléments, i.e. \mathbb{F}_{2^m} .

Les processus aléatoires $X(v, l)$, $Y(e)$ et $Z(v, l)$ sont traités comme des processus aléatoires discrets : $X(v, l) = \{X_0(v, l), X_1(v, l), \dots, X_m(v, l)\}$, $Y(e) = \{Y_0(e), Y_1(e), \dots\}$ et $Z(v, l) = \{Z_0(v, l), Z_1(v, l), \dots\}$. Ces processus génèrent une succession de symboles dans \mathbb{F}_{2^m} .

Soit $G = (V, E)$ un réseau de communication considéré sans délai. On peut dire que G est un réseau linéaire \mathbb{F}_{2^m} si pour tout lien e tel que $e = (v, u) \in E$, $Y(e)$ vérifie :

$$Y(e) = \sum_{l=1}^{\mu(v)} \alpha_{e,l} * X(v, l) + \sum_{e': \text{init}(e')=\text{term}(e)} \beta_{e',e} * Y(e')$$

où $\alpha_{e,l}$ et $\beta_{e',e}$ sont des éléments de \mathbb{F}_{2^m} .

Les sorties $Z(v, l)$ de n'importe quel nœud v sont une combinaison linéaire des $Y(e)$ où $e \in \Gamma_I(v)$:

$$Z(v, j) = \sum_{e': \text{init}(e')=v} \varepsilon_{e',j} * Y(e')$$

où $\varepsilon_{e',j} \in \mathbb{F}_{2^m}$.

Pour un réseau donné G et un ensemble donné de connexions ℓ , on définit le problème de réseau comme une paire (G, ℓ) . Ce problème donne des conditions algébriques sous lesquelles un ensemble de connexions est possible.

La solution de ce problème consiste à trouver les éléments $\alpha_{e,l}$, $\beta_{e',e}$ et $\varepsilon_{e',j}$ dans un corps fini convenable \mathbb{F}_{2^m} . S'il existe une solution au problème du codage réseau, donc le réseau est solvable.

La méthode de construction de la solution, présentée dans cette partie, relie le problème du codage réseau avec des concepts algébriques, avec le *théorème Min-cut Max-flow* ainsi qu'avec l'algorithme de *Ford-Fulkerson*.

Soit $G(V, E)$ un réseau avec une seule source et un seul récepteur. Dans ce cas, nous avons une seule connexion $c = (v, v', \chi(v, v'))$. Le problème de réseau est solvable si et

seulement si le débit de la connexion $R(c)$ est inférieur ou égal à la valeur minimale de toutes les coupes entre v et v' .

L'algorithme de *Ford- Fulkerson* donne une solution où tous les paramètres $\alpha_{e,l}$, $\beta_{e',e}$ et $\varepsilon_{e',j}$ sont égaux à 1 ou 0. Il donne une matrice de transfert qui définit la relation entre le vecteur d'entrée $\underline{x} = (X(v, 1), X(v, 2), \dots, X(v, \mu(v)))$ et les vecteurs de sortie

$$\begin{aligned} \underline{z} &= (Z(v', 1), Z(v', 2), \dots, Z(v', \nu(v))) \\ \underline{z} &= \underline{x} * M \end{aligned} \quad (2.1)$$

où M est une matrice dont les coefficients $(\alpha_{e,l}, \beta_{e',e}, \varepsilon_{e',j})$ sont des éléments de \mathbb{F}_2 . La matrice M est définie comme : $M = A * (I - F)^{-1} * B^T$ où les matrices A , B sont la matrice d'entrée et la matrice de sortie, définies comme suit :

Si le réseau contient plusieurs sources et récepteurs, le vecteur d'entrée est alors :

$$\begin{aligned} \underline{x} &= (\underline{x}_1, \underline{x}_2, \dots, \underline{x}_\mu) \\ &= (X(v_1, 1), X(v_1, 2), \dots, X(v_1, \mu(v_1)), X(v_2, 1), X(v_2, 2), \dots, X(v_{|V|}, \mu(v_{|V|}))) \end{aligned}$$

La matrice d'entrée A est de taille $\mu * |E|$ où $\mu = \sum_i \mu(v_i)$. Pour tout l , elle est définie de la manière suivante :

$$A_{i,j} = \begin{cases} \alpha_{e_j,l} & x_i = X\{term(e_j), l\} \\ 0 & \text{sinon} \end{cases}$$

De façon similaire, soit :

$$\begin{aligned} \underline{z} &= (z_1, z_2, \dots, z_\mu) \\ &= (Z(v_1, 1), Z(v_1, 2), \dots, Z(v_1, \nu(v_1)), Z(v_2, 1), Z(v_2, 2), \dots, X(v_{|V|}, \nu(v_{|V|}))) \end{aligned}$$

le vecteur de sortie. La matrice de sortie B est de taille $\nu * |E|$. Pour tout l , elle est définie comme :

$$B_{i,j} = \begin{cases} \varepsilon_{e_j,l} & z_i = Z\{init(e_j), l\} \\ 0 & \text{sinon} \end{cases}$$

Les éléments de la matrice F sont choisis tels que le déterminant de la matrice M ne soit pas zéro. Considérons le *lemme 1* et le *théorème 2* introduits dans [3], La matrice F est représentée comme une matrice adjacente de taille $|E| * |E|$. Elle est définie de la manière suivante :

$$F_{i,j} = \begin{cases} \beta_{e_i,e_j} & init(e_i) = term(e_j) \\ 0 & \text{sinon} \end{cases}$$

Il est clair que F est une matrice triangulaire strictement supérieure. La matrice $I - F$ est donc inversible (voir le *lemme 2* dans [3]). Cet inverse doit être défini dans le corps d'extension de \mathbb{F}_2 contenant tous les éléments $\beta_{e',e}$, c'est-à-dire dans $\mathbb{F}_2[\dots, \beta_{e',e}, \dots]$.

Considérons maintenant le réseau avec une source s et deux récepteurs z et y représentés dans l'exemple de la Figure 2.6 et appliquons la construction proposée.

$$\begin{aligned} Y(e_1) &= \alpha_{e_1,b_1} b_1 + \alpha_{e_1,b_2} b_2 \\ Y(e_2) &= \alpha_{e_2,b_1} b_1 + \alpha_{e_2,b_2} b_2 \end{aligned}$$

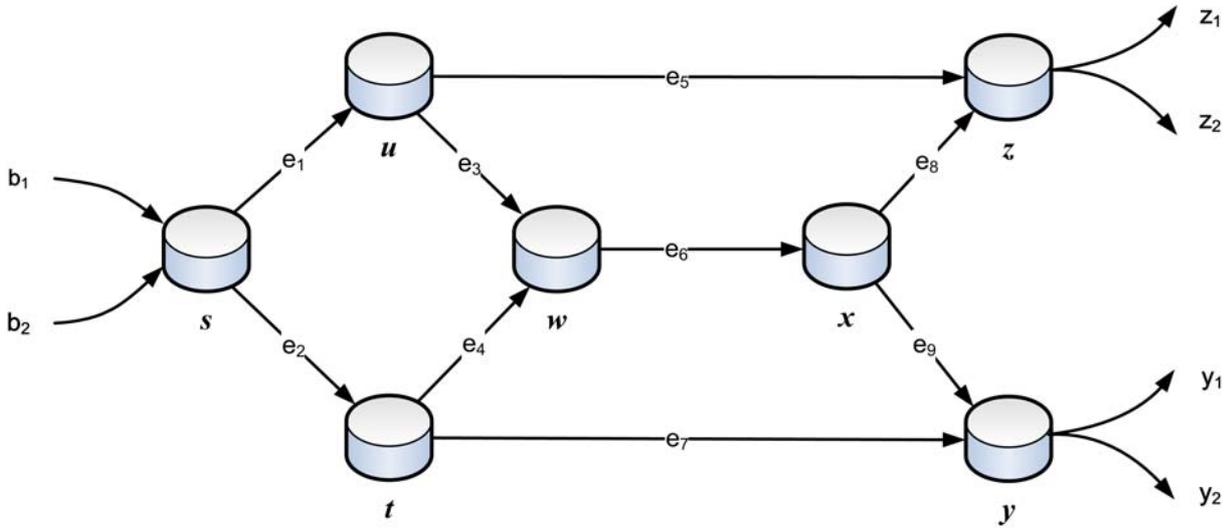


FIGURE 2.6 – Graphe de nœuds et de liens

$$\begin{bmatrix} Y(e_1) \\ Y(e_2) \end{bmatrix} = \begin{bmatrix} \alpha_{e_1, b_1} & \alpha_{e_1, b_2} \\ \alpha_{e_2, b_1} & \alpha_{e_2, b_2} \end{bmatrix} * \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$\begin{aligned} Y(e_3) &= \beta_{e_1, e_3} \cdot Y(e_1) \\ Y(e_5) &= \beta_{e_1, e_5} \cdot Y(e_1) \\ Y(e_4) &= \beta_{e_2, e_4} \cdot Y(e_2) \\ Y(e_7) &= \beta_{e_2, e_7} \cdot Y(e_2) \\ Y(e_6) &= \beta_{e_3, e_6} \cdot Y(e_3) + \beta_{e_4, e_6} \cdot Y(e_4) \\ Y(e_8) &= \beta_{e_6, e_8} \cdot Y(e_6) \\ Y(e_9) &= \beta_{e_6, e_9} \cdot Y(e_6) \\ Z_1 &= \varepsilon_{e_5, z_1} \cdot Y(e_5) + \varepsilon_{e_8, z_1} \cdot Y(e_8) \\ Z_2 &= \varepsilon_{e_5, z_2} \cdot Y(e_5) + \varepsilon_{e_8, z_2} \cdot Y(e_8) \\ Y_1 &= \varepsilon_{e_7, y_1} \cdot Y(e_7) + \varepsilon_{e_9, y_1} \cdot Y(e_9) \\ Y_2 &= \varepsilon_{e_7, y_2} \cdot Y(e_7) + \varepsilon_{e_9, y_2} \cdot Y(e_9) \end{aligned}$$

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} & e_5 & e_8 & e_7 & e_9 \\ \varepsilon_{e_5, Z_1} & \varepsilon_{e_8, Z_1} & 0 & 0 \\ \varepsilon_{e_5, Z_2} & \varepsilon_{e_8, Z_2} & 0 & 0 \\ 0 & 0 & \varepsilon_{e_7, Y_1} & \varepsilon_{e_9, Y_1} \\ 0 & 0 & \varepsilon_{e_7, Y_2} & \varepsilon_{e_9, Y_2} \end{bmatrix} \cdot \begin{bmatrix} Y(e_5) \\ Y(e_8) \\ Y(e_7) \\ Y(e_9) \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \cdot M$$

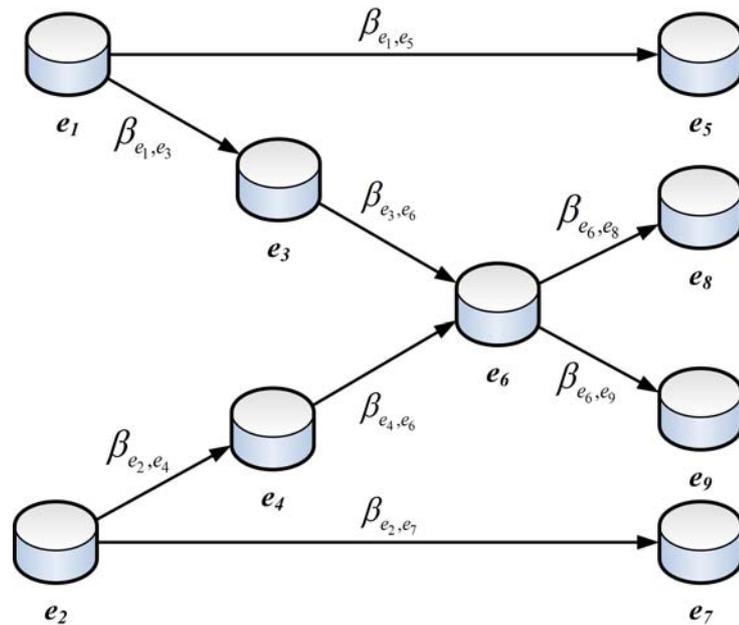


FIGURE 2.7 – Graphe d'étiquette des liens

Pour la Figure 2.7, la matrice F sera la suivante :

$$F = \begin{array}{c} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \end{array} \begin{array}{c} e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6 \quad e_7 \quad e_8 \quad e_9 \\ \left[\begin{array}{cccccccccc} 0 & 0 & \beta_{e_1,e_3} & 0 & \beta_{e_1,e_5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_{e_2,e_4} & 0 & 0 & \beta_{e_2,e_7} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_{e_3,e_6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_{e_4,e_6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_{e_6,e_8} & \beta_{e_6,e_9} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \end{array}$$

Et la matrice M est donnée par : $M = A * (I - F)^{-1} * B^T$ où les matrices A et B sont ainsi définies comme :

$$A = \begin{array}{c} b_1 \\ b_2 \end{array} \begin{array}{c} \left[\begin{array}{cccccccccc} \alpha_{e_1,b_1} & \alpha_{e_2,b_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha_{e_1,b_2} & \alpha_{e_2,b_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \end{array}$$

$$B = \begin{array}{c} Z_1 \\ Z_2 \\ Y_1 \\ Y_2 \end{array} \begin{array}{c} \left[\begin{array}{cccccccccc} 0 & 0 & 0 & 0 & \varepsilon_{e_5,Z_1} & 0 & 0 & \varepsilon_{e_8,Z_1} & 0 \\ 0 & 0 & 0 & 0 & \varepsilon_{e_5,Z_2} & 0 & 0 & \varepsilon_{e_8,Z_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \varepsilon_{e_7,Y_1} & 0 & \varepsilon_{e_9,Y_1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \varepsilon_{e_7,Y_2} & 0 & \varepsilon_{e_9,Y_2} \end{array} \right] \end{array}$$

Intuitivement, la matrice A peut être vue comme le codage avant l'entrée dans le réseau. De même, la matrice B peut être vue comme le décodage des informations qui

Chapitre 2. État de l'art et notions de base

parviennent à chacun des récepteurs. L'inverse de la matrice $I - F$ peut être calculé de la manière suivante :

$$(I - F)^{-1} = (I + F + F^2 + F^3 + F^4 + \dots)$$

Comme F est une matrice triangulaire strictement supérieure, elle est nilpotente. Ceci permet d'obtenir :

$$(I - F)^{-1} = \begin{bmatrix} 1 & 0 & \beta_{e_1,e_3} & 0 & \beta_{e_1,e_5} & \beta_{e_1,e_3}\beta_{e_3,e_6} & 0 & \beta_{e_1,e_3}\beta_{e_3,e_6}\beta_{e_6,e_8} & \beta_{e_1,e_3}\beta_{e_3,e_6}\beta_{e_6,e_9} \\ 0 & 1 & 0 & \beta_{e_2,e_4} & 0 & \beta_{e_2,e_4}\beta_{e_4,e_6} & \beta_{e_2,e_7} & \beta_{e_2,e_4}\beta_{e_4,e_6}\beta_{e_6,e_8} & \beta_{e_2,e_4}\beta_{e_4,e_6}\beta_{e_6,e_9} \\ 0 & 0 & 1 & 0 & 0 & \beta_{e_3,e_6} & 0 & \beta_{e_3,e_6}\beta_{e_6,e_8} & \beta_{e_3,e_6}\beta_{e_6,e_9} \\ 0 & 0 & 0 & 1 & 0 & \beta_{e_4,e_6} & 0 & \beta_{e_4,e_6}\beta_{e_6,e_8} & \beta_{e_4,e_6}\beta_{e_6,e_9} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \beta_{e_6,e_8} & \beta_{e_6,e_9} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Ensuite, la matrice $A(I - F)^{-1}$ sera :

$$A(I - F)^{-1} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} & S_{17} & S_{18} & S_{19} \\ S_{21} & S_{22} & S_{23} & S_{24} & S_{25} & S_{26} & S_{27} & S_{28} & S_{29} \end{bmatrix}$$

Où :

$$\begin{aligned} S_{11} &= \alpha_{e_1,b_1} \\ S_{21} &= \alpha_{e_1,b_2} \\ S_{12} &= \alpha_{e_2,b_1} \\ S_{22} &= \alpha_{e_2,b_2} \\ S_{13} &= \alpha_{e_1,b_1}\beta_{e_1,e_3} \\ S_{23} &= \alpha_{e_1,b_2}\beta_{e_1,e_3} \\ S_{14} &= \alpha_{e_2,b_1}\beta_{e_2,e_4} \\ S_{24} &= \alpha_{e_2,b_2}\beta_{e_2,e_4} \\ S_{15} &= \alpha_{e_1,b_1}\beta_{e_1,e_5} \\ S_{25} &= \alpha_{e_1,b_2}\beta_{e_1,e_5} \\ S_{16} &= \alpha_{e_1,b_1}\beta_{e_1,e_3}\beta_{e_3,e_6} + \alpha_{e_2,b_1}\beta_{e_2,e_4}\beta_{e_4,e_6} \\ S_{26} &= \alpha_{e_1,b_2}\beta_{e_1,e_3}\beta_{e_3,e_6} + \alpha_{e_1,b_2}\beta_{e_2,e_4}\beta_{e_4,e_6} \\ S_{17} &= \alpha_{e_2,b_1}\beta_{e_2,e_7} \\ S_{27} &= \alpha_{e_2,b_2}\beta_{e_2,e_7} \\ S_{18} &= \alpha_{e_1,b_1}\beta_{e_1,e_3}\beta_{e_3,e_6}\beta_{e_6,e_8} + \alpha_{e_2,b_1}\beta_{e_2,e_4}\beta_{e_4,e_6}\beta_{e_6,e_8} \\ S_{28} &= \alpha_{e_1,b_2}\beta_{e_1,e_3}\beta_{e_3,e_6}\beta_{e_6,e_8} + \alpha_{e_2,b_2}\beta_{e_2,e_4}\beta_{e_4,e_6}\beta_{e_6,e_8} \\ S_{19} &= \alpha_{e_1,b_1}\beta_{e_1,e_3}\beta_{e_3,e_6}\beta_{e_6,e_9} + \alpha_{e_2,b_1}\beta_{e_2,e_4}\beta_{e_4,e_6}\beta_{e_6,e_9} \\ S_{29} &= \alpha_{e_1,b_2}\beta_{e_1,e_3}\beta_{e_3,e_6}\beta_{e_6,e_9} + \alpha_{e_2,b_2}\beta_{e_2,e_4}\beta_{e_4,e_6}\beta_{e_6,e_9} \end{aligned}$$

La matrice B^T est ainsi obtenue :

$$B^T = \begin{array}{c} \begin{array}{cccc} Z_1 & Z_2 & Y_1 & Y_2 \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \varepsilon_{e_5, Z_1} & \varepsilon_{e_5, Z_2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon_{e_7, Y_1} & \varepsilon_{e_7, Y_2} \\ \varepsilon_{e_8, Z_1} & \varepsilon_{e_8, Z_2} & 0 & 0 \\ 0 & 0 & \varepsilon_{e_9, Y_1} & \varepsilon_{e_9, Y_2} \end{array} \right] \end{array}$$

La matrice M vaut donc :

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \end{bmatrix}$$

Où :

$$\begin{aligned} M_{11} &= \varepsilon_{e_5, Z_1} \alpha_{e_1, b_1} \beta_{e_1, e_5} + \varepsilon_{e_8, Z_1} \alpha_{e_1, b_1} \beta_{e_1, e_3} \beta_{e_3, e_6} \beta_{e_6, e_8} + \varepsilon_{e_8, Z_1} \alpha_{e_2, b_1} \beta_{e_2, e_4} \beta_{e_4, e_6} \beta_{e_6, e_8} \\ M_{21} &= \varepsilon_{e_5, Z_1} \alpha_{e_1, b_2} \beta_{e_1, e_5} + \varepsilon_{e_8, Z_1} \alpha_{e_1, b_2} \beta_{e_1, e_3} \beta_{e_3, e_6} \beta_{e_6, e_8} + \varepsilon_{e_8, Z_1} \alpha_{e_2, b_2} \beta_{e_2, e_4} \beta_{e_4, e_6} \beta_{e_6, e_8} \\ M_{12} &= \varepsilon_{e_5, Z_2} \alpha_{e_1, b_1} \beta_{e_1, e_5} + \varepsilon_{e_8, Z_2} \alpha_{e_1, b_1} \beta_{e_1, e_3} \beta_{e_3, e_6} \beta_{e_6, e_8} + \varepsilon_{e_8, Z_2} \alpha_{e_2, b_1} \beta_{e_2, e_4} \beta_{e_4, e_6} \beta_{e_6, e_8} \\ M_{22} &= \varepsilon_{e_5, Z_2} \alpha_{e_1, b_2} \beta_{e_1, e_5} + \varepsilon_{e_8, Z_2} \alpha_{e_1, b_2} \beta_{e_1, e_3} \beta_{e_3, e_6} \beta_{e_6, e_8} + \varepsilon_{e_8, Z_2} \alpha_{e_2, b_2} \beta_{e_2, e_4} \beta_{e_4, e_6} \beta_{e_6, e_8} \\ M_{13} &= \varepsilon_{e_7, Y_1} \alpha_{e_2, b_1} \beta_{e_2, e_7} + \varepsilon_{e_9, Y_1} \alpha_{e_1, b_1} \beta_{e_1, e_3} \beta_{e_3, e_6} \beta_{e_6, e_9} + \varepsilon_{e_9, Y_1} \alpha_{e_2, b_1} \beta_{e_2, e_4} \beta_{e_4, e_6} \beta_{e_6, e_9} \\ M_{23} &= \varepsilon_{e_7, Y_1} \alpha_{e_2, b_2} \beta_{e_2, e_7} + \varepsilon_{e_9, Y_1} \alpha_{e_1, b_2} \beta_{e_1, e_3} \beta_{e_3, e_6} \beta_{e_6, e_9} + \varepsilon_{e_9, Y_1} \alpha_{e_2, b_2} \beta_{e_2, e_4} \beta_{e_4, e_6} \beta_{e_6, e_9} \\ M_{14} &= \varepsilon_{e_7, Y_2} \alpha_{e_2, b_1} \beta_{e_2, e_7} + \varepsilon_{e_9, Y_2} \alpha_{e_1, b_1} \beta_{e_1, e_3} \beta_{e_3, e_6} \beta_{e_6, e_9} + \varepsilon_{e_9, Y_2} \alpha_{e_2, b_1} \beta_{e_2, e_4} \beta_{e_4, e_6} \beta_{e_6, e_9} \\ M_{24} &= \varepsilon_{e_7, Y_2} \alpha_{e_2, b_2} \beta_{e_2, e_7} + \varepsilon_{e_9, Y_2} \alpha_{e_1, b_2} \beta_{e_1, e_3} \beta_{e_3, e_6} \beta_{e_6, e_9} + \varepsilon_{e_9, Y_2} \alpha_{e_2, b_2} \beta_{e_2, e_4} \beta_{e_4, e_6} \beta_{e_6, e_9} \end{aligned}$$

Finalement, les flux sortants de ce réseau seront exprimés en fonction des flux entrants et de la matrice de transfert M comme suit :

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \varepsilon_{e_5, Z_1} & \varepsilon_{e_8, Z_1} & 0 & 0 \\ \varepsilon_{e_5, Z_2} & \varepsilon_{e_8, Z_2} & 0 & 0 \\ 0 & 0 & \varepsilon_{e_7, Y_1} & \varepsilon_{e_9, Y_1} \\ 0 & 0 & \varepsilon_{e_7, Y_2} & \varepsilon_{e_9, Y_2} \end{bmatrix} * \begin{bmatrix} Y(e_5) \\ Y(e_8) \\ Y(e_7) \\ Y(e_9) \end{bmatrix} = [b_1 \quad b_2] * M$$

Remarque : il faut noter que la construction algébrique du codage réseau présentée précédemment est un cas particulier où les délais et les cycles n'existent pas. Dans le cas général, où les délais sont considérés et où les cycles sont permis, la construction algébrique est remplacée par une construction des matrices basées sur des polynômes de $\mathbb{F}_2(D)[\dots, \alpha_{e,l}, \dots, \beta_{e',e}, \dots, \varepsilon_{e',j}, \dots]$ [2]. Les coefficients de ces polynômes sont des

Chapitre 2. État de l'art et notions de base

séries entières d'un délai variable D . Par exemple, la même information envoyée par une source peut être retardée de façon différente selon le chemin emprunté.

Soit $G = (V, E)$ un réseau de communication avec des délais de transmission. Dans ce cas, les processus aléatoires $X(v, i)$, $Z(u, l)$ et $Y(e)$, transmis sur un lien e , sont représentés par des séries entières en fonctions du délai D : $X(v, j)(D) = \sum_{l=0}^{\infty} X_l(v, j)D^l$, $Z(v, j)(D) = \sum_{l=0}^{\infty} Z_l(v, j)D^l$ et $Y(e)(D) = \sum_{l=0}^{\infty} Y_l(e)D^l$. Les équations du codage linéaire dans ce cas sont

$$Y_{t+1}(e) = \sum_{l=1}^{\mu(v)} \alpha_{e,l} * X_t(v, l) + \sum_{e': \text{init}(e')=\text{term}(e)} \beta_{e',e} * Y_t(e')$$

où $\alpha_{e,l}$ et $\beta_{e',e}$ sont des éléments de $\bar{\mathbb{F}}$ qui est défini comme la clôture algébrique de \mathbb{F}_2 . Les sorties $Z(v, l)$ du nœud v sont :

$$Z_{t+1}(v, j) = \sum_{l=t-m(v)}^t \varepsilon'_{j,t-l} * Z_l(v, j) + \sum_{l=t-m(v)}^t \sum_{e': \text{init}(e')=v} \varepsilon''_{e',j,t-l} * Y_l(e')$$

où $\varepsilon_{e',j,l}$ et $\varepsilon''_{e',j,l} \in \bar{\mathbb{F}}$ et $m(v)$ représente la mémoire nécessaire dans le récepteur v .

De façon similaire, les matrices A et B peuvent être représentées comme des matrices fonctions de D : $A(D)$ et $B(D)$. La matrice $(I - F)$, pour un réseau acyclique sans délai et qui est inversible car F est nilpotente, devient $(I - DF)$. Pour un réseau acyclique avec délai, cette matrice est inversible dans $\mathbb{F}_2[D, \dots, \beta_{e_i, e_j}, \dots]$ car son déterminant est un polynôme en D non nul.

On obtient alors la matrice de transfert correspondante : $M = A(D)(I - DF)^{-1}B(D)^T$.

2.3 Bénéfices et intérêts de codage réseau

Les avantages du codage réseau ont été présentés dans plusieurs contextes. Les auteurs de [1] ont prouvé qu'un gain de débit et de bande passante peut être obtenu en employant le codage réseau au lieu du routage classique. Dans [40], deux évaluations des avantages du codage réseau sont représentées de la manière suivante :

$$\text{Gain du codage réseau} = \frac{\text{Capacité du codage réseau}}{\text{Capacité du routage réseau}} - 1$$

$$\text{Économie de bande passante} = 1 - \frac{\text{Bande passante totale requise avec codage réseau}}{\text{Bande passante totale requise avec routage seulement}}$$

Le codage de réseau peut donc permettre à économiser la bande passante grâce au codage des informations [41].

Certains avantages du codage réseau sont illustrés dans un exemple donné dans [41] où l'on peut voir que le taux de transmission multicast dans le cas du codage réseau est considérablement plus grand que le taux de transmission multicast du cas du routage traditionnel. L'utilisation du codage réseau permet d'équilibrer la charge du trafic dans le réseau tout en la distribuant aux multiples liens.

Le codage réseau est non seulement employé pour économiser la bande passante et augmenter le débit mais il peut aussi être utile pour la robustesse du réseau, notamment lorsque les liens dans les réseaux sont défaillants, comme par exemple dans les réseaux sans fil.

Un autre intérêt du codage réseau dans les réseaux sans fil est la possibilité de réduire la quantité d'énergie par bit, ou en d'autres termes, la possibilité de réduire l'utilisation des ressources du réseau par rapport aux solutions de routage classiques [22][42][43].

De plus, l'avantage de codage réseau pour accéder et stocker des grands fichiers dans des réseaux pair-à-pair a été présenté dans [21]. Il est montré que le codage réseau peut obtenir un gain de l'ordre de 20 – 30% comparé avec l'utilisation de codes à effacement et 2 – 3 fois mieux comparé à la transmission d'informations non codées.

Dans les prochaines sections, nous détaillerons certains des principaux avantages du codage réseau.

2.3.1 Augmentation de la capacité multicast et du débit

L'augmentation de la capacité multicast a déjà été présentée dans les parties précédentes. Le point particulièrement important est que, avec un codage réseau optimal, chaque récepteur peut atteindre son débit maximum.

Cet avantage peut être exprimé sous plusieurs formes. En effet, comme le codage réseau permet d'améliorer l'utilisation des liens, il permet aussi d'augmenter le débit moyen observé par les différents récepteurs.

La Figure 2.8 est un exemple de réseau où le codage réseau permet de réduire le temps nécessaire pour acheminer les informations a et b de la source (point en vert) vers les 3 récepteurs (les restes des points).

En effet, avec un routage classique, le délai nécessaire est de 3 unités de temps (on considère que le temps de transfert entre 2 nœuds est toujours d'une unité de temps).

Par contre, avec le codage réseau, ce délai peut être ramené à 2 unités de temps (comme indiqué sur la Figure 2.8(b)).

2.3.2 Diminution de l'énergie par bit

On peut voir cet avantage dans le contexte des réseaux sans fil ou ad hoc. Cet avantage est illustré sur la Figure 2.9 où l'on suppose que chaque nœud diffuse l'information qu'il veut transmettre. Sans codage réseau, on peut voir sur la Figure 2.9(a) que 5 transmissions sont nécessaires pour transmettre un bit vers les 2 récepteurs alors qu'avec le codage réseau (Figure 2.9(b)) seulement 9 transmissions sont nécessaires pour transmettre 2 bits [17].

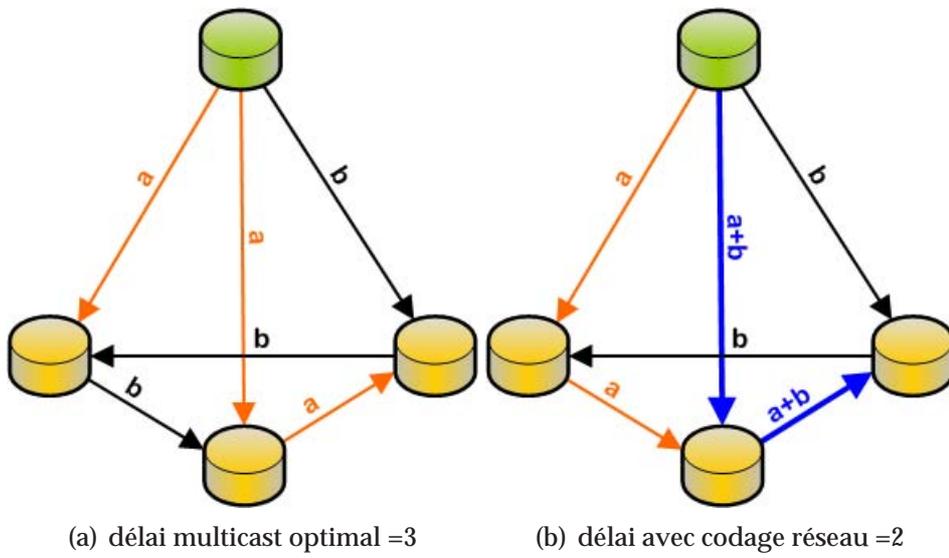


FIGURE 2.8 – Le délai de réception des nœuds récepteurs

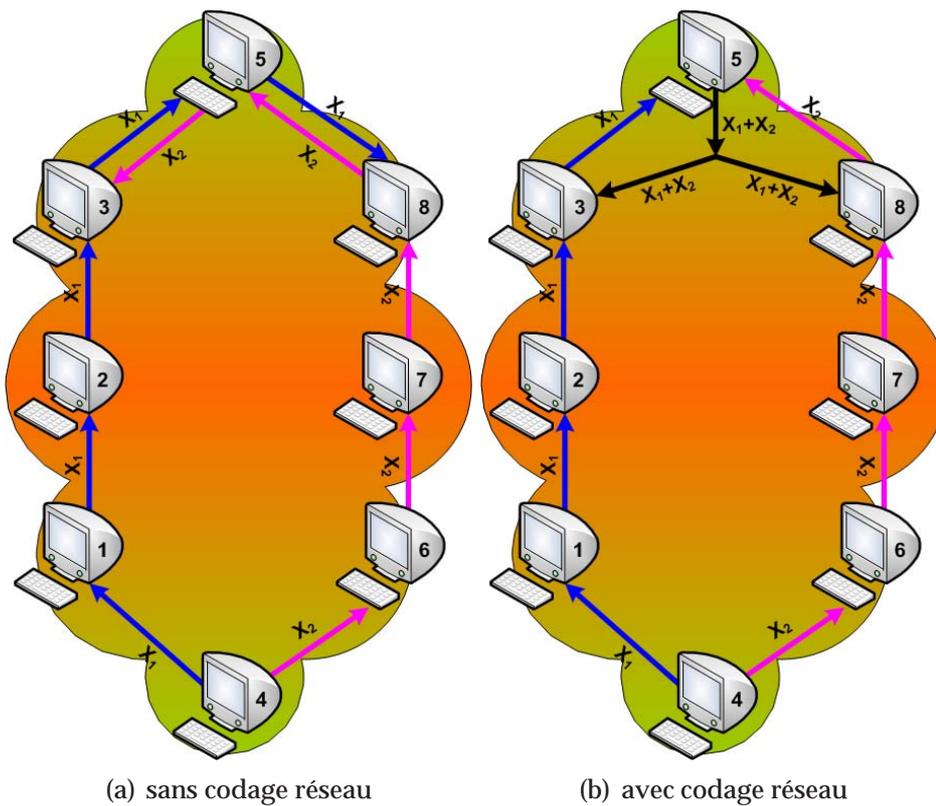


FIGURE 2.9 – Économie d'énergie grâce au codage réseau sur un exemple de réseau ad hoc sans fil

2.3. Bénéfices et intérêts de codage réseau

Le décodage des données dans un récepteur est possible quand ce récepteur récupère un nombre suffisant de paquets codés. La nouveauté du codage réseau par rapport au codage à effacement traditionnel que le codage des données ne se fait pas seulement au niveau de sources mais aussi dans tous les routeurs du réseau. Considérons la

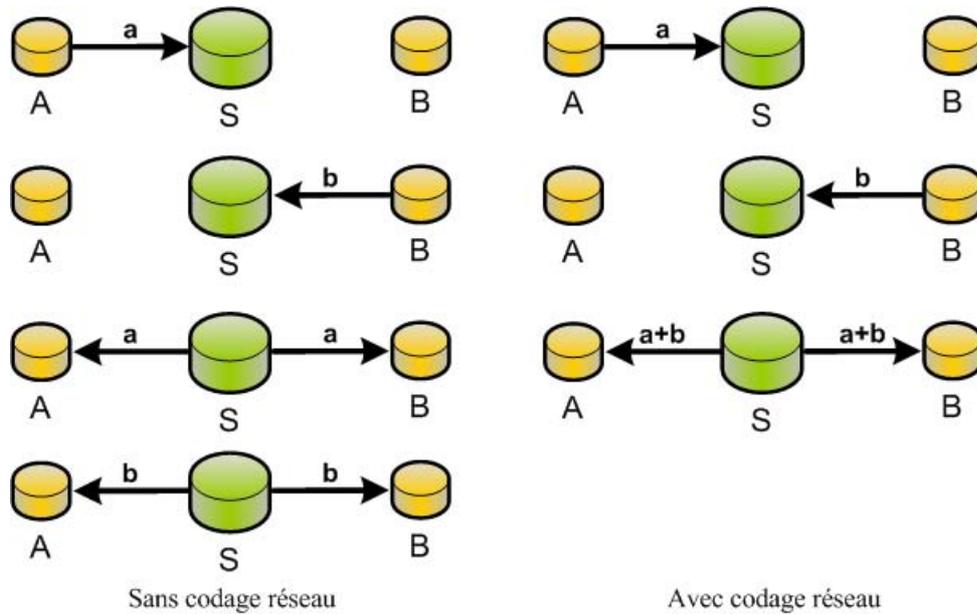


FIGURE 2.10 – Exemple de l'intérêt du codage réseau

Figure 2.10, où les nœuds A et B veulent échanger leurs paquets via un nœud intermédiaire S . Chaque nœud envoie un paquet vers la station S . D'après le codage réseau, la station S diffuse la combinaison linéaire des paquets au lieu de diffuser chaque paquet reçu vers les nœuds. Finalement chaque nœud peut récupérer les informations qui lui manquent, ce qui permet de diminuer le nombre total de transmissions.

2.3.3 Robustesse et tolérance aux fautes

On peut considérer que la robustesse et l'adaptabilité font partie des principaux avantages du codage réseau. L'utilisation de codage réseau dans certains contextes, comme dans les réseaux sans fil où les pertes des paquets ou de liens sont courantes, peut offrir plusieurs avantages.

Dans ce domaine, la technique FEC (Forward Error Correction) [44], qui protège les transmissions contre les erreurs et les pertes de paquets, peut être utilisée. Cette technique ne nécessite pas de ré-émission des données, contrairement aux techniques classiques (comme la demande automatique de répétition (ARQ)), si le pourcentage d'erreurs ou de pertes est inférieur à un seuil prédéterminé.

L'émetteur ajoute de la redondance afin de permettre au destinataire de détecter et de corriger une partie des erreurs. Cela permet d'éviter la retransmission, et donc de

faire des économies de bande, voire d'assurer la transmission dans certaines situations où il n'y a pas de voie de retour [15][34][45].

Cette technique s'applique surtout à des types de données ne pouvant pas supporter les ré-émissions des données et les retards, telles que la voix ou la vidéo.

Malgré l'apparition des nouveaux types de codes (tels que les codes "Fountain" [46][47]) bien adaptés à ces types d'application, les codes correcteurs d'erreurs ou d'effacement restent des mécanismes de bout-en-bout (codage à la source et décodage aux récepteurs) sans codage dans les nœuds intermédiaires.

Ces nœuds intermédiaires sont autorisés à faire des codages sur les informations reçus lorsque le codage réseau est appliqué. Cela permet d'apporter des améliorations incontestables. Par exemple, considérons la transmission d'information entre une source A et une destination B via un routeur S qui est capable d'effectuer des opérations de codage réseau sur ses informations reçues.

Avec la technique FEC de bout-en-bout, B peut décoder les informations reçues avec un taux $T_1 = (1 - \epsilon_{AS})(1 - \epsilon_{SB})$ où ϵ_{ij} est la probabilité qu'un paquet soit perdu sur le lien entre i et j . Tandis que si on autorise le routeur S à décoder et à re-coder l'information avant de l'envoyer vers B on obtient un débit optimal égal à : $T_2 = \min\{(1 - \epsilon_{AS}), (1 - \epsilon_{SB})\}$ [15][34][45].

Clairement, il faut intégrer le fait que les codages et les décodages dans les nœuds intermédiaires peuvent causer des délais supplémentaires parce qu'il faut attendre que le nombre suffisant de paquets soit reçu avant de procéder à l'opération de décodages/re-codages.

Pour contourner ce problème, Lun et al. [48] ont proposé un schéma de codage linéaire aléatoire qui atteint la capacité pour des réseaux de communication par paquets, unicast ou multicast, avec pertes sur tout canal où les paquets arrivent selon un processus admettant un taux d'arrivée moyen.

Dans ce schéma, les nœuds intermédiaires réalisent des opérations de codage linéaire aléatoire sans attendre tous les paquets d'un bloc. Un des résultats supplémentaires de ce papier est l'expression du taux de décroissance de la probabilité d'erreur en fonction du délai de codage.

2.4 Applications de codage réseau dans différents domaines et types de réseaux

Tous les problèmes évoqués précédemment font que l'application du codage réseau à des réseaux réels n'est pas immédiate dans la plupart des cas. Toutefois, de plus en plus de travaux sont proposés pour appliquer le codage réseau dans des contextes réels.

2.4.1 Internet

Dans le domaine de l'Internet, deux modèles ont été proposés pour utiliser le codage réseau. Ces deux modèles ont été présentés dans [5].

2.4. Applications de codage réseau dans différents domaines et types de réseaux

2.4.1.1 Modèle de paquet

Considérons un graphe acyclique (V, E) avec une unité de capacité pour chaque lien, une source $s \in V$, et une ensemble de récepteurs $T \subseteq E$.

La **capacité de diffusion** h est le nombre minimum de liens dans chaque coupe entre la source et un récepteur. Chaque lien $e \in E$ qui sort d'un nœud $v = in(e)$ porte un symbole $y(e)$ qui est une combinaison linéaire des symboles $y(e')$ sur les liens entrants e' :

$$y(e) = \sum_{e': out(e')=v} m_e(e')y(e')$$

Le vecteur de codage local $m(e) = [m_e(e')]_{e': out(e')=v}$ représente la fonction de codage au nœud v sur un lien e . Si v est la source s , on rajoute des liens artificiels e'_1, \dots, e'_h qui entrent dans s en portant h symboles source $y(e') = x_i, i = 1, \dots, h$. La valeur $y(e)$ transmise sur chaque lien $e \in E$ est la combinaison linéaire $y(e) = \sum_{i=1}^h g_i(e)x_i$ de ces symboles source.

Le vecteur de h coefficients $g(e) = [g_1(e), \dots, g_h(e)]$ peut être déterminé par

$$g(e) = \sum_{e': out(e')=v} m_e(e')g(e')$$

Le vecteur $g(e)$ est appelé le vecteur de codage global sur un lien e .

Chaque récepteur t reçoit sur ses h (ou plus) liens entrants e_1, \dots, e_h les symboles :

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}$$

On peut récupérer les symboles source x_1, \dots, x_h via la matrice G_t des vecteurs de

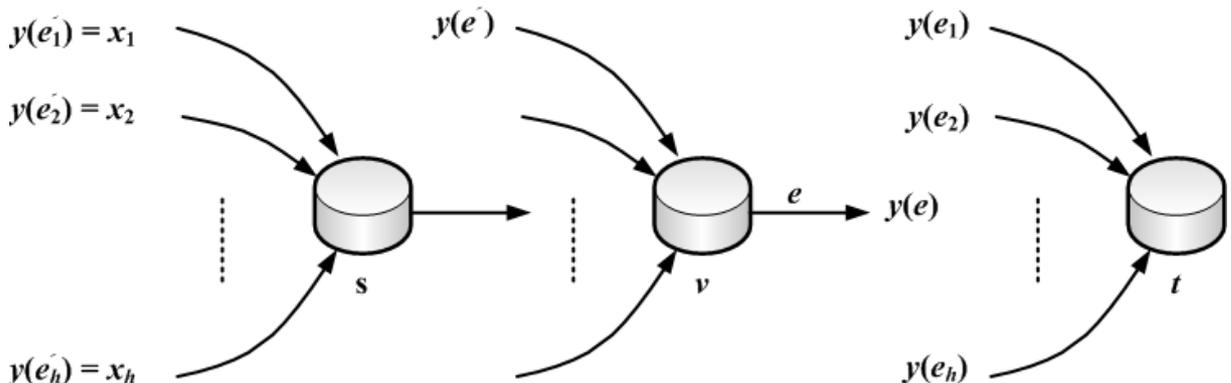


FIGURE 2.11 – Symboles reçus par un récepteur

codage global $g(e_1), \dots, g(e_h)$.

Lorsque les vecteurs de codage locaux sont générés aléatoirement, cette récupération peut être réalisée avec une grande probabilité si les symboles appartiennent à un

Chapitre 2. État de l'art et notions de base

corps fini suffisamment grand. Pratiquement, dans un réseau de paquets, les symboles $y(e)$ transmis sur un lien e sont divisés en paquets et une taille de corps égale 2^8 est suffisante. Classiquement, dans l'Internet, la taille maximale d'un paquet sans l'entête est légèrement supérieure à 1400 octets. Chaque paquet IP peut donc porter à peu près $N = 1400$ symboles si la taille du corps est 2^8 ou environ $N = 700$ si la taille du corps est 2^{16} . Ainsi, les symboles $y(e)$ transmis sur chaque lien e sont regroupés en vecteurs :

$$y(e) = [y_1(e), y_2(e), \dots, y_N(e)]$$

Chaque vecteur est alors considéré comme une combinaison linéaire

$$y(e) = \sum_{e': out(e')=v} m_e(e') y(e')$$

des vecteurs $y(e')$ sur les liens e' entrants $v = in(e)$. De la même manière, les symboles de la source x_i transmis vers la source sur les liens artificiels e'_i sont groupés en vecteurs $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,N}]$ et donc chaque récepteur peut retrouver (avec une grande probabilité) les h vecteurs de la source x_1, \dots, x_h à partir de n'importe quels h paquets reçus.

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} y_1(e_1) & y_2(e_1) & \cdots & y_N(e_1) \\ \vdots & \vdots & \ddots & \vdots \\ y_1(e_h) & y_2(e_h) & \cdots & y_N(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{h,1} & x_{h,2} & \cdots & x_{h,N} \end{bmatrix}$$

Chaque paquet contient un vecteur de dimension h représentant le vecteur de codage global $g(e)$. Chaque récepteur peut récupérer les vecteurs de la source x_1, \dots, x_h en réalisant une **élimination de Gauss** sur les vecteurs de codage des h paquets entrants.

$$\begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) & y_1(e_1) & y_2(e_1) & \cdots & y_N(e_1) \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) & y_1(e_h) & y_2(e_h) & \cdots & y_N(e_h) \end{bmatrix} = G_t \begin{bmatrix} 1 & \cdots & 0 & x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & x_{h,1} & x_{h,2} & \cdots & x_{h,N} \end{bmatrix}$$

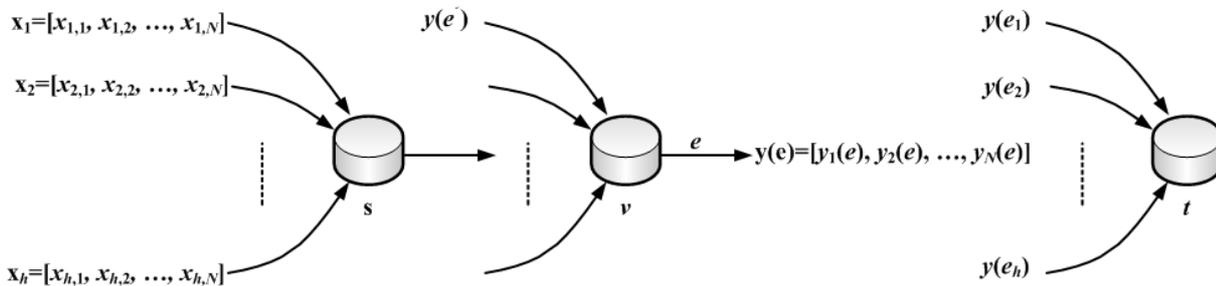


FIGURE 2.12 – Chaque récepteur reçoit une combinaison de vecteurs de la source

Le point négatif de ce schéma est la transmission des h symboles représentant le codage dans chaque paquet. Toutefois, les avantages de ce schéma sont très importants :

2.4. Applications de codage réseau dans différents domaines et types de réseaux

les récepteurs peuvent faire le décodage sans connaître la topologie du réseau ou des fonctions de codage même si des nœuds des liens sont ajoutés ou supprimés. Les récepteurs peuvent décoder malgré les pertes de certains paquets. Notons que le décodage est réalisable même si les vecteurs de codage local ont été choisis aléatoirement.

A cause des effacements et des pertes, la valeur de la coupe minimale pour n'importe quel nœud est inférieure à h . Pour cela on a besoin d'une protection contre les effacements. La nécessité de cette technique vient de que, même dans les conditions idéales la source ne connaît exactement la capacité de diffusion. Une solution qui peut être utilisée dans ce cas-là est la technique de transmission avec une protection inégale des données (*PET* : Priority Encoding Transmission [49]). Cette technique est un schéma de protection contre les effacements dans lequel les vecteurs de la source sont divisés en h couches d'importance croissante. Les couches ont un degré de protection proportionnel à leur importance.

2.4.1.2 Modèle de file d'attente

Dans les réseaux réels, les paquets ne sont pas synchronisés. Sur des liens de capacités différentes, les paquets sont transmis séquentiellement avec d'autres paquets qui sont liés aux mêmes vecteurs de la source x_1, \dots, x_h sur le même lien. Les paquets sur des liens différents sont en général sujets à des capacités différentes. De même, le nombre de paquets transmis sur un lien qui est lié à un même ensemble de s vecteurs de la source sont généralement variables à cause des pertes de paquets ou des congestions.

Pour toutes ces raisons, le modèle de file d'attente a été proposé .

Tous les paquets constitués à partir du même ensemble de h vecteurs de la source x_1, \dots, x_h sont considérés de la même génération. Le nombre h représente la taille de la génération. Tous les paquets de la même génération sont marqués avec le même indice de génération. Les générations successives reçoivent des indices de génération croissants successifs.

Un ou deux octets (ce qui implique des incréments modulo 2^8 ou 2^{16}) dans chaque entête de paquet est suffisant pour faire la distinction entre les générations successives dans le réseau. En plus de l'indice de la génération dans l'entête du paquet, un autre mécanisme est utilisé par chaque nœud pour synchroniser les paquets entrants et sortants, ce mécanisme est appelé le "buffering".

Avec ce mécanisme, les paquets qui arrivent dans un nœud sur n'importe quel lien entrant sont mis dans une file d'attente et classés suivant l'indice de la génération. Puis, quand se présente une occasion de transmission sur un lien sortant, un paquet est généré par une combinaison linéaire aléatoire de tous les paquets de la génération courante qui sont prêts et existant dans la file d'attente. Périodiquement, une nouvelle génération courante est choisie et les paquets correspondant à l'ancienne génération courante sont supprimés. Les combinaisons linéaires des paquets sont choisies aléatoirement pour chaque paquet sortant, et varient donc dans le temps. Toutefois, ceci ne présente pas de nouveaux problèmes au décodeur car les vecteurs de codage global sont inclus dans les paquets [5][50].

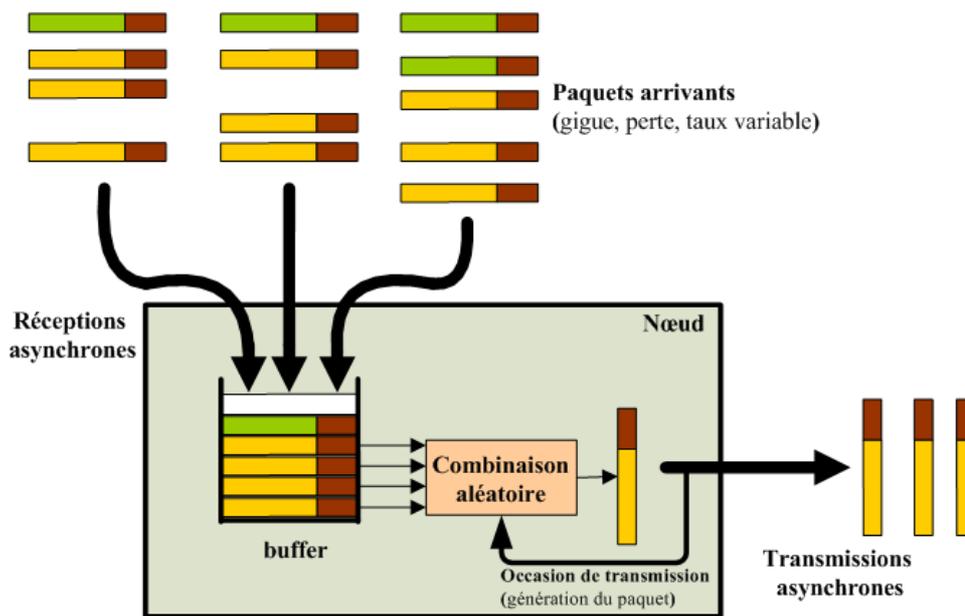


FIGURE 2.13 – Principe du buffering

2.4.2 Réseaux de recouvrement "Overlay"

L'application du codage réseau dans les réseaux overlays (réseaux de recouvrements) a été proposée dans [4]. Ce paragraphe en présente les principaux résultats.

Les avantages des réseaux overlay proviennent de deux propriétés fondamentales :

1. les nœuds d'un réseau overlay sont en pratique des machines de haut niveau et ont des capacités supérieures pour réaliser les opérations de base de stockage et d'expédition par rapport aux réseaux de plus bas niveau.
2. la topologie des réseaux overlay, définie au dessus d'un réseau *IP* de grande taille, peut être construite et modifiée facilement dans le but de réaliser un codage réseau.

L'intérêt d'utiliser le codage réseau dans ce type de réseau est d'améliorer le débit de bout-en-bout d'une transmission multicast. Cet objectif est obtenu grâce à l'introduction de deux nouvelles notions.

Tout d'abord, la notion habituelle que des données peuvent seulement être répliquées et expédiées par les nœuds n'est plus considérée. En effet, en tant que système de haut niveau, chaque nœud possède la capacité de coder et de décoder des données au niveau de message en utilisant des codes linéaires efficaces.

Ensuite, la seconde notion classique qui est abandonnée est celle consistant à considérer que la topologie d'un réseau multicast de la source aux récepteurs doit être un arbre. [4] propose un ensemble d'algorithmes distribués pour construire des graphes multicast multi-redondant comme topologie du réseau multicast. Il indique ensuite comment construire un code réseau adéquat et l'appliquer sur ce type de graphe.

Les résultats présentés dans [4] montrent que des gains significatifs de débit de

2.4. Applications de codage réseau dans différents domaines et types de réseaux

bout-en-bout peuvent être obtenus par cette technique par rapport à des techniques de routage classique (protocole Narada). Toutefois, Ces performances restent inférieures à celles des algorithmes optimaux (*DVMRP*) [4].

2.4.3 Réseaux ad-hoc et sans fil

Même si, dans [1], le codage réseau a été présenté pour des transmissions multicast dans réseaux filaires, les réseaux sans fil sont particulièrement bien adaptés pour le codage réseau en raison de la nature des liens sans fil qui compliquent le routage.

L'application du codage réseau aux réseaux de paquet sans fil permet de réaliser des gains de performance importants comparé aux réseaux de routage sans fil traditionnels [51].

L'échange d'information entre deux nœuds dans un réseau sans fil (conversation audio, vidéo conférence entre deux participants, transmission de messages instantanée) peut être effectué efficacement en exploitant le codage réseau et la propriété de diffusion de la couche physique offerte par le milieu sans fil [7]. En plus des gains potentiels du débit, le codage réseau permet d'employer des solutions plus économiques en termes de l'utilisation des ressources que des solutions de routage.

Le codage réseau peut être appliqué dans plusieurs types de réseaux pour faire partager des fichiers comme dans les réseaux maillés sans fil, les réseaux sans fil avec un système d'antenne réparti (DAS) et les réseaux utilisant la coopération entre les utilisateurs.

L'intérêt du codage réseau est encore plus important en présence de sources multiples [52] [53].

Chapitre 2. État de l'art et notions de base

3

Codage réseau et réseaux avec des garanties de qualité de service (QoS)

Sommaire

| | | |
|------------|---|-----------|
| 3.1 | Introduction | 40 |
| 3.2 | Réseaux avec des garanties de qualité de service (QoS) | 40 |
| 3.2.1 | Concepts et exemples de réseaux avec des garanties de QoS . . | 40 |
| 3.3 | Besoins et applications du codage réseau au sein des réseaux avec QoS garantie | 49 |
| 3.3.1 | Intérêts du codage réseau pour les réseaux avec QoS garantie . | 49 |
| 3.3.2 | Conditions d'application du codage réseau | 51 |
| 3.4 | Le calcul réseau | 52 |
| 3.4.1 | Intérêts du calcul réseau pour notre problématique | 53 |
| 3.4.2 | Les notions de base du calcul réseau | 53 |

3.1 Introduction

Dans ce chapitre, Nous discutons et proposons l'application de codage réseau dans les domaines des réseaux avec des garanties de qualité de service (QoS).

Dans la première partie, nous présentons les concepts des réseaux avec des garanties de QoS. Ensuite plusieurs exemples de ce genre de réseau seront cités. Puis, dans la deuxième partie, nous expliquons pourquoi et comment appliquer le codage réseau dans ce type de réseaux. Finalement dans la troisième partie, nous ferons une présentation synthétique du **calcul réseau** et de ses notions de base.

3.2 Réseaux avec des garanties de qualité de service (QoS)

3.2.1 Concepts et exemples de réseaux avec des garanties de QoS

3.2.1.1 Internet et Qualité de Service

La Qualité de Service (QoS) désigne la capacité à fournir des niveaux de service réseaux aux applications et/ou aux utilisateurs.

La Qualité de Service dans des réseaux à commutation de paquets représente la faculté d'offrir une garantie acceptable sur des paramètres réseau comme la perte de paquets et la latence pour un service précis (voix sur IP, vidéo-conférence, etc.).

Un niveau élevé de service dans les réseaux est limité par les éléments réseau, comme les éléments extrémités, (envoi et réception) et les éléments intermédiaires. Il est nécessaire de considérer les limites des éléments réseau pour résoudre le problème de la garantie de QoS du réseau. Une QoS garantie offre des garanties sur la bande passante, les limites des délais de bout-en-bout et les tailles des files d'attente.

Le service offert par la couche réseau (IP) de l'Internet est de type "Best Effort", sans connexion et non fiable. Il est basé sur la philosophie du premier arrivé, premier servi. Ce service permet de maximiser l'utilisation des ressources et d'assurer la robustesse. Il implique le même traitement pour l'ensemble des flux qui traversent le réseau de communication. Par conséquent, ce service n'offre pas des garanties en termes de Qualité de Service.

La variabilité des délais de transmission (gigue) et des pertes de paquets n'a pas d'effets sur les applications typiques d'Internet telles que l'E-mail ou le transfert de fichiers. En revanche, les applications temps réel ne peuvent pas se satisfaire de ce type de service.

Il faut noter que, sur l'Internet, il n'y a aucune connaissance sur le chemin emprunté par les différents paquets, contrairement aux réseaux à commutation de circuits où durant toute la période de la communication, un circuit est réservé.

Un autre inconvénient d'Internet est l'absence de garantie sur la régularité du débit pour les services qui exigent une stabilité ou une bande passante déterminée.

Pour gérer ce type de problème, il existe des mécanismes de QoS permettant de fournir un certain niveau de prévisibilité et de contrôle au-delà du service de "Best Effort" d'IP en classifiant les différents flux réseau et en réservant de la bande passante

3.2. Réseaux avec des garanties de qualité de service (QoS)

pour les services nécessitant un service continu (temps réel).

Plusieurs modèles ont été proposés pour que les réseaux IP fournissent de la QoS. Les modèles des Services Intégrés (IntServ) et les modèles des Services Différenciés (DiffServ) sont les principaux.

3.2.1.1.1 Les Services Intégrés (IntServ) :

Le modèle d'IntServ propose deux classes de service qui visent le trafic en temps réel en plus de la classe "Best Effort" actuelle. Ces deux classes sont :

1. Service Garantie "Guaranteed Service" : ce service garantit une limite supérieure du délai de transmission de l'ensemble des paquets appartenant à un flux de cette classe.
2. Service de charge contrôlée "Controlled Load Service" : ce service est mise en œuvre pour les flux qui exigent un service fiable et amélioré par rapport au service "Best Effort" classique.

Ce modèle est conçu pour fonctionner aussi bien avec les transmissions multicasts qu'unicasts.

Dans le modèle des services intégrés, les ressources (e.g. bande passante) sont explicitement gérées afin de répondre aux exigences des applications. Ceci implique que la réservation de ressource et le contrôle d'admission sont les modules de construction principaux de ce modèle.

Pour réserver les ressources demandés pour un flux dans la classe précise, le protocole de signalisation (RSVP : Resource ReServation Protocol) est employé sur le chemin entre la source et le récepteur. IntServ garantit le niveau de QoS via le protocole de la réservation de ressource (RSVP) et le contrôle d'admission.

Pour faire une réservation de ressources, l'émetteur envoie une requête RSVP vers le récepteur. Si tous les routeurs, qui sont traversés par cette requête ont les ressources nécessaires pour le flux considéré, le récepteur envoie un message de réservation de ressources en réponse à la requête de l'émetteur.

Chaque routeur doit identifier le flux et les réservations des ressources correspondantes. L'acceptation des flux est déterminée selon un mécanisme de contrôle d'admission en fonction des ressources disponibles et du contrat entre le client et le fournisseur de service Interne (ISP : Internet Service Provider).

IntServ offre de fortes garanties en termes de qualité de service aux différents flux ayant réservé des ressources. Cependant, ce service a des limites dans le déploiement à grande échelle car chaque routeur traversé par un flux doit garder l'identification et les ressources réservées par ce flux. Sachant que le nombre de flux qui traversent les routeurs peut être très grand (surtout pour ceux qui sont situées au centre du réseau), ce service peut générer des délais dans ces routeurs.

En outre, le protocole de réservation de ressource (RSVP) est employé en tant que protocole de signalisation pour des réservations de ressource de bout-en-bout. L'implémentation de ce service doit donc être réalisée sur tous les routeurs traversés pour offrir un service garanti de bout-en-bout. L'inconvénient d'IntServ est son manque de flexibilité. De plus, le trafic généré par la signalisation peut augmenter significativement les communications entre les routeurs dans les grands réseaux.

Pour résoudre ces problèmes, des nouveaux types de services ont été proposés sous le nom des Services Différenciés (DiffServ) [54].

3.2.1.1.2 Les Services Différenciés (DiffServ) :

Les services différenciés ont été proposés comme une évolution d'IntServ et comme une réponse à ses problèmes de déploiement à grande échelle.

Dans ce service, les paquets qui entrent le réseau sont marqués par ordre de priorité. Ce marquage est aussi utilisé par les routeurs pour classer les paquets et pour les mettre dans les files d'attente. Les services différenciés prennent en compte un nombre limité de classes de service. Ils réduisent donc le nombre d'identifications et de ressources à réserver dans les routeurs (notamment pour les routeurs de centre du réseau).

La classification des flux se fait donc dans les routeurs à l'entrée du domaine de services différenciés. La complexité des opérations et la charge de traitement dans ces routeurs devient alors plus importante que celles effectuées dans les routeurs au centre de réseau. Malgré ça on peut considérer que le nombre de flux à traiter dans ces routeurs sont limités car ces routeurs se situent à l'entrée des domaines DiffServ et peuvent donc supporter des opérations de traitement plus lourdes.

Deux classes de services sont principalement proposées par les services différenciés. Ces classes de service sont :

1. La transmission accélérée (Expedited Forwarding : EF) : cette classe possède la plus haute priorité. Elle réduit au maximum les délais et la gigue (variations de délais) des flux qui l'utilisent car ceux-ci sont servis en premier. Les paquets sont supprimés si le trafic excède le seuil de charge maximum fixé par la politique locale.
2. La transmission garantie ou assurée (Assured Forwarding : AF) : cette classe permet de définir des priorités de délivrance des paquets. Les paquets qui respectent la politique locale sont délivrés avec une priorité haute. Les paquets d'AF qui ne respectent pas la politique locale ne sont pas livrés à la priorité indiquée. Au lieu de cela, ils sont classés à une priorité inférieure, mais ils ne sont pas supprimés.

Avec ce type de service, il n'est pas nécessaire d'indiquer ou de marquer les paramètres du type du service désiré ou de faire des réservations de ressource. En revanche, ces paramètres sont programmés dans des routeurs.

La priorité de cette classe de service se situe entre le priorité de la classe Expedited Forwarding et celle de la classe "Best Effort" dans les routeurs [54].

En plus de classes précédentes, il y a la classe "Best Effort" de la couche IP qui maximise l'utilisation des ressources en simplifiant les traitements dans les routeurs. Cette classe n'offre pas de garanties pour une qualité de service donnée. Les délais et les taux de pertes sont importants dans cette dernière classe de service en cas de congestion du réseau car elle ne dispose que des ressources non utilisées par les classes de services précédentes.

3.2. Réseaux avec des garanties de qualité de service (QoS)

3.2.1.2 ATM et Qualité de Service

L'ATM est un mode de transmission à haut débit dont le concept a été défini dans les années 80. Cette technique vient comme une suite logique à d'autres modes de transmission comme la commutation par paquets.

ATM, (Mode de Transfert Asynchrone), découpe les données en cellules de taille petite et fixe (chaque cellule contient une entête et une charge utile de 53 octets = 5 octets d'entête et 48 octets des données). Ceci facilite les traitements des données.

Le terme asynchrone se rapporte à la façon dont la bande passante est assignée parmi des connexions et des utilisateurs. "*Le réseau fonctionne en mode asynchrone, mais la commutation utilisée permet la synchronisation de l'horloge d'émission et de l'horloge de réception*". ATM utilise la commutation par circuit virtuel (VC). Les cellules dans la même connexion virtuelle maintiennent leur ordre séquentiel. L'en-tête sert à identifier si les cellules appartiennent à la même connexion virtuelle.

Les deux derniers octets de charge utile contiennent un code CRC de protection de la charge utile contre les erreurs. La correction d'erreur se fait seulement sur les données des entêtes et non pas sur la charge utile car les données de l'entête sont importantes pour l'acheminement. L'entête contient également un indicateur de type de charge utile.

Le réseau assure que les ressources nécessaires sont disponibles avant la définition d'une connexion virtuelle ce que limite la perte de cellules. En conséquence, l'intervention de l'ATM pour éviter la perte résultant de débordements ou d'encombres est aussi limitée.

Grâce aux cellules, qui sont petites et de taille constante, et au mode asynchrone, l'ATM est bien adapté pour fonctionner avec des réseaux à haut débit (fibres optiques) et permet d'offrir des qualités de service et des débits variés et d'acheminer de très nombreux types de services [55][56].

ATM est une technologie de réseau à commutateurs utilisant une signalisation d'"Interface Utilisateur-Réseau" (User-Network Interface : UNI) pour contrôler des connexions de bout-en-bout dans des réseaux d'ATM. La plupart des connexions ATM sont établies sur des circuits virtuels commutés (SVCS) sur lesquels des connexions sont créées et supprimées en fonction de la signalisation UNI.

Un des principaux intérêts de l'ATM est sa capacité à fournir un réseau téléphonique sûr sans avoir besoin de PVCs. Les PVCs (Permanent VC), sont des connexions logiques établies en permanence entre deux systèmes ATM offrant à la fois des niveaux de performance et de flexibilité importants.

Plus précisément, les commutateurs d'ATM ont la capacité de fonctionner à plusieurs niveaux de service :

- Débit binaire constant (CBR) : c'est un service de transfert des données aux taux fixes. CBR est employé pour les applications qui exigent une synchronisation précise pour garantir la bande passante et la QoS.
- Débit binaire variable (VBR) : c'est un service de transfert des données fournissant un débit spécifié, mais sans envoyer les données régulièrement comme dans CBR. VBR est un choix classique pour la VoIP et les applications de vidéoconfé-

rence.

- Débit binaire disponible (ABR) : c'est un service qui garantit une capacité minimum, mais qui permet au trafic de monter à des niveaux plus élevés lorsque la bande passante est disponible.
- Débit binaire non spécifié (UBR) : c'est un service utilisé pour acheminer le trafic qui ne nécessite pas de synchronisation. UBR n'a aucun mécanisme pour ajuster dynamiquement la quantité de bande passante disponible pour une connexion.

On peut noter que l'IP ne spécifie pas de débit binaire. En raison de sa capacité à gérer la QoS, l'ATM est particulièrement adapté pour les trafics multimédia, qui doivent avoir un CBR ou VBR en temps réel pour fonctionner efficacement. Mais il est également important pour le trafic régulier, car il permet aux opérateurs de bien gérer la bande passante en temps réel. La commutation d'ATM fonctionne en combinant plusieurs circuits virtuels (VCs) dans un chemin virtuel (VP). Plusieurs VPs sont à leur tour combinés dans un circuit physique (un câble ou une fibre optique).

Chaque cellule d'ATM s'identifie avec un VCI et un VPI (des identificateurs de VC et de VP) pour indiquer aux commutateurs d'ATM comment la cellule doit être manipulée. Quand la cellule se déplace entre les commutateurs, ses valeurs de VCI et de VPI sont réécrites pour diriger la cellule vers son prochain destinataire. De cette façon, le système peut assurer l'intégrité de chemin et l'adaptabilité [55][56].

L'ATM emploie un ensemble de mécanismes de gestion du trafic pour fournir des niveaux de service. Par exemple, dans chaque cellules ATM, il y a un bit pour la priorité de perte (CLP : Cell Loss Priority), ce bit indiquant si une cellule ATM peut être abandonnée en cas de congestion. L'ATM emploie également un mécanisme de "seau percé" pour reformer les flux de trafic quand ils entrent dans le réseau. D'autres mécanismes d'ATM sont employés pour séparer le trafic par type de service et ainsi maximiser l'utilisation de la bande passante. ATM emploie quatre paramètres du trafic de source :

- Taux maximum de cellules (PCR) : c'est le taux maximum auquel les cellules peuvent être transportées sur une liaison.
- La taille maximum de la rafale (MBS) : c'est le nombre maximum de cellules qui peuvent être transférées dans un flot continu.
- Taux de cellules viable (SCR) : c'est le taux moyen de transfert des cellules admissible à long terme pour une liaison.
- Taux minimum des cellules (MCR) : c'est le taux minimum de transfert des cellules admissible pour une liaison.

ATM utilise des mécanismes de QoS pour offrir les meilleurs services en termes de QoS. En réalité, une grande partie du trafic d'ATM dans l'Internet utilise les niveaux de service du débit binaire non spécifié (UBR) parce que la plupart des connexions dans l'Internet sont asynchrones et donc effectuées avec délai et gigue. En plus, la plupart des réseaux ATM sont des réseaux centraux (backbone) et pas des réseaux de bout-en-bout [56] [57].

3.2. Réseaux avec des garanties de qualité de service (QoS)

3.2.1.3 Les réseaux embarqués

"Un système embarqué est un ensemble de fonctions périodiques ou aperiodiques exécutées par un ensemble de ressources de calcul ou de communication interconnectées" [58]. Les principales caractéristiques et contraintes de ces réseaux sont :

- Le partage des ressources (bus de données multiplexés, ressources de traitement gérées par un exécutif temps réel, ressources de calcul et de communication banalisées). Un exécutif est "un système d'exploitation qui ordonnance les exécutions des tâches et protège l'accès aux ressources partagées et qui reçoit et transmet les signaux de synchronisations".
- Les éléments de réseau sont bien contrôlés (maîtrisés), (tous les nœuds "émetteurs, récepteurs, routeurs/commutateurs", et tous les liens sont sous contrôles). La maîtrise du comportement temps réel signifie la maîtrise des délais d'exécution et de communication, la maîtrise des gîgues, la maîtrise des dimensionnes des bus, des files d'attente, de la puissance de calcul, etc.
- Déterminisme temps réel du réseau [58].

Déterminisme du réseau

Le déterminisme du réseau est sa capacité à borner certains paramètres. Par exemple, il est parfois nécessaire de connaître la borne supérieure de la latence. Si on suppose que les transmissions des systèmes terminaux ne sont pas synchronisées, les paquets de données arrivent distribués dans le temps et la possibilité de collisions et de congestions existe. Ceci a comme conséquence des augmentations sur les bornes des délais de transferts.

Même si les transmissions sont synchronisées, la connaissance des bornes maximums des délais de transferts est un problème important. Ce problème peut être étudié en utilisant en particulier la théorie du calcul réseau pour proposer des enveloppes de trafic pour ce réseau. Par exemple, une étude sur le déterminisme sur un réseau de type Ethernet redondant et fiabilisé AFDX (Avionics Full Duplex switched ethernet :AFDX) a été réalisée dans [58].

Robustesse et tolérance aux fautes

La redondance des données d'un réseau est utilisée dans les cas de perte d'informations dues à des congestions ou à des pannes dans les liens ou les routeurs. Dans les réseaux embarqués, la fiabilité est très importante et doit être particulièrement élevée. Les moyens qui sont mis en œuvre pour améliorer la robustesse globale du système consistent souvent à dupliquer les matériels.

3.2.1.3.1 Le réseau embarqué AFDX

Le réseau AFDX est constitué de calculateurs et de réseaux de communication partagés. Les applications avioniques, distribuées sur ces architectures, ont certaines contraintes (maîtrise des latences, qualité de service, échange de données assurées et fiables, ...).

Pour le réseau avionique, chaque requête de transmission dans le réseau doit être transmise, sans tenir compte du type de données, avec un délai de transit maximal dans le réseau (aussi appelée latence de bout-en-bout). Ce délai doit être garanti.

La notion de garantie de service fournit ces deux types de garantie : une limite supérieure du délai et une largeur de bande constante. Elle permet d'établir une connexion logique entre un nœud émetteur et un ou plus nœuds récepteurs. Les paquets qui appartiennent à la même connexion définissent un flux.

Il est important de noter que le réseau AFDX utilise les notions de redondance et de lien virtuel. Une redondance matérielle du réseau est prévue afin d'augmenter la robustesse du réseau AFDX. Elle consiste à doubler le réseau (qui peut ainsi être vu comme deux réseaux superposés). Chaque abonné (système terminal ou "end system") aura ainsi une connexion sur chaque réseau qui lui permettra d'émettre sur les deux réseaux et de sélectionner un des deux paquets à la réception [59] [60].

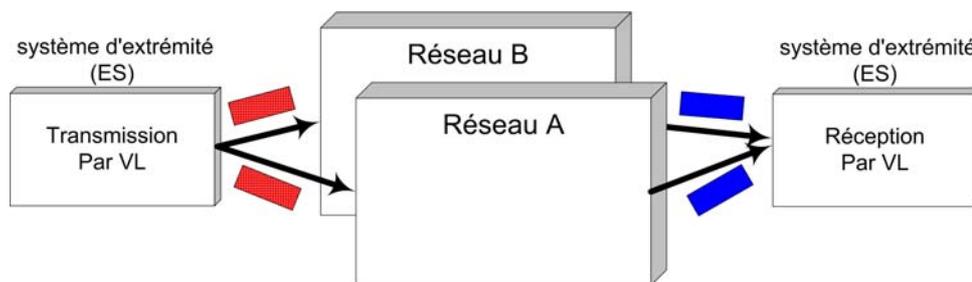


FIGURE 3.1 – Le concept de réseau redondant

3.2.1.3.1.1 lien virtuel (VL)

Le concept le plus important introduit par l'AFDX est le concept des liens virtuels qui permet d'isoler les transferts des données entre un émetteur et des récepteurs. Un lien virtuel définit une connexion unidirectionnelle logique d'un système terminal (source) jusqu'à un ou plusieurs systèmes terminaux (destinations) et un sens de transfert (voir la Figure 3.2) :

Chaque lien virtuel a une largeur de bande maximum dédiée. Un lien virtuel a :

- Un abonné émetteur unique, un ou plusieurs abonnés en réception.
- Un temps maximum garanti de transfert des trames vers les récepteurs et ça quelque soit le comportement du reste du réseau.
- Un identifiant unique.

Pour chaque lien virtuel, le système terminal doit maintenir les données dans l'ordre où elles sont délivrées par l'application, dans la transmission et la réception.

Le commutateur (switch) connaît par une table de configuration statique les VL qu'il doit commuter ainsi que le nombre de trames autorisées par VL. Le concept de VL se concrétise par deux services implémentés dans les abonnés :

- Un service d'admission AFDX : son rôle est de permettre à l'application d'accéder aux VL en émission
- Un service de réception AFDX : il a la charge de mettre à disposition des applications les données utiles et ceci dès la réception.

Pour plus de détails, voir [59] [60].

3.2. Réseaux avec des garanties de qualité de service (QoS)

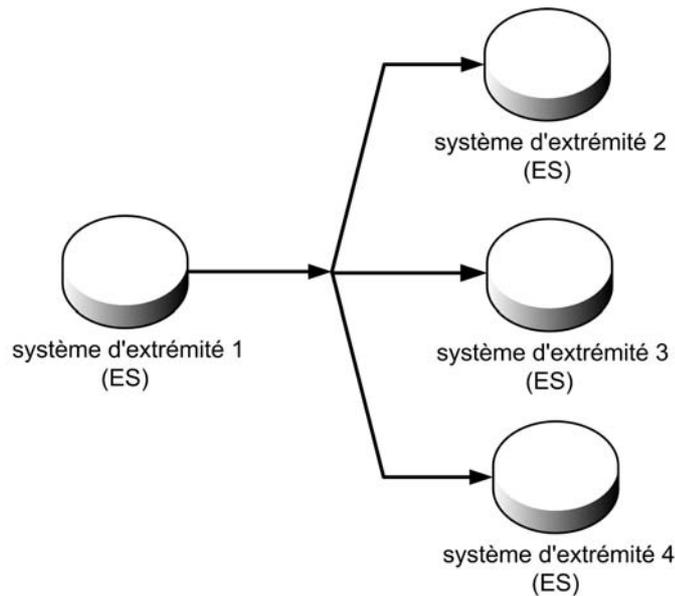


FIGURE 3.2 – Un lien virtuel

3.2.1.3.1.2 Représentation des flux

Le réseau AFDX assure à chaque paquet un cheminement unique de bout-en-bout grâce aux VL. Au niveau de chaque abonné, tout paquet est affecté à l'émission à un VL prédéfini par configuration. Le paquet contient dans la partie adressage l'identification du VL et cette identification est utilisée pour le diriger dans le réseau. Par conséquent, sur tout le cheminement du paquet, à chaque traversée d'un commutateur, un service vérifie si ce paquet est autorisé à y passer. De même, à la réception, l'abonné vérifie l'appartenance du paquet à un VL autorisé en réception. Si ce n'est pas le cas le paquet est détruit. La bande passante allouée à un lien virtuel est définie comme le nombre de "paquets" (frames AFDX) envoyés par seconde et leur taille. La bande passante n'est toutefois pas suffisante pour caractériser un trafic de paquets sur un lien. Par conséquent, le traitement du lien virtuel est réalisé par un mécanisme de contrôle de flux qui régule les flux de données produits par différentes sources qui appartiennent à ce système terminal. Ce mécanisme fournit une partition pour les niveaux de réseau [59] [60].

Une frame AFDX de taille minimum se compose de 84 octets au total. Le flux des frames liés à un lien virtuel particulier est caractérisé par deux paramètres :

- **BAG** : intervalle d'affectation de largeur de bande passante (**B**andwidth **A**llocation **G**ab). Un mécanisme de contrôle de la largeur de bande passante. Le BAG représente l'espace minimum entre les premiers bits de deux frames AFDX consécutives.
- **gigue** : la variation de la latence.

Pour garantir les BAGs de tous les VL, on doit appliquer un contrôle de flux c'est-à-dire que tous les paquets courants doivent passer dans un régulateur. En plus, si on a de multiples VL, on doit mettre un ordonnanceur (Scheduler) multiplexeur qui

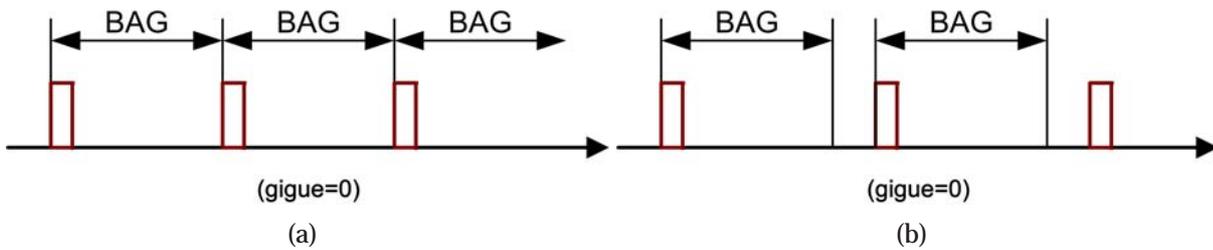


FIGURE 3.3 – (a) : Le BAG dans un VL pour un flux de données maximum sur la largeur de bande. (b) : Le BAG dans un VL pour un flux de données non maximum sur la largeur de bande.

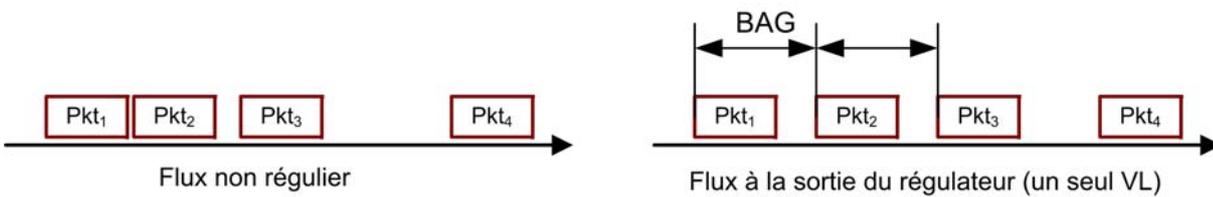


FIGURE 3.4 – Régulation du flux d'un lien virtuel

réalise le multiplexage des différents flux. Le but du régulateur est la limitation du

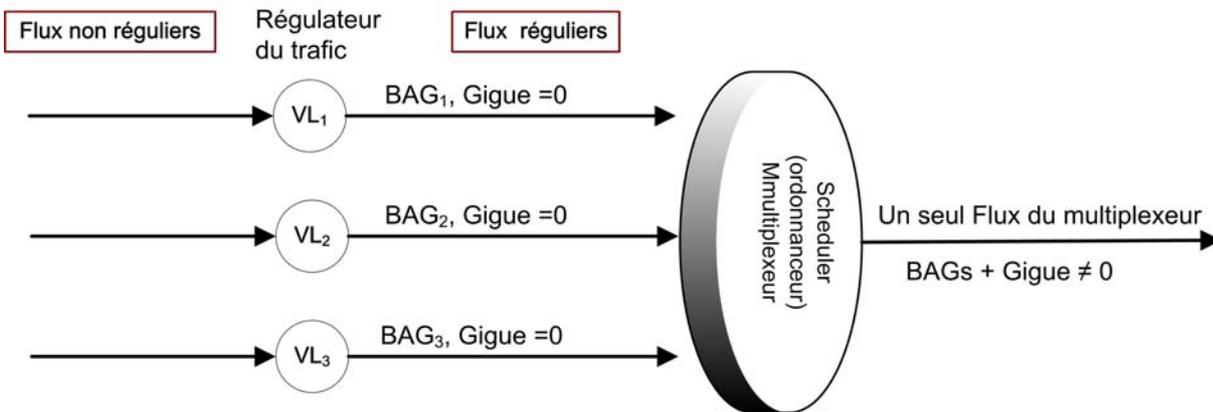


FIGURE 3.5 – Mécanisme de control de flux avec un ordonnanceur

débit des paquets instantané dans un lien virtuel par un **espacement de paquets**. A la sortie de l'ordonnanceur, et pour un lien virtuel donné, les frames peuvent apparaître dans un intervalle de temps limité. Cet intervalle est défini comme la gigue maximum admissible. L'utilisation maximale de la largeur de bande de chaque VL est caractérisée par leur BAG et la taille maximale autorisées de la frame du VL (notée L^{max}). Cette valeur est donnée par la formule L^{max}/BAG ou la Bande passante [59] [60].

3.3. Besoins et applications du codage réseau au sein des réseaux avec QoS garantie

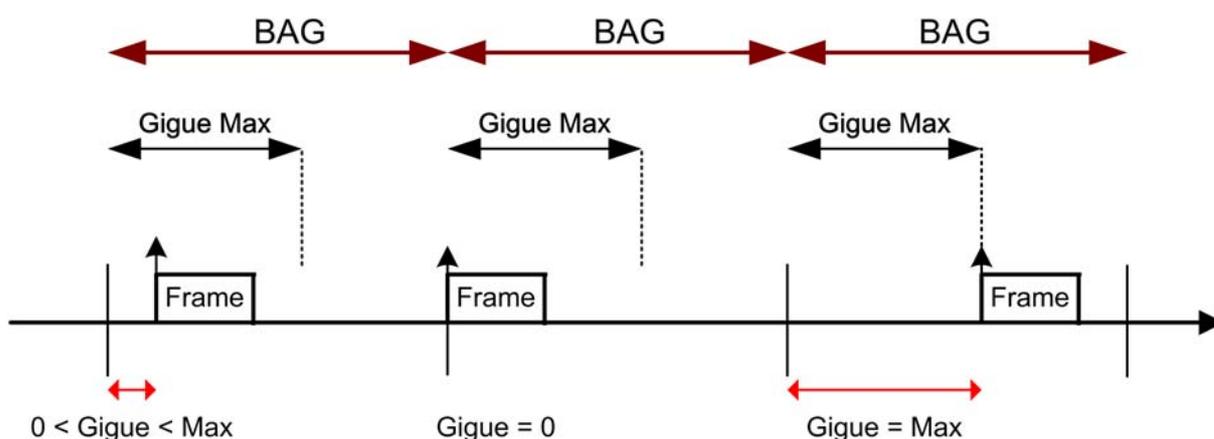


FIGURE 3.6 – L'effet de la gigue sur un flux de données maximum sur la largeur de bande

3.3 Besoins et applications du codage réseau au sein des réseaux avec QoS garantie

Comme on peut le voir sur les principales références du codage réseau, ce domaine est relativement récent. De plus, comme indiqué dans la partie 3.1, les résultats théoriques ne sont pas directement applicables sur des réseaux réels. Les principaux travaux proposant d'utiliser le codage réseau dans des réseaux réels ont été présentés dans les sections précédentes.

Dans les parties qui suivent, nous proposons d'étudier l'utilisation possible du codage réseau dans un autre type de réseau : les réseaux avec QoS garantie.

3.3.1 Intérêts du codage réseau pour les réseaux avec QoS garantie

Sans considérer un réseau avec des garanties de QoS en particulier, nous pouvons constater que le contexte des réseaux avec des garanties de QoS semble favorable à l'utilisation du codage réseau. Les principales raisons de cet optimisme sont les suivantes :

- La plupart des intérêts du codage réseau ont un sens dans un contexte avec des garanties de QoS :
 1. Délai maximum de traversée du réseau.
 2. Taille des files d'attente des routeurs.
 3. Optimisation du rapport coûts (quantité de matériel) / niveau de robustesse.
- On peut considérer que l'ensemble des matériels et des logiciels sont homogènes (un système embarqué par exemple). De plus, les problèmes de synchronisation entre les flux (système embarqué), les problèmes de taille uniforme de paquet semblent pouvoir être résolus (ATM). De plus, dans le contexte embarqué,

la taille de ces réseaux étant habituellement "raisonnable", on peut penser que chaque nœud peut avoir une connaissance globale du réseau et donc du code utilisé. Ceci permettrait, en particulier, d'éviter d'inclure dans chaque paquet le codage utilisé, comme dans la partie 2.4.1.

- Dans les réseaux embarqués et ATM, on peut considérer généralement qu'une "autorité supérieure" possède une vision globale du réseau. On peut donc penser que, si les avantages du codage réseau sont clairement exposés et clairement compris, la décision d'implémenter cette technique peut être prise.

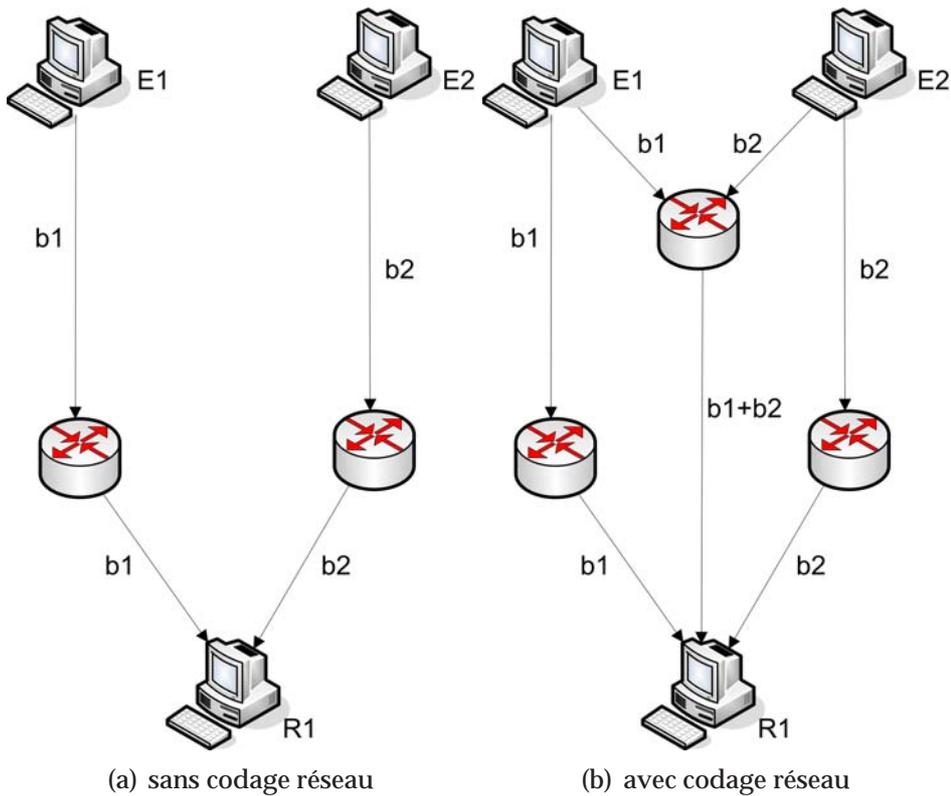


FIGURE 3.7 – L'intérêt du codage réseau en termes de robustesse

Les réseaux avec QoS garantie ont des contraintes en ce qui concerne les qualités des services offerts aux flux (contrat de service). Deux points clés sont examinés en vue de ce contrat de service. Il s'agit de la fiabilité du service ou la tolérance aux fautes et du déterminisme.

La tolérance aux fautes est généralement apportée grâce à une redondance matérielle et logicielle. Le codage réseau permet d'optimiser la quantité de matériel pour un niveau de tolérance aux fautes donné.

L'utilisation de codage réseau peut être intéressante dans ce contexte car dans certains cas, comme sur l'exemple représenté sur la Figure 3.7. Il permet d'optimiser le compromis niveau de robustesse / quantité de matériel.

En ce qui concerne le déterminisme, le codage réseau, qui améliore de manière générale l'utilisation des liens, doit permettre de diminuer la borne maximum de délai de transit dans le réseau.

3.3. Besoins et applications du codage réseau au sein des réseaux avec QoS garantie

En effet, comme l'explique l'exemple sur la Figure 3.8 qui présente un réseau de transmission avec 2 sources de transmissions multicast, on observe un point de congestion-

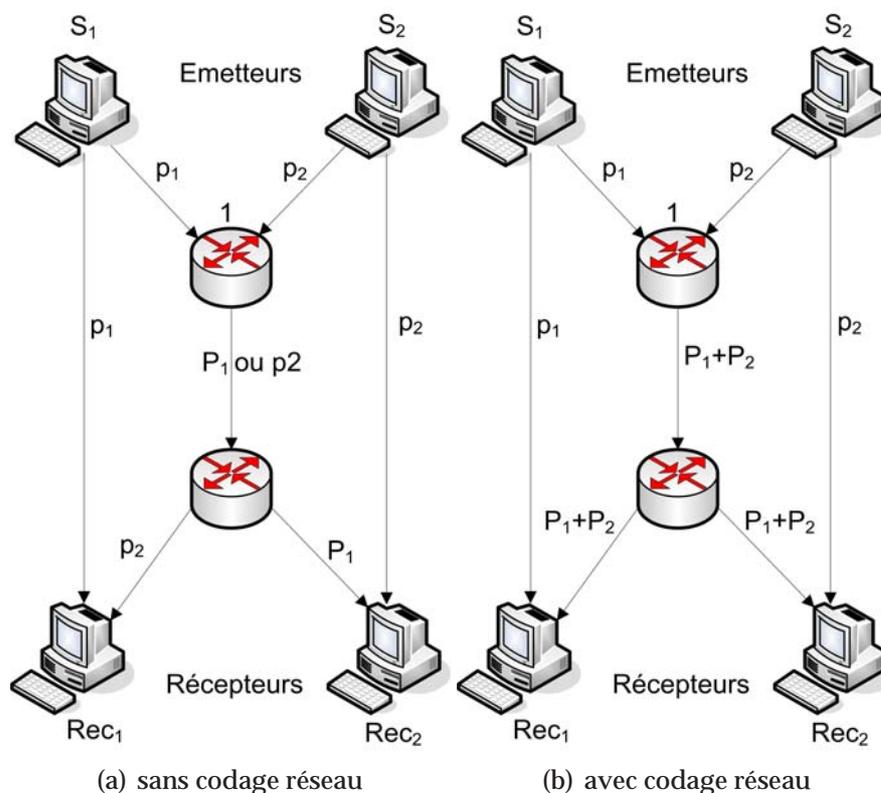


FIGURE 3.8 – L'intérêt du codage réseau en termes de déterminisme

tion potentiel sur le routeur 1. Cette congestion aura nécessairement une influence négative sur les délais maximum de transfert des données. De plus, la taille de file d'attente en 1 devra être dimensionnée en conséquence. On peut voir qu'avec le codage réseau (schéma 3.8(b)), on peut supprimer le point de congestion (ou en pratique, le réduire). Ceci a bien évidemment une conséquence positive sur la borne maximale du délai. En termes de calcul réseau, il est à prévoir que la pente de la courbe représentant l'enveloppe de trafic sortant soit réduite.

3.3.2 Conditions d'application du codage réseau

En plus des différents points positifs communs à tous les réseaux avec QoS garantie présentés dans la Section 3.2.1, la définition même des réseaux ATM et embarqués fait que les conditions d'application du codage réseau semblent très favorables.

En effet, les réseaux ATM en eux-mêmes, qui sont des réseaux commutés (le réseau ATM est un réseau à commutateurs), font en sorte que le trafic soit routé à différents points du réseau, ce qui offre un bon opportunité d'appliquer le codage réseau.

De plus, les notions de circuit virtuel (VC) et de chemin virtuel (VP) qui peuvent traiter de liens multicast, semble tout à fait pouvoir être adaptées pour un codage ré-

seau. En effet, la taille des paquets (cellules) transmis font qu'il semble tout à fait faisable de combiner des paquets issus de plusieurs chemins virtuels avec les mêmes caractéristiques.

Par rapport aux réseaux embarqués, ils sont aussi des réseaux commutés (par exemple, le réseau embarqué AFDX est un réseau Ethernet commuté) et ils utilisent aussi des notions de lien virtuel (VL) et d'espacement entre ces paquets ce qui facilite l'adaptation de codage réseau

Un autre point positif concerne l'utilisation possible des Scheduler et régulateurs de trafics déjà existants dans les réseaux embarqué AFDX qui devraient permettre de synchroniser les différents flux.

En résumé, il semble que le codage puisse avoir les avantages suivants dans ce contexte :

1. amélioration de l'efficacité globale du système avec QoS garantie.
2. possibilité de réduction des coûts (à niveau de robustesse équivalent).
3. augmentation de la fiabilité et de la performance de ces réseaux.

3.4 Le calcul réseau

Le calcul réseau est une théorie récente permettant d'analyser les problèmes des systèmes de files d'attente représentant les réseaux de communication. Il est basé sur l'algèbre min-plus. Le calcul réseau permet en particulier d'obtenir des limites déterministes sur d'autres paramètres de Qualité-de-Service (QoS) tels que la taille maximum de file d'attente et le délai de bout-en-bout dans un réseau de commutation de paquets tel que l'Internet. Il constitue l'une des principales théories qui peuvent être utilisées pour fournir des garanties de QoS. Si l'un des domaines d'applications naturels concerne les applications multimédia, cette théorie est cependant applicable à une variété de problèmes concernant les flux dans les réseaux.

Certains de ces concepts ont d'abord été présentés par R. L. Cruz [11, 12]. Il a supposé que les flux à l'entrée d'un réseau de communication sont modélés de façon spécifique et qu'une certaine garantie est offerte par chaque élément de réseau. En introduisant des techniques basées sur l'algèbre traditionnelle, il a montré comment calculer les bornes supérieures de délai dans des éléments de réseau.

En s'appuyant sur les travaux de R. L. Cruz, C. S. Chang [61] a proposé d'employer l'algèbre $(\min, +)$ (l'opération d'addition est remplacée avec l'opération "min" et la multiplication est remplacée par "+") au lieu de l'algèbre habituelle pour simplifier les calculs. Il a défini la notion de convolution $(\min, +)$, la fermeture sous-additive (sub-additive closure) d'une fonction non décroissante. Il a montré l'optimalité du modelage de type seau percé (leaky bucket shaping).

Les concepts du calcul réseau ont été développés un peu plus tard par J-Y. Le Boudec et P. Thiran [10]. Ces auteurs ont montré qu'il est possible de modéliser analytiquement divers types de garanties de réseau, que les limites sont strictes et qu'elles sont applicables de bout-en-bout.

Nous expliquerons dans la suite l'importance du calcul réseau et les raisons qui nous ont poussées à l'utiliser dans notre travail. Nous présenterons ensuite les notions de base du calcul réseau.

3.4.1 Intérêts du calcul réseau pour notre problématique

Nous verrons dans les chapitres suivants que l'un des principaux objectifs de travail est l'analyse du temps de traversée des paquets dans un réseau et de la taille des files d'attente. Plus précisément, notre but est de déterminer les pires cas des délais et des tailles des files d'attente au niveau d'un routeur et au niveau d'un réseau complet. Le calcul réseau permet justement d'obtenir des garanties déterministes sur ces caractéristiques du réseau.

De plus, cette théorie a l'avantage de modéliser les flots de données à l'entrée du réseau selon des contraintes du contrat de trafic spécifique qui comprend des paramètres permettant de présenter un flot, par exemple, selon un modèle de débit maximal ρ et de la taille maximale de backlog σ [10]. Cette caractérisation a l'avantage aussi d'être assez simple à manipuler contrairement à la théorie des files d'attente et à d'autres approches (modélisations) stochastiques qui sont utilisées quand on n'a pas de connaissances totales des modèles des flux d'arrivées. En effet, ces approches sont, jusqu'à présent, un peu complexes pour réaliser des analyses complètes d'un réseau de bout-en-bout et qui ne produisent que des garanties probabilistes et non pas des bornes déterministes [62].

Dans le cas où on a une connaissance complète sur les flux d'entrée, il sera possible de caractériser les flux d'arrivées avec des bornes supérieures. Donc, l'approche du calcul réseau est très utile car elle permet d'évaluer les performances du réseau en termes de délai et de tailles de files d'attente [62].

Outre les contraintes sur les flux qui exigent que les flots du trafic entrant soient contrôlés par une courbe d'arrivée, chaque élément du réseau offre un service minimum représenté par une courbe de service. Cette courbe représente le pire cas du service offert par cet élément.

Cet ensemble de raisons nous a conduit à utiliser le calcul réseau pour quantifier l'utilisation du codage réseau dans notre contexte.

3.4.2 Les notions de base du calcul réseau

Dans cette section nous étudierons l'algèbre Min-Plus et les concepts les plus importants tels que des courbes d'arrivée et des courbes de service.

Considérons un système S , qui reçoit des données en entrée et qui délivre des données après un délai variable. Un flot de données F transmis sur un lien peut être représenté comme une *fonction cumulative* $R(t)$, telle que pour tout $y > x$, $R(y) - R(x)$ est la quantité de données transmise sur ce lien durant l'intervalle de temps $[x, y]$.

Supposons que $R(t)$ soit une fonction d'entrée représentant le trafic cumulé à l'entrée jusqu'à l'instant t . $R^*(t)$ représente le trafic cumulé à la sortie jusqu'à l'instant t .

Chapitre 3. Codage réseau et réseaux avec des garanties de qualité de service (QoS)

Une hausse de la pente de la fonction R pendant un certain intervalle de temps représente une rafale de trafic. Tous ces concepts sont tirés des références [10] [11] [12].

Soit \mathcal{F} l'ensemble des fonctions croissantes au sens large, c'est-à-dire telles que $f \in \mathcal{F}$ si et seulement si $f(s) \leq f(t)$ pour tout $s \leq t$.

1. **Min-plus convolution :**

Soient f et g deux fonctions de \mathcal{F} . La convolution min-plus de deux fonctions f et g est défini comme :

$$(f \otimes g)(t) = \inf_{0 \leq s \leq t} [f(t-s) + g(s)]$$

2. **Min-Plus Déconvolution :**

Soient f et g deux fonctions de \mathcal{F} . La déconvolution min-plus de f par g est la fonction :

$$(f \oslash g)(t) = \sup_{u \geq 0} [f(t+u) - g(u)]$$

3. **Délai Virtuel :**

le délai virtuel à l'instant t est :

$$d(t) = \inf[\tau \geq 0 : R(t) \leq R^*(t + \tau)]$$

4. **Courbe d'Arrivée $\alpha(t)$ ou Courbe d'Arrivée Maximale :**

Soit F un flot de données définie par sa fonction cumulative $R(t)$. Soit $\alpha(t)$ une fonction croissante définie pour $t \geq 0$ (i.e. $\alpha(t) \in \mathcal{F}$). On dit que R est contraint par $\alpha(t)$ si pour tout $s \leq t$:

$$R(t) - R(s) \leq \alpha(t - s)$$

On dit aussi que R est limitée par $\alpha(t)$ et que R admet $\alpha(t)$ comme courbe d'arrivée maximale.

Plus généralement, nous disons qu'un flux R est α -smooth, où $\alpha(t)$ est une fonction croissante au sens large, si la quantité de données qui peuvent être observées sur le flux sur n'importe quelle fenêtre de temps de durée t est inférieure à $\alpha(t)$.

5. **Courbe d'Arrivée affine $\alpha_{(\sigma, \rho)}(t)$:**

Une classe courante de courbes d'arrivée est la classe des fonctions affines. Une courbe d'arrivée affine α peut être représentée sous la forme

$$\alpha(t) = \sigma + \rho * t$$

pour $t > 0$ et 0 autrement, où σ représente le backlog et ρ représente le débit maximal. Dans ce cas, $R(t)$ est borné par $\alpha(\sigma, \rho)$:

$$R(t) - R(s) \leq \sigma + \rho(t - s)$$

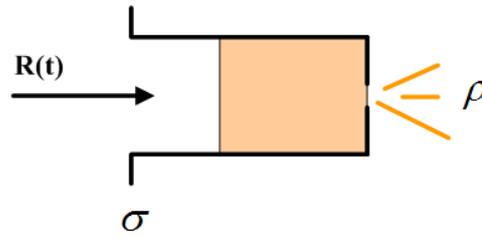


FIGURE 3.9 – Contrôleur du flux de type "Seau Percé"

La courbe $\alpha(\sigma, \rho)(t)$ représente la courbe d'arrivée d'un contrôleur de flux de type "seau percé" (leaky bucket controller) avec un taux de fuite ρ et une taille de seau de σ .

Le contenu d'un seau percé se comporte exactement comme une file d'attente servie au taux ρ , et de la capacité σ . Ainsi, le flux $R(t)$ est conforme à ce seau percé si et seulement si le contenu de ce seau n'excède jamais σ .

6. Courbe d'arrivée minimale ou enveloppe :

L'enveloppe d'un flux R est définie par $R \oslash R$. Par définition, nous avons :

$$(R \oslash R)(t) = \sup_{v \geq 0} [R(t+v) - R(v)]$$

7. Courbe de Service :

La courbe de service est utilisée pour décrire le comportement des éléments réseau. Considérons un système S et un trafic dans S caractérisé par des fonctions du trafic cumulé à l'entrée et à la sortie R et R^* . On dit que S offre au flux R une courbe de service β si β est une fonction positive croissante telle que pour tout $t \geq 0$:

$$R^*(t) \geq \inf_{s \leq t} (R(s) + \beta(t-s)) \quad (3.1)$$

Cette formule peut aussi s'écrire sous la forme suivante :

$$R^*(t) \geq (R \otimes \beta)(t).$$

Exemples de courbes de service :

- Un élément avec un débit constant C : cet élément offre une courbe de service $\beta(t) = Ct$.
- Un serveur avec un délai $\leq T$ (guaranteed delay server) : ce serveur offre une courbe de service $\delta(t) = +\infty$ pour $t > T$ et 0 pour $0 \leq t \leq T$ (une fonction de Dirac au point T).
- Un serveur avec un débit de service C et un délai de traversée T , offre une courbe de service débit-latence $\beta_{C,T}(t) = C(t-T)^+$.

8. Borne maximale de la taille de file d'attente ou Arrière de trafic (Backlog) :

Avec les notions de courbe d'arrivée maximale et de courbe de service, nous pouvons calculer les bornes maximales de la taille de file d'attente (l'arrière de trafic)

et du délai. L'arriéré de trafic d'un flux R dans un nœud à l'instant t est la quantité de données "en transit" dans le nœud. Supposons un flux R , contraint par la courbe d'arrivée α , qui traverse un système offrant une courbe de service β . L'arriéré de trafic B , défini comme $R(t) - R^*(t)$ pour tout t , satisfait :

$$B = R(t) - R^*(t) \leq v(\alpha, \beta)$$

où $v(\alpha, \beta) = \sup_{s \geq 0} [\alpha(s) - \beta(s)]$ représente la déviation verticale maximale entre les deux courbes α et β .

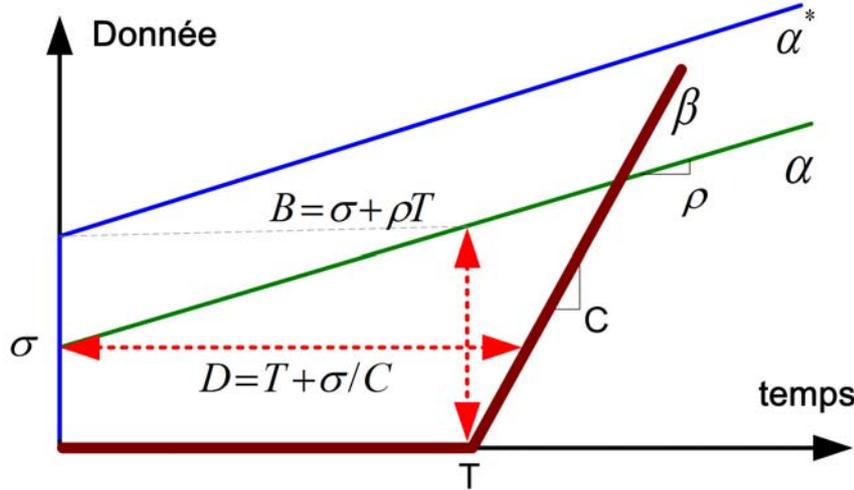


FIGURE 3.10 – Bornes maximales des délais et des tailles des files d'attente

9. **Borne maximal du délai D :**

Supposons un flux R , contraint par la courbe d'arrivée α , qui traverse un système S offrant une courbe de service β . Le délai maximal D subi par ce flux pour tout t satisfait :

$$D \leq h(\alpha, \beta)$$

où $h(\alpha, \beta)$ représente la déviation horizontale maximale entre les deux courbes α et β .

10. **Les Fonctions en escalier (Staircase Functions) $v_{T,\tau}$:**

Les fonctions en escalier sont utilisées pour caractériser des flux périodiques. Par exemple, un flux générant des paquets de taille L où l'intervalle de temps entre deux paquets varie entre T et $T + \tau$:

$$v_{T,\tau}(t) = \begin{cases} \lceil \frac{t+\tau}{T} \rceil & \text{si } t > 0 \\ 0 & \text{autrement} \end{cases}$$

où l'intervalle $T > 0$ et la tolérance $0 \leq \tau \leq T$.

11. **Flux de Sortie :**

Supposons un flux $R(t)$ contraint par une courbe d'arrivée $\alpha(t)$, qui traverse un système offrant une courbe de service β . Le flux de sortie R^* est contraint par la courbe d'arrivée : $\alpha \circ \beta$, où $(\alpha \circ \beta)(t) = \sup_{s \geq 0} \{\alpha(t+s) - \beta(s)\}$.

12. **Modeleur de trafic (Shaper) :**

Un modeleur de trafic dans un nœud de réseau stocke les paquets de données d'un flux entrant dans une file d'attente et les retarde tels que le flux sortant soit conforme à des spécifications du trafic données. Par exemple, il peut s'assurer que le flux sortant a une régularité bornée, ou que les paquets du flux sortant ont un temps minimum entre arrivées déterminé. Le flux à la sortie du modeleur est alors borné par une courbe d'arrivée.

$$R^*(t) - R^*(s) \leq \alpha(t - s)$$

où $s \leq t$ pour tous les s, t .

13. **Modeleur de type seau percé glouton (Greedy Leaky Bucket Shapers) :**

Un modeleur de type seau percé glouton est un cas spécial d'un modeleur du trafic. Il assure que le flux sortant respecte des spécifications du trafic prédéfinies, tout en garantissant que les paquets du flux entrant seront transmis le plus tôt possible[10].

$$R^* = R \otimes \sigma$$

où σ est la courbe de service d'un modeleur glouton (sub-additive et $\sigma(0) = 0$)

Chapitre 3. Codage réseau et réseaux avec des garanties de qualité de service (QoS)

4

Nouvelles stratégies de codage réseau

Sommaire

| | | |
|------------|---|-----------|
| 4.1 | Introduction et hypothèses | 60 |
| 4.2 | Stratégie Orientée Réseau (NOS : Network-Oriented Strategy) | 62 |
| 4.2.1 | Introduction | 62 |
| 4.2.2 | Définitions et hypothèses | 62 |
| 4.2.3 | Architecture d'un nœud de codage | 63 |
| 4.2.4 | NOS au niveau d'un nœud de codage | 64 |
| 4.2.5 | NOS au niveau d'un réseau | 66 |
| 4.2.6 | Discussion sur le décodage | 71 |
| 4.3 | Stratégie Orientée Flux (FOS : Flow-Oriented Strategy) | 71 |
| 4.3.1 | Introduction | 71 |
| 4.3.2 | Définitions et hypothèses | 71 |
| 4.3.3 | FOS au niveau d'un nœud de codage | 72 |
| 4.3.4 | FOS au niveau d'un réseau | 75 |
| 4.3.5 | Discussion | 77 |
| 4.4 | Stratégie de Transfert Rapide (FFS : Fast Forwarding Strategy) | 77 |
| 4.4.1 | Introduction | 77 |
| 4.4.2 | Définitions et hypothèses | 78 |
| 4.4.3 | Présentation de la stratégie FFS | 79 |
| 4.4.4 | FFS au niveau d'un nœud de codage | 81 |
| 4.4.5 | FFS au niveau d'un réseau | 85 |
| 4.4.6 | FFS au niveau de récepteurs | 85 |
| 4.4.7 | Discussion et considérations pratiques | 87 |
| 4.5 | Stratégie Classique de Routage/Multiplexage (MS : Multiplexing Strategy) | 89 |

| | | |
|-------|------------------------------------|----|
| 4.5.1 | Introduction | 89 |
| 4.5.2 | MS au niveau d'un nœud | 89 |
| 4.5.3 | MS au niveau d'un réseau | 90 |
| 4.6 | Conclusion générale | 92 |

4.1 Introduction et hypothèses

Nous proposons d'introduire le codage réseau dans la classe des réseaux fournissant des qualités de service (QoS) garanties. Rappelons que dans ce type de réseaux, les routeurs mettent en application des stratégies assurant aux flux un niveau de QoS garantie exprimé en termes de délai de bout-en-bout et/ou du débit minimal. En contrepartie, les flux respectent des contraintes de régularité et de débit maximal. Les différentes garanties et contraintes sont exprimées comme un ensemble de bornes caractérisant le réseau.

Les hypothèses utilisées dans la suite sont les suivantes :

Nous supposons qu'un réseau est représenté par un graphe acyclique orienté $G = (V, E)$ où V est un ensemble de nœuds et E un ensemble de liens orientés. Les liens sont caractérisés par une capacité donnée et un délai de transmission maximal.

L'ensemble de nœuds est divisé en trois catégories. La première comprend les *nœuds sources* qui produisent chacun un ou plusieurs flux. Ces flux peuvent être unicast ou multicast. Les transmissions sur les liens sont considérées comme fiables, ce qui exclut toute perte de paquet. Les sources partagent la même horloge, mais les flux produits ne sont pas synchronisés. Ils sont indépendants et peuvent éventuellement être inactifs. Les flux produits par les sources sont considérés comme une séquence des paquets de même longueur L . Nous considérons que tous les paquets d'un flux suivent le même chemin dans le réseau.

Le deuxième ensemble de nœuds se compose des *nœuds de codage*. Ces nœuds peuvent effectuer des opérations de codage réseau linéaires. Ces nœuds se composent d'éléments de réseau tels que des files d'attente et/ou des modeleurs et chaque élément garantit un niveau de service.

Le troisième ensemble de nœuds se compose des *nœuds récepteurs* qui reçoivent et décodent les flux combinés.

Sous ces hypothèses, nous considérons qu'un code réseau est déterminé à l'avance, ce qui implique que chaque nœud de codage connaît la ou les combinaisons linéaires à réaliser sur les flux en entrée de telle sorte à produire les flux de sortie.

Ces conditions particulières nécessitent une stratégie de codage particulière sur les nœuds de codage. Par exemple, un nœud ne doit pas "bloquer" un paquet indéfiniment dans le but de le combiner avec un paquet arrivant d'un autre flux car ceci aurait des conséquences dramatiques sur son délai de bout-en-bout.

La stratégie de codage doit tenir compte à la fois des intérêts des flux et du réseau. En effet, d'une part, elle doit assurer un niveau minimal de combinaison dans les flux sortants afin de diminuer la quantité totale de données transmises sur les liens sortants.

Mais d'autre part, elle doit éviter de contraindre fortement les flux en retardant les paquets afin de les combiner avec des paquets d'autres flux.

Dans cette section, nous proposons trois stratégies de codage réseau :

- La **première stratégie de codage (Stratégie Orientée Réseau)** est décrite par une architecture de nœud basée sur des modeleurs de type seau percé (leaky bucket shapers) et des files d'attente. Les flux sont considérés comme une séquence continue de paquets non synchronisés. Les paquets sont codés en fonction de leur date d'arrivée dans chaque nœud. Comme les flux ne sont pas synchronisés, les dates d'arrivée sont aléatoires. Ce codage peut donc être considéré comme un "codage aléatoire". Il faut noter que, contrairement au codage réseau classique (tel que présenté dans la section 2.2.3.3), l'aspect aléatoire de ce code concerne seulement les dates d'arrivée des paquets et non pas les choix aléatoires des coefficients.

L'avantage principal de cette stratégie est sa souplesse. En contre-partie, l'aspect aléatoire de ce code ne permet pas d'assurer une fiabilité totale sur le décodage des paquets par les récepteurs.

L'étude de cette stratégie est faite au niveau d'un nœud de codage. Elle est ensuite généralisée au niveau d'un réseau complet. Elle permet en particulier d'obtenir des bornes supérieures sur les délais de bout-en-bout au niveau des récepteurs. Cette généralisation est faite par l'utilisation d'une matrice de transfert définie via l'algèbre min-plus du calcul réseau. Cette stratégie a été présentée à la conférence ISITA 2006 [63].

- La **deuxième stratégie de codage (Stratégie Orientée Flux)** utilise le concept de code en bloc pour assurer aux récepteurs que le décodage puisse être réalisé. Nous considérons que chaque source produit au maximum un paquet par flux dans l'intervalle $[t_i, t_i + \Delta[$. Les paquets produits par les sources durant cet intervalle forment les paquets d'information du $i^{\text{ème}}$ bloc (mot de code).

Le code réseau linéaire combine seulement les paquets appartenant au même bloc. Pour effectuer une combinaison linéaire de plusieurs paquets, chaque nœud doit attendre tous les paquets du même bloc. Puisque le réseau fournit des garanties sur les délais, un nœud est capable de calculer une limite de temps ou un délai fixé pour la réception d'un bloc donné.

Si un paquet d'un bloc donnée n'est pas reçu avant le délai fixé ou la limite de son bloc, le nœud suppose que la source n'a pas produit de paquet dans cet intervalle de temps et effectue des combinaisons avec les paquets disponibles. Nous étudions cette stratégie au niveau du nœud puis au niveau réseau. Cette stratégie était le sujet de l'article [64] présenté à conférence Networking 2007.

- La **troisième stratégie de codage** présentée dans [65] est appelée **Stratégie de Transfert Rapide (Fast Forwarding Strategy)**. Comme dans la deuxième stratégie, nous utilisons le concept de code en bloc. La principale différence est que les paquets sont autorisés de quitter le nœud de codage même si tous les paquets de ce bloc transitant par ce nœud n'ont pas encore atteint ce nœud. Cette approche permet de réduire les délais d'attente inutiles des paquets dans les routeurs.

Finalement, à titre de comparaison, l'**approche classique de routage/ multiplexage** est aussi étudiée dans cette section.

Plusieurs stratégies introduites dans cette section seront comparées à partir d'une évaluation théorique dans le chapitre 5 et par simulation au chapitre 6.

4.2 Stratégie Orientée Réseau (NOS : Network-Oriented Strategy)

4.2.1 Introduction

Le but de la stratégie présentée dans cette partie est de réduire au maximum la quantité de données stockée dans le réseau (c'est-à-dire dans les files d'attente des nœuds de codage) tout en assurant un niveau de codage suffisant afin de réduire le débit des flux en sortie. Cette stratégie assure que le débit maximal des flux en sortie est le maximum des débits des flux d'entrée. L'idée principale consiste à combiner les paquets des différents flux présents dans un nœud au même moment. Une des conséquences est que certains paquets sont transmis sans être combinés avec d'autres paquets. Toutefois, ils sont multipliés par le coefficient déterminé par le code réseau.

Dans cette partie, nous étudierons tout d'abord les délais de transmission et le backlog à l'aide de l'algèbre min-plus du calcul réseau. Cette approche est ensuite généralisée au niveau d'un réseau complet en utilisant une matrice de transfert définie dont les éléments appartiennent à l'algèbre min-plus du calcul réseau.

4.2.2 Définitions et hypothèses

Considérons un nœud de codage avec n flux en l'entrée et un flux en sortie. Ces flux sont respectivement représentés par leurs fonctions cumulatives R_1, R_2, \dots, R_n et R^{out} . Les flux d'entrée sont indépendants, non-synchronisés et composés de paquets de longueur L . Ils peuvent être temporairement inactifs. Nous considérons que les liens ont une capacité infinie (cette contrainte sera discutée plus tard).

On peut noter que les résultats présentés dans cette section peuvent être facilement généralisés aux nœuds ayant plusieurs flux en sortie.

Nous considérons qu'un flux d'entrée R_i , (pour $i = 1, 2, \dots, n$) est contraint par une courbe d'arrivée α_i , où $\alpha_i(t) = \sigma_i + L * v_{L/\rho_i, -L/\rho_i}(t)$ pour $t > 0$ et 0 autrement.

Cette fonction correspond à une fonction en escalier avec une taille de backlog (ou arriéré de trafic) σ_i et un débit moyen maximal ρ_i (voir la Figure 4.1).

Nous considérons que les valeurs de $\sigma_1, \sigma_2, \dots, \sigma_n$ sont des multiples de L . Nous supposons aussi que $\rho = \max(\rho_1, \rho_2, \dots, \rho_n)$ et $T = L/\rho = \min(T_1, T_2, \dots, T_n)$ où $T_1 = L/\rho_1, T_2 = L/\rho_2, \dots, T_n = L/\rho_n$.

Avant de présenter de nouvelles définitions, il est important de noter que le flux de sortie est composé de paquets construits comme des combinaisons linéaires des paquets issus des flux entrants. A tout flux entrant, on peut faire correspondre le sous-ensemble de paquets du flux sortant contenant dans sa combinaison linéaire un paquet provenant de ce flux entrant. Ces "sous-flux" sont utilisés dans les définitions suivantes.

Nous définissons :

4.2. Stratégie Orientée Réseau (NOS : Network-Oriented Strategy)

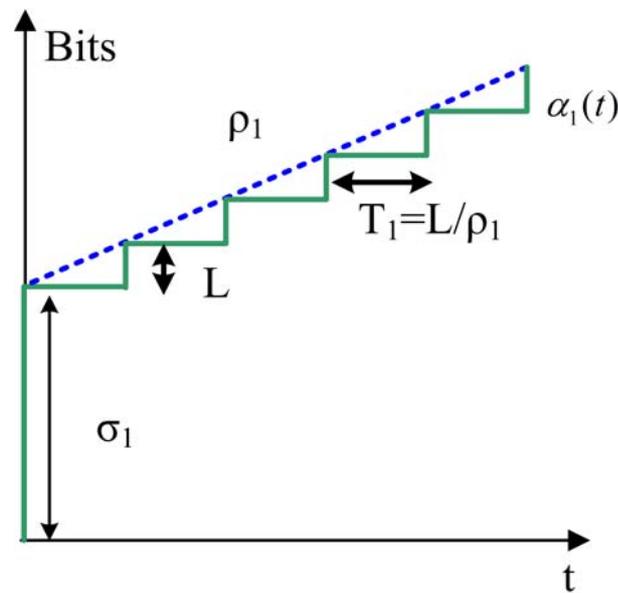


FIGURE 4.1 – Courbe d'arrivée du flux F_1 en forme d'escalier

- R_i^{out} comme la fonction cumulative du sous-ensemble de données de R_i^{out} obtenu à partir des données de R_i .
- Le délai de traversée subi par un paquet de données dans un nœud comme la différence entre le moment d'arrivée de ce paquet et le moment de son départ du nœud (sous la forme d'un paquet combiné).
- L'arriéré de trafic (backlog) d'un flux dans le nœud à l'instant t comme la quantité de données "en transit" dans le nœud.

4.2.3 Architecture d'un nœud de codage

L'architecture que nous proposons est représentée sur la Figure 4.2. Un *Nœud de Codage* est composé de n modeleurs de type seau percé glouton, $LBS_1, LBS_2, \dots, LBS_n$, et n files d'attente, B_1, B_2, \dots, B_n .

Le flux R_i (pour $i = 1, 2, \dots, n$) traverse le modeleur LBS_i et la file d'attente B_i en séquence.

La fonction de LBS_i consiste à modeler le flux F_i tel qu'il y ait au moins un intervalle de temps T entre deux paquets. LBS_i a un backlog égal à σ_i et offre une courbe de service $L * v_{T,0}(t)$ au flux F_i .

Les files d'attente B_1, B_2, \dots, B_n sont *synchronisées* et offrent la même courbe de service $L * v_{T,-T}(t)$ aux différents flux. En d'autres termes, après chaque intervalle de temps T , toutes les files d'attente font sortir un paquet (si elle en contiennent au moins un).

Les n paquets sortants des files d'attente sont alors multipliés par un coefficient scalaire et additionnés pour former le paquet codé.

Les opérations de codage ajoutent un délai supplémentaire. Nous considérons que

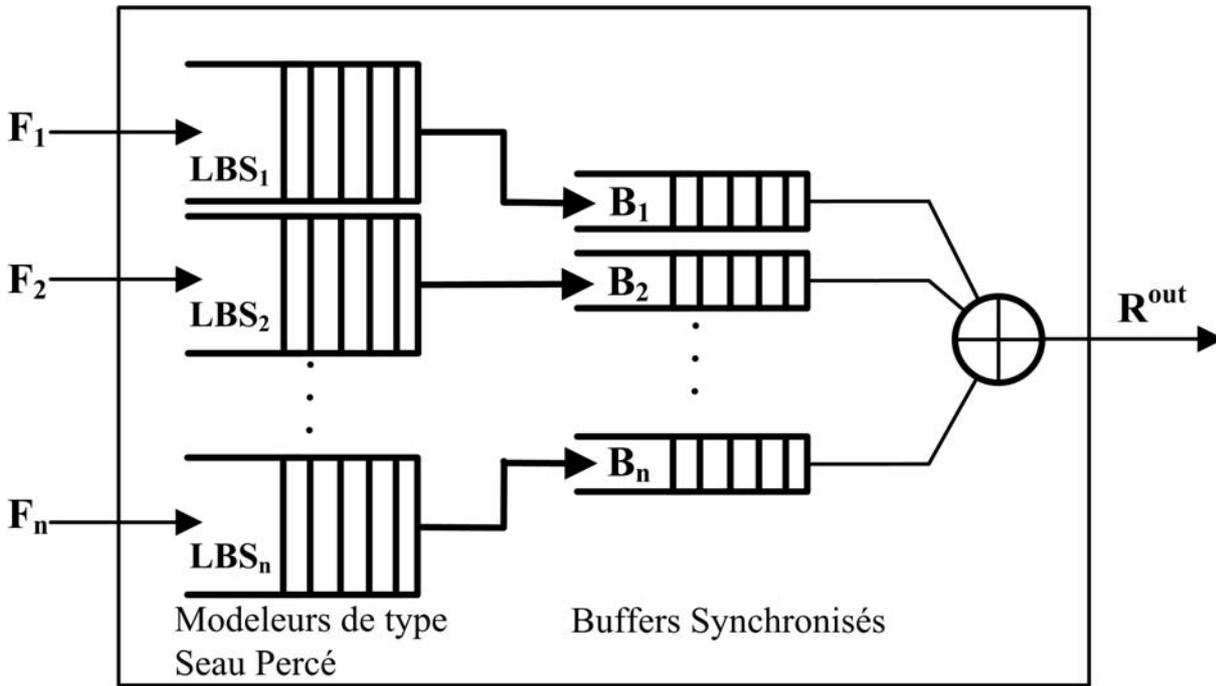


FIGURE 4.2 – Architecture d'un nœud de codage

ce délai est borné supérieurement par T_{lc} et ne dépend pas de nombre de paquets combinés. Nous considérons donc que la courbe de service de ces opérations est $\delta_{T_{lc}}$ et que $0 < T_{lc} \leq T$.

Si une des files d'attente ne contient aucun paquet, les paquets quittant les autres files d'attente sont multipliés par des coefficients scalaires, combinés et expédiés. On peut considérer que le paquet manquant est virtuellement remplacé par un paquet de zéros.

Remarque : Dans cette stratégie, nous avons supposé que les files d'attente sont synchronisées. En fait, une solution alternative peut être considérer en supposant que les modeleurs sont, en plus de leurs tâches traditionnelles, capable de se synchroniser pour faire sortir simultanément les paquets.

4.2.4 NOS au niveau d'un nœud de codage

Considérons l'architecture de nœud présentée ci-dessus :

Théorème 1. *Il existe une politique de service pour un nœud de codage qui assure pour $i = 1, 2, \dots, n$, que :*

1. *La courbe de service β_i fournie par le nœud à R_i est égale à $L * v_{T, -T - T_{lc}}(t)$.*
2. *L'arriéré de trafic (backlog) du flux R_i présent dans le nœud est $\sigma_i + L$.*
3. *Le délai maximal subi par un paquet de R_i est $T_{lc} + T(1 + \sigma_i/L)$.*

4.2. Stratégie Orientée Réseau (NOS : Network-Oriented Strategy)

4. R^{out} est contraint par

$$(\alpha_1 \otimes L * v_{T,-T-T_{lc}}) \vee (\alpha_2 \otimes L * v_{T,-T-T_{lc}}) \vee \dots \vee (\alpha_n \otimes L * v_{T,-T-T_{lc}})$$

Démonstration. 1). Par définition, le flux qui sort de LBS_i est égal à $R_i(t) \otimes L * v_{T,0}(t)$. Ce flux est ensuite servi par la file d'attente B_i avec une courbe de service $L * v_{T,-T}(t)$ et combiné avec un service $\delta_{T_{lc}}$. Il s'ensuit que

$$\begin{aligned} R_i^{out}(t) &\geq \{R_i(t) \otimes L * v_{T,0}(t)\} \otimes L * v_{T,-T}(t) \otimes \delta_{T_{lc}} \\ &\geq R_i(t) \otimes \{L * v_{T,0}(t) \otimes L * v_{T,-T}(t) \otimes \delta_{T_{lc}}\} \\ &\geq R_i(t) \otimes L * v_{T,-T-T_{lc}}(t) \end{aligned}$$

Ainsi, $L * v_{T,-T-T_{lc}}(t)$ est une courbe de service offerte par le nœud à R_i .

2). D'après le point 9 - Section 3.4, l'arriéré de trafic (backlog) du flux R_i , noté $b_i(t)$, est tel que :

$$\begin{aligned} b_i(t) &\leq \sup_{s \geq 0} \{\alpha_i(s) - L * v_{T,-T-T_{lc}}(s)\} \\ &\leq \sup_{s \geq 0} \{\sigma_i + L * v_{T_i,-T_i}(s) - L * v_{T,-T-T_{lc}}(s)\} \\ &\leq \sigma_i + L \end{aligned}$$

3). D'après les points 10, 11 de la Section 3.4, le délai maximal, noté $d_i(t)$ subi par un paquet du flux R_i est borné supérieurement par :

$$\begin{aligned} d_i(t) &\leq \sup_{s \geq 0} \{\inf\{\tau \geq 0 : \alpha_i(s) \leq L * v_{T,-T-T_{lc}}(s + \tau)\}\} \\ &\leq \sup_{s \geq 0} \{\inf\{\tau \geq 0 : \sigma_i + L * v_{T_i,-T_i}(s) \leq L * v_{T,-T-T_{lc}}(s + \tau)\}\} \\ &\leq T_{lc} + \frac{\sigma_i + L}{\rho} = T_{lc} + T(1 + \sigma_i/L) \end{aligned}$$

(voir la Figure 3.10).

4). Puisque

- Les files d'attente B_1, B_2, \dots, B_n sont synchronisées.
- Les paquets qui quittent les n files d'attente sont combinés.
- Une courbe d'arrivée de R_i^{out} est égale à

$$\alpha_i(t) \otimes \beta_i(t) = \alpha_i(t) \otimes \{L * v_{T,0} \otimes L * v_{T,-T} \otimes \delta_{T_{lc}}\} = \alpha_i \otimes L * v_{T,-T-T_{lc}}$$

Les paquets des tous les flux quittent les files d'attente avec une parfaite synchronisation. Dans ce contexte, les paquets des différents flux, qui arrivent exactement au même moment, sont combinés ensemble formant le flux sortant R^{out} . Ces flux

Chapitre 4. Nouvelles stratégies de codage réseau

sortant sont alors des flux des paquets de taille L , avec un intervalle de temps multiple de T entre deux paquets consécutifs. En conséquence, à chaque instant, le flux total qui sort du nœud est égal au maximum des flux entre tous les flux R_i^{out} . Il est donc nécessairement contraint par :

$$(\alpha_1 \otimes L * v_{T,-T-T_{lc}}) \vee (\alpha_2 \otimes L * v_{T,-T-T_{lc}}) \vee \dots \vee (\alpha_n \otimes L * v_{T,-T-T_{lc}})$$

□

Remarque : Les hypothèses considérées sur la capacité des liens étaient que tous les liens ont une capacité infinie. Selon l'architecture du nœud de codage proposée, nous pouvons réduire sans dommage la capacité du lien de sortie à $\max_{i=1,2,\dots,n}(\rho_i)$

4.2.5 NOS au niveau d'un réseau

Le but de cette section est de généraliser les travaux précédents au niveau d'un réseau complet. Nous allons analyser, puis déterminer les propriétés et les limites supérieures des délais de transmission d'un paquet dans un réseau composé de nœuds ayant l'architecture et les propriétés décrites dans la section précédente.

Une matrice de transfert de ce réseau sera également présentée en utilisant les notions de calcul réseau.

Considérons un réseau représenté par un graphe acyclique orienté $G = (V, E)$ avec un ensemble de nœuds $V = \{v_1, \dots, v_m\}$ et un ensemble de liens $E = \{e_1, \dots, e_p\}$. Dans cette section, nous supposons que les transmissions d'un paquet sur un lien sont immédiates (délai égal à 0). Cette hypothèse sera modifiée par la suite.

Nous assignons à chaque lien e_i une capacité C_{e_i} et nous considérons que s nœuds parmi le m sont des nœuds sources et q sont des nœuds récepteurs.

Nous supposons que les nœuds sources produisent des flux R_i , $i = 1, \dots, k$, respectivement contraints par une courbe d'arrivée α_i .

Un nœud de codage combine ses flux d'entrée pour produire les flux de sortie selon un code réseau linéaire déterminé *a priori*. Nous considérons que les flux d'entrée de ce nœud R_1, \dots, R_r sont respectivement contraints par les courbes d'arrivée $\alpha_{1,in}, \dots, \alpha_{r,in}$. Nous notons $\beta_{i,j}$, ($i = 1, \dots, r$ et $j = 1, \dots, g$) la courbe de service offerte par le nœud de codage au flux d'entrée R_i vers le $j^{\text{ème}}$ flux de sortie $R_{j,out}$.

Suivant le point 4) du théorème 1, $R_{j,out}$ est contraint par la courbe d'arrivée $\alpha_{j,out}$ définie par :

$$\alpha_{j,out} = \alpha_{1,in} \otimes \beta_{1,j} \vee \alpha_{2,in} \otimes \beta_{2,j} \vee \dots \vee \alpha_{r,in} \otimes \beta_{r,j}$$

Nous supposons que les nœuds récepteurs produisent des flux R_i^* , $i = 1, \dots, n$, respectivement contraints par une courbe d'arrivée α_i^* . Chaque récepteur offre aussi des courbes de service à ses flux d'entrée vers ses flux de sortie.

Un exemple d'un tel réseau est illustré sur la Figure 4.3.

La généralisation au niveau du réseau des résultats obtenus au niveau des nœuds revient à déterminer le service offert par le réseau aux flux d'entrée vers les récepteurs.

4.2. Stratégie Orientée Réseau (NOS : Network-Oriented Strategy)

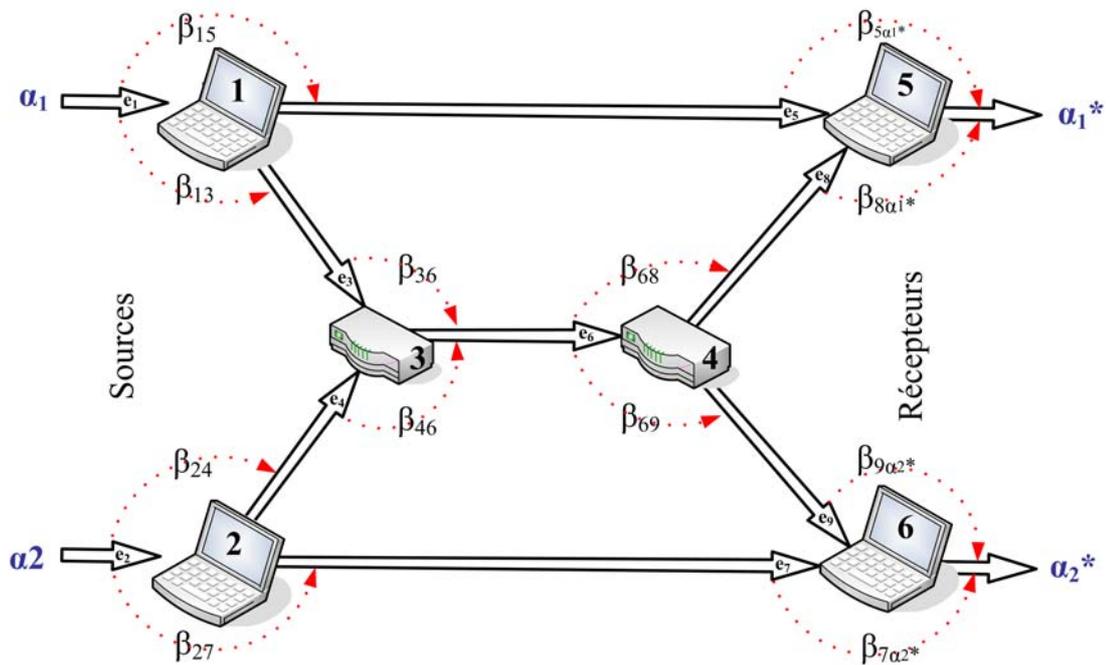


FIGURE 4.3 – Un réseau acyclique, orienté avec des liens sans délais et avec deux sources qui produisent les flux d'entrée R_1, R_2 avec des courbes d'arrivée respective α_1 et α_2 . Les deux récepteurs sont les nœuds (5) et (6) qui produisent les flux de sortie R_1^*, R_2^* avec des courbes d'arrivée respective α_1^* et α_2^* . Les nœuds (1), (2), (3) et (4) sont des *nœuds de codage*. Chaque nœud offre une courbe de service donnée à ses flux d'entrée vers ses flux de sortie.

Chapitre 4. Nouvelles stratégies de codage réseau

En d'autres termes, nous cherchons à définir une matrice de transfert M permettant d'exprimer les courbes d'arrivée des flux de sortie en fonction de celles des flux d'entrée. Ceci peut être représenté par :

$$[\alpha_1^*, \dots, \alpha_n^*] = [\alpha_1, \dots, \alpha_k] \circ M \quad (4.1)$$

où M représente une matrice de transfert.

La construction de la matrice de transfert proposée ici suit la méthode de la construction de la matrice de transfert d'un code réseau linéaire présentée dans [2]. La différence majeure est que, dans [2], les nœuds de codage effectuent des combinaisons linéaires dans un corps fini du type :

$$Y = X_1 * \beta_1 + X_2 * \beta_2$$

où X_1 , X_2 et Y sont des processus aléatoires qui représentent les flux et β_1 et β_2 appartiennent à un corps fini.

Dans notre contexte, les nœuds effectuent des opérations sur des courbes de service et d'arrivée, telles par exemple,

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

où X_1 , X_2 et Y sont des courbes d'arrivée et β_1 et β_2 sont des courbes de service.

Par rapport à la construction de la matrice définie sur les corps finis, la principale difficulté ici est de définir les opérations algébriques devant être appliquées pour "multiplier" des vecteurs de courbes d'arrivées par des matrices de courbes de service.

Tout d'abord, nous rappelons trois règles de calcul réseau :

- 1). $(\alpha \circ \beta_1) \circ \beta_2 = \alpha \circ (\beta_1 \otimes \beta_2)$
- 2). $(\alpha \circ \beta_1) \vee (\alpha \circ \beta_2) = \alpha \circ (\beta_1 \wedge \beta_2)$
- 3). $(\alpha_1 \vee \alpha_2) \circ \beta = (\alpha_1 \circ \beta) \vee (\alpha_2 \circ \beta)$

Les points 1) et 3) sont directement tirés de [10]. L'équation 2) peut être directement obtenue de la définition de l'opération $\circ : (f \circ g)(t) = \sup_{u \geq 0} [f(t+u) - g(u)]$

En effet, nous trouvons

$$\begin{aligned} ((\alpha \circ \beta_1) \vee (\alpha \circ \beta_2))(t) &= (\alpha \circ \beta_1)(t) \vee (\alpha \circ \beta_2)(t) \\ &= \sup_{u \geq 0} [\alpha(t+u) - \beta_1(u)] \vee \sup_{u \geq 0} [\alpha(t+u) - \beta_2(u)] \\ &= \sup_{u \geq 0} [\alpha(t+u) - (\beta_1 \wedge \beta_2)(u)] \\ &= \alpha \circ (\beta_1 \wedge \beta_2) \end{aligned}$$

Les deux premiers points concernent des opérations incluant seulement une courbe d'arrivée et plusieurs courbes de service. Ces opérations sont, en règle générale, celles qui doivent être employées pour calculer la multiplication des matrices des courbes de service. Un autre avantage de ces opérations est qu'elles permettent la concaténation des courbes de service.

Le dernier point inclut deux courbes d'arrivée. Ainsi, il sera utilisé dans la multiplication du vecteur des courbes d'arrivée et de la matrice des courbes de service (une multiplication matrice-vecteur).

4.2. Stratégie Orientée Réseau (NOS : Network-Oriented Strategy)

Rappelons que p , k et n représentent respectivement le nombre de liens, de flux d'entrée et de flux de sortie. Nous dirons qu'un lien (ou un flux) i est connecté à un lien (ou un flux) j si la tête du lien i est la queue du lien j .

Dans [2], la matrice de transfert M est définie comme : $M = A * (I - F)^{-1} * B^T$ où les matrices A et B sont respectivement les matrices d'entrée et de sortie. La matrice F est la matrice d'adjacence. Définissons ces trois matrices dans notre contexte.

La matrice A représente les courbes de service appliquées sur les flux en entrée du réseau. Nous prenons comme convention de noter i le lien faisant entrer le flux R_i dans le réseau (voir Figure 4.3). Avec cette convention la matrice $A = (a_{i,j})_{i=1,\dots,k;j=1,\dots,p}$ est définie comme suit : si le lien i est connecté au lien j , alors $a_{i,j} = \beta_{i,j}$ où $\beta_{i,j}$ est la courbe de service offerte par le nœud au flux provenant du lien i et transmis sur le lien j . Si les liens ne sont pas connectés, $a_{i,j} = 0$, ce qui correspond à un service nul.

Pour l'exemple présenté sur la Figure 4.3, la matrice A est :

$$A = \begin{bmatrix} 0 & 0 & \beta_{13} & 0 & \beta_{15} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_{24} & 0 & 0 & \beta_{27} & 0 & 0 \end{bmatrix}$$

Nous définissons ensuite la matrice d'adjacence $F = (f_{i,j})_{i=1,\dots,p;j=1,\dots,p}$ comme suit : Si le lien i est connecté au lien j , alors $f_{i,j} = \beta_{i,j}$ où $\beta_{i,j}$ est la courbe de service offerte par le nœud au flux provenant du lien i et transmis (codé) sur le lien j .

Pour le réseau de la Figure 4.3, nous avons :

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_{36} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_{46} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_{68} & \beta_{69} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Intuitivement, la multiplication d'un vecteur d'entrée par M indique l'état des flux d'entrée après un saut (hop - le passage dans un nœud). De même, la multiplication d'un vecteur par M^i indique l'état des flux d'entrée après i sauts.

Puisque le graphe G est acyclique, la matrice d'adjacence peut être représentée comme une matrice triangulaire strictement supérieure. Elle est alors nilpotente. Nous pouvons calculer la matrice $(I - F)^{-1} = [I + F + F^2 + F^3 + \dots]$ qui indique les états des flux d'entrée dans le réseau.

Chapitre 4. Nouvelles stratégies de codage réseau

Pour le réseau de la Figure 4.3, $(I - F)^{-1}$ est égal à :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \beta_{36} & 0 & \beta_{36} \otimes \beta_{68} & \beta_{36} \otimes \beta_{69} & 0 \\ 0 & 0 & 0 & 1 & 0 & \beta_{46} & 0 & \beta_{46} \otimes \beta_{68} & \beta_{46} \otimes \beta_{69} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \beta_{68} & \beta_{69} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

La matrice B représente les courbes de service appliquées sur les flux sortant du réseau. Donc, la matrice $B = (b_{i,j})_{i=1,\dots,p;j=1,\dots,n}$ est définie comme suit :

Si le lien i est connecté au lien sortant j , alors $b_{i,j} = \beta_{i,j}$. Si les liens ne sont pas connectés, $b_{i,j} = 0$, ce qui correspond à un service nul.

Pour l'exemple présenté sur la Figure 4.3, la matrice B est égale à :

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & \beta_{5\alpha_1^*} & 0 & 0 & \beta_{8\alpha_1^*} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta_{7\alpha_2^*} & 0 & \beta_{9\alpha_2^*} \end{bmatrix}$$

La matrice de transfert est alors égale à :

$$M = A \times (I - F)^{-1} \times B^T$$

Pour le réseau de la Figure 4.3, $A \times (I - F)^{-1}$ est égale à :

$$\begin{bmatrix} 0 & 0 & \beta_{13} & 0 & \beta_{15} & \beta_{13} \otimes \beta_{36} & 0 & \beta_{13} \otimes \beta_{36} \otimes \beta_{68} & \beta_{13} \otimes \beta_{36} \otimes \beta_{69} \\ 0 & 0 & 0 & \beta_{24} & 0 & \beta_{24} \otimes \beta_{46} & \beta_{27} & \beta_{24} \otimes \beta_{46} \otimes \beta_{68} & \beta_{24} \otimes \beta_{46} \otimes \beta_{69} \end{bmatrix}$$

Finalement, M est égale à :

$$\begin{bmatrix} \beta_{15} \otimes \beta_{5\alpha_1^*} \wedge \beta_{13} \otimes \beta_{36} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*} & \beta_{13} \otimes \beta_{36} \otimes \beta_{69} \otimes \beta_{9\alpha_2^*} \\ \beta_{24} \otimes \beta_{46} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*} & \beta_{27} \otimes \beta_{7\alpha_2^*} \wedge \beta_{24} \otimes \beta_{46} \otimes \beta_{69} \otimes \beta_{9\alpha_2^*} \end{bmatrix}$$

Maintenant, nous pouvons donc exprimer les courbes d'arrivée des flux de sortie en fonctions de celles des flux d'entrée :

$$\begin{bmatrix} \alpha_1^* & \alpha_2^* \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix} \otimes M$$

On peut ainsi obtenir une expression de la courbe d'arrivée de chacun des flux de sortie. Par exemple :

$$\begin{aligned} \alpha_1^* &= \{\alpha_1 \otimes (\beta_{15} \otimes \beta_{5\alpha_1^*} \wedge \beta_{13} \otimes \beta_{36} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*})\} \vee \{\alpha_2 \otimes (\beta_{24} \otimes \beta_{46} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*})\} \\ &= \{\alpha_1 \otimes (\beta_{15} \otimes \beta_{5\alpha_1^*}) \vee \alpha_1 \otimes (\beta_{13} \otimes \beta_{36} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*})\} \vee \{\alpha_2 \otimes (\beta_{24} \otimes \beta_{46} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*})\} \\ &= \alpha_1 \otimes (\beta_{15} \otimes \beta_{5\alpha_1^*}) \vee \alpha_1 \otimes (\beta_{13} \otimes \beta_{36} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*}) \vee \alpha_2 \otimes (\beta_{24} \otimes \beta_{46} \otimes \beta_{68} \otimes \beta_{8\alpha_1^*}) \end{aligned}$$

En effet, le codage réseau permet d'obtenir ces résultats même si le lien 6 a une capacité finie de l'ordre de débit maximum des flux.

4.2.6 Discussion sur le décodage

L'approche utilisée dans la cette stratégie est définie de telle sorte à optimiser l'utilisation des ressources réseau telles que le temps passé par les paquets dans les nœuds de codage et la taille des files d'attente.

Même si nous supposons que le code réseau est fixe, c.-à-d. que les nœuds de codage effectuent les mêmes opérations linéaires sur les flux d'entrée, la variabilité du débit, les différences de longueur entre les multiples chemins et la gigue impliquent que le codage ait un aspect aléatoire.

Il s'ensuit que, même si le code réseau est fixe, le récepteur doit décoder un code aléatoire. En d'autres termes, les combinaisons de données observées par un nœud récepteur ne sont pas toujours identiques.

L'analyse de la performance de décodage de cette approche n'est pas traitée dans ce document. Toutefois, plusieurs travaux ont montré que le codage aléatoire peut fournir un niveau statistique de fiabilité aussi important que souhaité [6]. Le paramètre important est principalement la taille du corps fini qui influe sur la probabilité d'inversion du système linéaire correspondant aux paquets codés reçus [6] [66].

Il faut noter que ce niveau de fiabilité peut être amélioré soit en ajoutant des chemins redondants dans le code fixe de réseau ou en envoyant des paquets FEC redondants de la source [6][5] [67].

Malgré tout, le coté aléatoire de cette stratégie est gênante pour un réseau censé fournir des garanties de qualité de service. Pour cette raison, nous avons souhaité proposer d'autres stratégies fournissant des garanties (sections 4.3 et 4.4). Le "prix" de ces garanties est l'utilisation du concept de code en bloc qui complexifie le système global.

4.3 Stratégie Orientée Flux (FOS : Flow-Oriented Strategy)

4.3.1 Introduction

Comme indiqué précédemment, le premier but de cette stratégie est de fournir une fiabilité totale. Ceci est réalisé en utilisant le concept de code en bloc introduit dans [5]. Ce concept implique que les combinaisons de paquets ne peuvent être réalisées qu'entre des paquets appartenant au même bloc.

4.3.2 Définitions et hypothèses

Nous reprenons les définitions et les hypothèses de réseau présentées dans la partie 4.1. Nous définissons un lien orienté reliant le nœud v_i au nœud v_j par $e_{i,j}$. Chaque lien $e_{i,j}$ a une capacité $C_{i,j}$ (*bits/sec*), ce qui signifie qu'un paquet de L bits sera transmis en $L/C_{i,j}$ secondes. Comme le système doit fournir des garanties de QoS, nous considérons que, pour chaque lien $e_{i,j}$, le délai de transmission maximum d'un paquet de L bits est connu et égal à $L/C_{i,j} + T_{i,j} = \omega_{i,j} + T_{i,j}$.

Chapitre 4. Nouvelles stratégies de codage réseau

En d'autres termes, le lien $e_{i,j}$ a la courbe de service de taux-latence $\beta_{C_{i,j}, T_{i,j}}(t)$. Nous supposons que la capacité du lien sortant est supérieure ou égale au maximum des capacités d'entrée. En d'autres termes, la capacité de lien sortant doit supporter, sans congestion, tous les débits des liens d'entrée. Classiquement, avec le codage réseau, il suffit que ce lien sortant ait une capacité de l'ordre du débit maximum des flux d'entrée (c.-à-d. dans le pire cas, la capacité du lien sortant doit être supérieure ou égale à la capacité maximum des capacités des liens d'entrée).

Il faut noter qu'un des intérêts du codage réseau est d'améliorer la performance de débit global dans des réseaux où justement, des congestions apparaissent lorsque l'on utilise une stratégie de routage classique. Dans cette thèse, nous avons choisi de prendre cette hypothèse de réseaux sur-provisionnés pour illustrer le fait que le codage réseau permet d'obtenir des gains sur la garantie de délai, même lorsqu'il n'améliore pas les débits.

Comme dans la partie précédente, nous supposons que le temps nécessaire pour effectuer une combinaison linéaire de plusieurs paquets est assez petit, qu'il ne dépend pas du nombre de paquets et qu'il est borné supérieurement par T_{lc} . Nous supposons qu'un code réseau linéaire a été déterminé *a priori* pour le réseau considéré. En conséquence, chaque nœud sait comment combiner les flux d'entrée pour produire des flux de sortie.

Les flux générés par les sources sont composés de paquets de longueur fixe L . Ils vérifient deux contraintes. La première est liée à la notion du bloc. Nous supposons que toutes les sources découpent le temps en intervalles $[t_i, t_{i+1}[$ de longueur fixe Δ ($t_{i+1} = t_i + \Delta$). Dans chacun de ces intervalles de temps, chaque source génère au plus un paquet. L'ensemble des paquets générés par les différentes sources dans le même intervalle de temps $[t_i, t_{i+1}[$ constitue l'ensemble des paquets d'information du $i^{\text{ème}}$ bloc (mot de code). Si certaines sources ne génèrent pas de paquets dans cet intervalle de temps, le paquet d'information de cette source est considéré comme nul.

Dans cette stratégie, les nœuds intermédiaires ne combinent que des paquets du même bloc, cela implique qu'ils combinent des paquets générés quasi-simultanément par les sources. Ces nœuds réalisent les mêmes opérations de codage sur chaque bloc. Chaque récepteur reçoit donc des blocs qui ont tous été codés de la même manière (en utilisant les mêmes combinaisons linéaires). Les opérations linéaires réalisés par ce récepteur pour décoder sont donc les mêmes pour tous les blocs.

Au niveau des flux, il faut noter que le fait que les sources génèrent au plus un paquet par intervalle de temps impose une contrainte sur le débit et le niveau de variabilité du flux. En effet, avec cette contrainte, le débit maximal du flux est $\rho = L/\Delta$.

La deuxième contrainte sur le flux est dictée par une courbe d'arrivée affine (voir les points 2 et 3 - dans la Section 3.4). Le débit et le niveau de variabilité imposés par cette courbe peuvent correspondre à celles imposées par la notion du bloc.

4.3.3 FOS au niveau d'un nœud de codage

Considérons un nœud intermédiaire avec n flux d'entrée et un flux sortant (voir la Figure 4.4). Avec les hypothèses présentées dans la Section 4.3.2, ce nœud combine

4.3. Stratégie Orientée Flux (FOS : Flow-Oriented Strategy)

les paquets des flux d'entrée issus du même bloc en employant la stratégie de codage suivante. La combinaison linéaire correspondante à un bloc i est effectuée dès que au moins l'un des points suivants est vérifié :

- tous les paquets du bloc i sont dans les files d'attente
- un ou plusieurs paquets du bloc i ne sont pas dans les files d'attente et la date-limite du temps d'arrivée du bloc i est excédée.
- un ou plusieurs paquets du bloc i ne sont pas dans leurs files d'attente et des paquets du bloc $i + 1$ sont dans ces files d'attente.

Les deux derniers points indiquent que le flux correspondant ne contient pas tous les paquets du bloc i . Dans ce cas, la combinaison linéaire est seulement faite avec les paquets du bloc i présentes dans le nœud. Ceci n'aura aucun incidence sur le décodage final car le paquet décodé correspondant sera simplement le paquet plein de zéros.

Notons que chaque nœud connaît la date-limite d'arrivée des paquets de chaque bloc. En effet, chaque lien fournit des délais de transmission garantis, et chaque nœud fournit des délais de service garantis. De proche en proche, un nœud est capable de calculer la date-limite d'arrivée d'un paquet de chaque bloc.

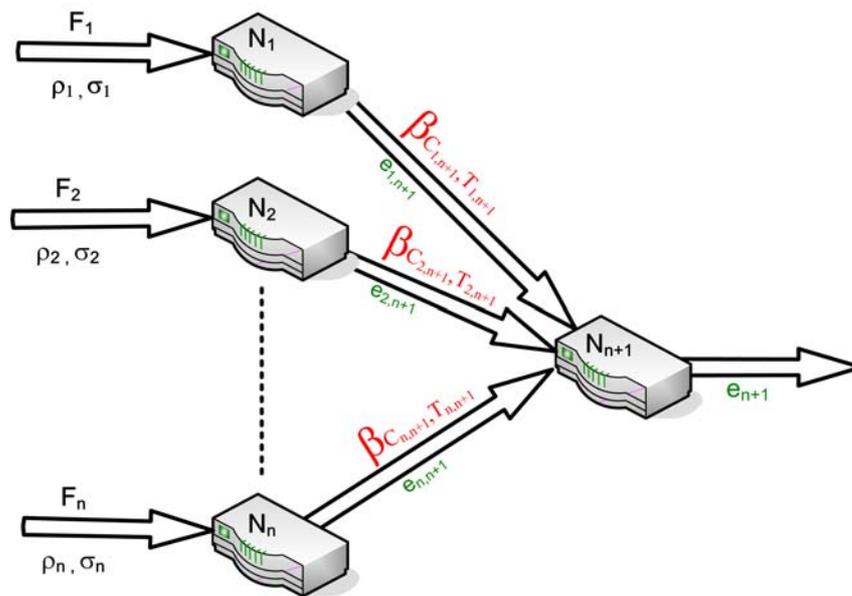


FIGURE 4.4 – Nœud avec n flux d'entrée

4.3.3.1 Analyse des délais

Les délais subis par les paquets dans un nœud ont plusieurs causes.

D'abord, pour effectuer une combinaison linéaire de plusieurs paquets, le nœud doit attendre les paquets suivant les règles définies dans le paragraphe précédent.

Le temps maximum passé par un paquet du flux F_i dans le nœud N_{n+1} pour attendre des paquets du même bloc qui arrivent d'autres liens est dénoté par $T_{B_i}^{n+1}$. Nous représentons ce délai avec la courbe de service $\beta_{B_i}^{n+1}(t) = \delta_{T_{B_i}^{n+1}}(t)$.

Chapitre 4. Nouvelles stratégies de codage réseau

En second lieu, la combinaison linéaire des paquets ajoute un délai borné supérieurement par T_{lc} . La courbe de service associée à cette opération est $\delta_{T_{lc}}$.

Finalement, le traitement du flux codé dans un nœud intermédiaire N_{n+1} est représenté par une courbe de service $\beta_{C_{n+1}, \tau_{n+1}}(t) = C_{n+1}(t - \tau_{n+1})$ où C_{n+1} est la capacité du lien sortant et τ_{n+1} est le temps de service de ce flux dans ce nœud.

Nous notons $\beta_{n+1}^i(t)$ la courbe de service totale fournie par le nœud pour combiner et transmettre un paquet appartenant au flux F_i . Puisque la courbe de service d'un nœud intermédiaire est la convolution des courbes de service offertes par ses éléments, la courbe de service offerte par un nœud intermédiaire N_{n+1} à un flux F_i est donc :

$$\begin{aligned}\beta_{n+1}^i(t) &= \beta_{C_{n+1}, \tau_{n+1}}(t) \otimes \delta_{T_{B_i}^{n+1}} \otimes \delta_{T_{lc}}(t) \\ &= \beta_{C_{n+1}, \tau_{n+1} + T_{B_i}^{n+1} + T_{lc}}\end{aligned}$$

Le service fourni par les nœuds récepteurs est égal à celui fourni par les nœuds intermédiaires. En effet, le décodage du code réseau est simplement une combinaison linéaire des paquets d'entrée. Ainsi, le problème est identique à celui des nœuds intermédiaires.

4.3.3.2 Délai maximum dans un nœud intermédiaire de premier ordre

Pour présenter ce calcul, considérons la Figure 4.4. Nous considérons que le nœud intermédiaire est de premier ordre, c'est à dire qu'il reçoit tous ses flux d'entrée directement à partir des sources. Chaque flux F_i , $i = 1, \dots, n$, est contraint par $\alpha_{\rho_i, \sigma_i}$. Ces flux sont transmis sur les liens $\{e_{1, n+1}, \dots, e_{n, n+1}\}$ vers un nœud intermédiaire N_{n+1} où ils seront combinés. La courbe de service du lien $e_{i, n+1}$ est $\beta_{C_{i, n+1}, T_{i, n+1}}$.

Considérons un paquet produit dans l'intervalle de temps $[t, t + \Delta]$. Supposons que le temps de traversée du nœud N_i par le flux F_i soit nul et que le délai de transmission sur le lien $e_{i, n+1}$ soit dans l'intervalle $[\omega_{i, n+1}, T_{i, n+1} + \omega_{i, n+1}]$. Il s'ensuit que le temps d'arrivée le plus court de ce paquet dans ce nœud est $t + \omega_{i, n+1}$ et le temps d'arrivée le plus tardif est $t + \Delta + T_{i, n+1} + \omega_{i, n+1}$.

Le temps d'attente maximum d'un paquet du flux F_i dans la file d'attente de réception B_i est

$$T_{B_i}^{n+1} = \Delta + \max_{\substack{j=1, \dots, n \\ j \neq i}} [T_{j, n+1} + \omega_{j, n+1}] - [\omega_{i, n+1}]$$

Pour un nœud situé à plusieurs "sauts" des sources, la dernière formule peut être interprétée comme suit : le délai maximum d'attente, qui peut être subi par un paquet d'un bloc quelconque du flux F_i dans une file d'attente d'un nœud de codage est la somme de la durée de l'intervalle d'un bloc Δ et la différence entre le délai maximum subi par les paquets de ce bloc et le délai minimum subi par ce paquet.

La courbe de service associée est $\beta_{B_i}^{n+1}(t) = \delta_{T_{B_i}^{n+1}}(t)$. Les opérations liées à la combinaison linéaire et la transmission offrent respectivement les courbes de service $\beta_{T_{lc}}(t) = \delta_{T_{lc}}(t)$ et $\beta_{C_{n+1}, \tau_{n+1}}(t) = C_{n+1}(t - \tau_{n+1})$ où C_{n+1} est la capacité de e_{n+1} .

Supposons que $C_{i, n+1} \leq C_{n+1}$ et $\rho_i \leq C_{i, n+1}$ pour $i = 1, 2, \dots, n$. La courbe de service totale offerte au flux F_i par le nœud codage N_{n+1} est $\beta_{n+1}^i(t) = \beta_{C_{n+1}, T_i^*}$, où $T_i^* = T_{B_i}^{n+1} + T_{lc} + \tau_{n+1}$.

4.3. Stratégie Orientée Flux (FOS : Flow-Oriented Strategy)

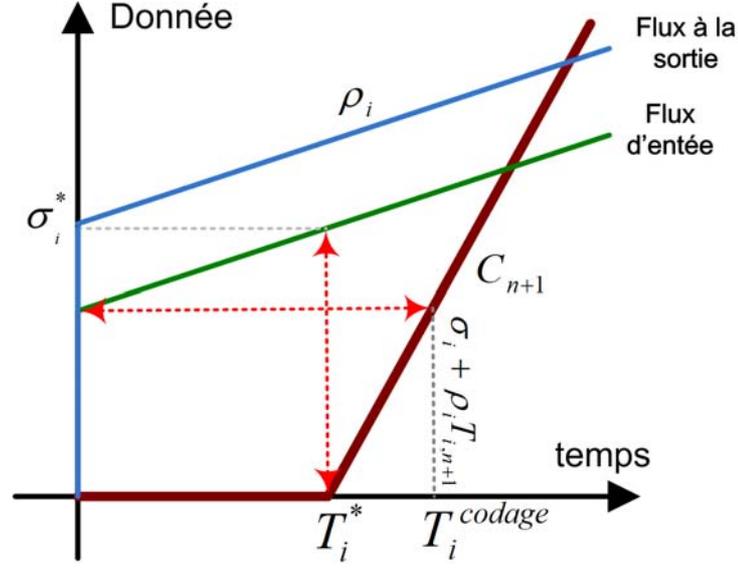


FIGURE 4.5 – Courbe de sortie

Le délai maximum passé par un paquet de F_i dans ce nœud est

$$T_i^{\text{codage}} = T_{B_i}^{n+1} + T_{lc} + \tau_{n+1} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}} \quad (4.2)$$

Une courbe d'arrivée du sous-flux du flux sortant correspondant à F_i en sortie de ce nœud intermédiaire est $[\alpha_{\rho_i, \sigma_i^*}]$ où

$$\sigma_i^* = \sigma_i + \rho_i T_{i,n+1} + \rho_i (T_{B_i}^{n+1} + T_{lc} + \tau_{n+1})$$

Cette valeur de σ_i^* correspond à la quantité de données maximale du flux F_i stockée dans le nœud de codage (arriéré de trafic).

4.3.4 FOS au niveau d'un réseau

Le but de cette section est d'étendre les résultats précédents et d'analyser les délais au niveau d'un réseau. Considérons un flux d'entrée F_i , contraint par $\alpha_{\rho_i, \sigma_i}(t)$, vérifiant les contraintes décrites dans la section précédente et transmis successivement à travers n nœuds intermédiaires et n liens.

On considère que chaque lien $e_{i,k}$ dans ce chemin offre une courbe de service $\beta_{C_{i,k}, T_{i,k}}(t) = C_{i,k}(t - T_{i,k})$. La courbe de service offerte par un nœud intermédiaire N_k au flux F_i vers le lien sortant $e_{k,r}$ est égale à :

$$\begin{aligned} \beta_{C_{k,r}, T_i^*}(t) &= \beta_{B_i}^k(t) \otimes \beta_{T_{lc}}(t) \otimes \beta_{C_{k,r}, \tau_k}(t) \\ &= C_{k,r}[t - T_i^*] \end{aligned}$$

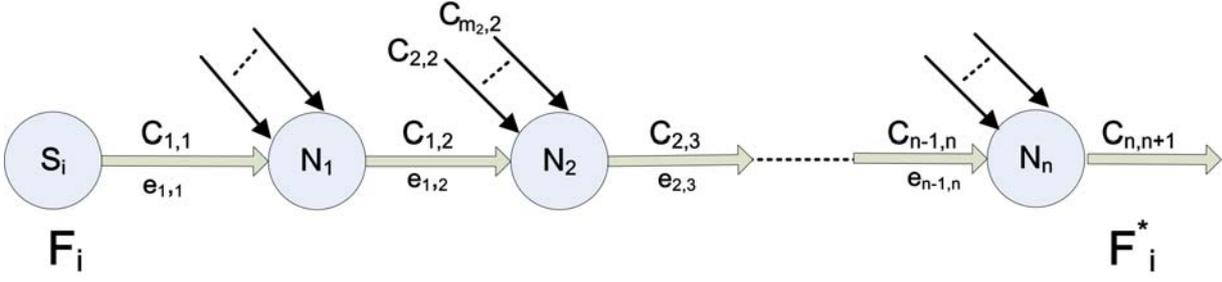


FIGURE 4.6 – Transmission d'un flux à travers n nœuds.

où

$$T_i^* = T_{B_i}^k + T_{lc} + \tau_k$$

Dans ce cas $T_{B_i}^k$ sera, comme nous l'avons montré dans la partie précédente, donnée par

$$T_{B_i}^k = \Delta + \max_{\substack{j=1,\dots,m_k \\ j \neq i}} \{T_{j,k} + \omega_{j,k} + [A_{j,k}^X]_{max}\} - \{\omega_{i,k} + [A_{i,k}^X]_{min}\} \quad (4.3)$$

Les dates minimales et maximales de sortie du nœud N_i sont notées $[A_{i,k}^X]_{min}$ et $[A_{i,k}^X]_{max}$.

Ensuite, la courbe de service offerte par le chemin total à un paquet du flux F_i est

$$\beta_{C_{tot}^i, T_{tot}^i}^i(t)$$

où

$$T_{tot}^i = T_{1,1} + \sum_{k=1}^{n-1} T_{k,k+1} + \sum_{k=1}^n \{T_{B_i}^k + T_{lc} + \tau_k\} \quad (4.4)$$

et

$$C_{tot}^i = \min(C_{1,1}, C_{1,2}, \dots, C_{n-1,n}) \quad (4.5)$$

Puis, le délai maximum d'un paquet du flux F_i , contraint par $\alpha_{\rho_i, \sigma_i}$, de la source jusqu'à la sortie de ce chemin est égal à

$$T_i^{codage} = T_{tot}^i + \frac{\sigma_i}{C_{tot}^i}$$

Finalement, la courbe d'arrivée de F_i à la fin de ce chemin est :

$$\alpha_{\rho_i, \sigma_i^*}^i(t) = \alpha_{\rho_i, \sigma_i}^i(t) \otimes \beta_{C_{tot}^i, T_{tot}^i}^i(t)$$

où

$$\rho_i \leq C_{tot}^i \quad \sigma_i^* = \sigma_i + \rho_i * T_{tot}^i$$

4.3.5 Discussion

Nous pouvons étendre le travail précédent pour calculer les tailles des files d'attente nécessaires à chaque routeur et à chaque récepteur. Cette taille de file d'attente sera toujours inférieure ou égale au backlog dans n'importe quel nœud.

Par exemple, la taille de file d'attente pour le flux F_i à la fin de chemin précédent (nœud N_n) est inférieure ou égale à $[\sigma_i + \rho_i * T_{tot}^i]$.

Nous notons également que la valeur de Δ , qui définit la longueur des intervalles des blocs, est très importante car elle joue un rôle essentielle dans le calcul des délais et des tailles de files d'attente. En effet, pour un débit fixé, une grande valeur de Δ assure le codage d'un grand nombre de paquets dans les nœuds intermédiaires. En contrepartie, les délais de bout-en-bout et les tailles des files d'attente seront aussi augmentées. Par exemple, ce paramètre doit être choisi avec soin car il apparaît au niveau de chaque nœud intermédiaire (équation 4.3), ce qui implique qu'il apparaît N fois au niveau du délai observé par un récepteur 4.4.

Globalement, cette stratégie est une implémentation "naturelle" du codage réseau classique. Elle est bien adaptée à des flux régulier qui produisent des paquets dans chaque bloc.

Par contre, avec des flux irréguliers, les nœuds intermédiaires feront attendre les paquets d'un flux jusqu'à la date limite, ce qui entraînera des délais de bout-en-bout importants.

4.4 Stratégie de Transfert Rapide (FFS : Fast Forwarding Strategy)

4.4.1 Introduction

Le système présenté dans la Section 4.3 a été conçu pour fonctionner avec un certain nombre de flux. Lorsque tous les flux sont actifs, c'est-à-dire que des paquets sont générés par les sources dans la plupart des générations, ce système fonctionne bien. Toutefois, lorsque certains flux sont inactifs, les autres flux doivent les "attendre" dans les nœuds de codage et, par conséquent, leurs délais de bout-en-bout sont augmentés. L'amélioration que nous proposons permet de résoudre ce problème en autorisant les paquets à quitter le nœud de codage, même si l'ensemble du bloc n'est pas arrivé. Cette stratégie est appelée *stratégie de transfert rapide*.

4.4.1.1 Objectifs et définition de la stratégie de transfert rapide

Comme indiqué dans le paragraphe précédent, l'objectif de cette stratégie est d'assouplir la stratégie FOS de telle sorte à ce que des flux inactifs ne retardent pas les autres flux.

Une solution classique pour assouplir le fonctionnement du codage réseau de pour l'appliquer dans des contextes pratiques est l'utilisation du codage réseau aléatoire.

Cette approche permet au codage réseau de s'adapter à la topologie du réseau, à la variabilité du débit des flux ainsi que de supporter des pertes de paquets tout en conservant ses principaux avantages. Cette approche a été implémentée avec succès dans le contexte de réseaux maillés sans fil, de réseaux ad-hoc ou de réseaux pair-à-pair. Son principe ainsi que ses principales applications sont détaillées dans la partie 2.2.3.3.

Dans le contexte de cette thèse, nous avons considéré une topologie de réseau fixe et une connaissance de tous les paramètres du réseau. Nous avons aussi considéré un taux de perte nul. Sous ces hypothèses, le principal intérêt du codage réseau aléatoire est sa capacité d'adaptation à la variation du débit des flux.

La stratégie orientée réseau, (NOS- section 4.2) peut être vue comme une implémentation de cette approche dans un mode continu, c'est-à-dire dans le cas où l'on ne distingue pas de génération ou de blocs. Comme indiqué précédemment, le principal défaut de cette stratégie dans notre contexte est qu'elle n'est pas déterministe. Une amélioration de la stratégie NOS pourrait consister à intégrer la notion de bloc (ou de génération). On pourrait alors borner les probabilités d'échec du décodage $((1 - d/q)^\eta)$ où q est la taille du corps fini, η est le nombre de liens et d est le nombre des récepteurs) et les rendre aussi petite que souhaité en choisissant un corps fini suffisamment grand. Malgré tout, cette approche ne peut pas être considérée comme déterministe dans l'absolu, ce qui eut être rédhibitoire dans certains domaines d'application (tels que les réseaux avioniques).

Pour cette raison, nous n'avons pas utilisé le concept de codage réseau aléatoire. Nous proposons donc dans cette partie une stratégie de codage déterministe ayant l'avantage d'être plus souple que la stratégie FOS.

Cette stratégie, dite de transfert rapide, hérite de FOS de l'utilisation d'un code fixe et du concept de code en bloc. La principale différence est que, grâce à l'intégration d'une somme de contrôle dans les paquets, les nœuds de codage intermédiaires sont autorisés à réaliser des combinaisons de paquets partielles et à transmettre ces combinaisons partielles dès que des opportunités de transmission se présentent.

4.4.2 Définitions et hypothèses

Les hypothèses sur le réseau sont les mêmes que dans la Section 4.3.2. Toutes les sources sont synchronisées et dans chaque intervalle de temps $[t_i, t_{i+1}[$, au plus un paquet est généré par chaque source. Tous les paquets produits pendant le même intervalle de temps portent le même numéro de bloc.

En plus des hypothèses faites dans le système précédent, nous considérons que chaque paquet contient une somme de contrôle (checksum) calculée sur la charge utile entière de paquet. Nous proposons d'employer une somme de contrôle classique utilisée dans les couches hautes de protocoles IP, comme *par exemple* celui d'UDP [68]. Puisqu'il est défini comme "le complément à un de la somme sur 16 bits des compléments à un" des données, il est basé sur les sommes de nombres entiers de 16 bits (modulo 2^{16}), et ainsi il n'est pas linéaire dans aucun sous-corps du corps employé par le code réseau (nous supposons que ce corps a la forme \mathbb{F}_{2^m}). Il s'ensuit que la combinaison linéaire (par le code réseau) des sommes de contrôle de deux paquets n'est pas égale

4.4. Stratégie de Transfert Rapide (FFS : Fast Forwarding Strategy)

avec une grande probabilité à la somme de contrôle de la même combinaison linéaire des deux paquets. Notons que contrairement à la somme de contrôle, les CRC (cyclic redundancy check - voir, par exemple, [69]) sont linéaires sur le corps \mathbb{F}_2 . En effet, puisque $\mathbb{F}_2 \subset \mathbb{F}_{2^m}$, les combinaisons linéaires dans \mathbb{F}_{2^m} sont également des combinaisons linéaires dans \mathbb{F}_2 . La propriété de non linéarité de la somme de contrôle est utilisée par le récepteur pour vérifier la validité d'un décodage actuel (voir la Section 4.4.6).

La structure du nœud intermédiaire est également modifiée. La différence principale est que les nœuds contiennent seulement une file d'attente. La gestion de cette file d'attente avec la nouvelle stratégie est décrite dans le prochain paragraphe.

4.4.3 Présentation de la stratégie FFS

Comme indiqué précédemment, le but de cette stratégie est d'implémenter un codage "totalement" fiable tout en évitant les temps d'attente inutiles des paquets dans les routeurs. Ces temps d'attente sont le plus souvent dûs au fait que toutes les sources n'ont pas généré de paquets dans l'intervalle de temps associé à ce bloc. Le nœud de codage intermédiaire doit alors attendre la date limite d'arrivée des paquets de cette génération pour s'apercevoir qu'aucun paquet n'a été généré et finalement pour transmettre la combinaison linéaire réalisée avec les paquets de ce bloc qui sont arrivées.

A cette observation, il faut rajouter le fait que dans un système pratique, le lien de communication n'est pas forcément utilisée en permanence. Le codage réseau, qui permet d'avoir un gain de débit en transmettant une combinaison linéaire de plusieurs paquets au lieu de transmettre séparément les différents paquets, n'est alors pas utile.

L'idée principale de cette approche est de faire en sorte que les paquets soient combinés uniquement lorsque cela est nécessaire, c'est-à-dire lorsque le nœud intermédiaire est engorgé.

Cette stratégie peut être décrite à partir de l'exemple présenté sur la Figure 4.7. La sous-figure (a) décrit le fonctionnement classique implémenté par la stratégie FOS.

Au temps t_1 , les paquets P_1 et P_2 sont combinés par le nœud N_1 pour produire le paquet $P'_{1,1}$. Cette combinaison est réalisée dans tous les cas, même si les deux paquets n'arrivent pas en même temps. Dans ce cas, N_1 attend l'autre paquet pour réaliser la combinaison linéaire.

Au temps t_2 , le même type d'opération est réalisé par le nœud N_2 avec les paquets $P'_{1,1}$, P_3 et P_4 pour produire le paquet final P''_1 envoyé au récepteur.

La sous-figure (b) illustre le fonctionnement de l'algorithme FFS dans les cas où les paquets atteignent les nœuds de codage à des moments différents. Au temps t_1 , le paquet P_1 est présent dans N_1 . Si le lien de sortie est libre, N_1 code P_1 en $P'_{1,1}$ comme si P_2 ne devait pas arriver.

Au temps t_2 , P_2 atteint le nœud N_1 . Ce dernier code alors P_2 en générant $P'_{1,2}$. Dans le même temps, N_2 combine $P'_{1,1}$ et P_3 (c'est à dire les paquets du même bloc présents à cet instant) pour produire P''_1 .

Au temps t_3 , N_2 combine les paquets $P'_{1,2}$ et P_4 pour produire P''_2 et le transmettre vers le récepteur R .

Chapitre 4. Nouvelles stratégies de codage réseau

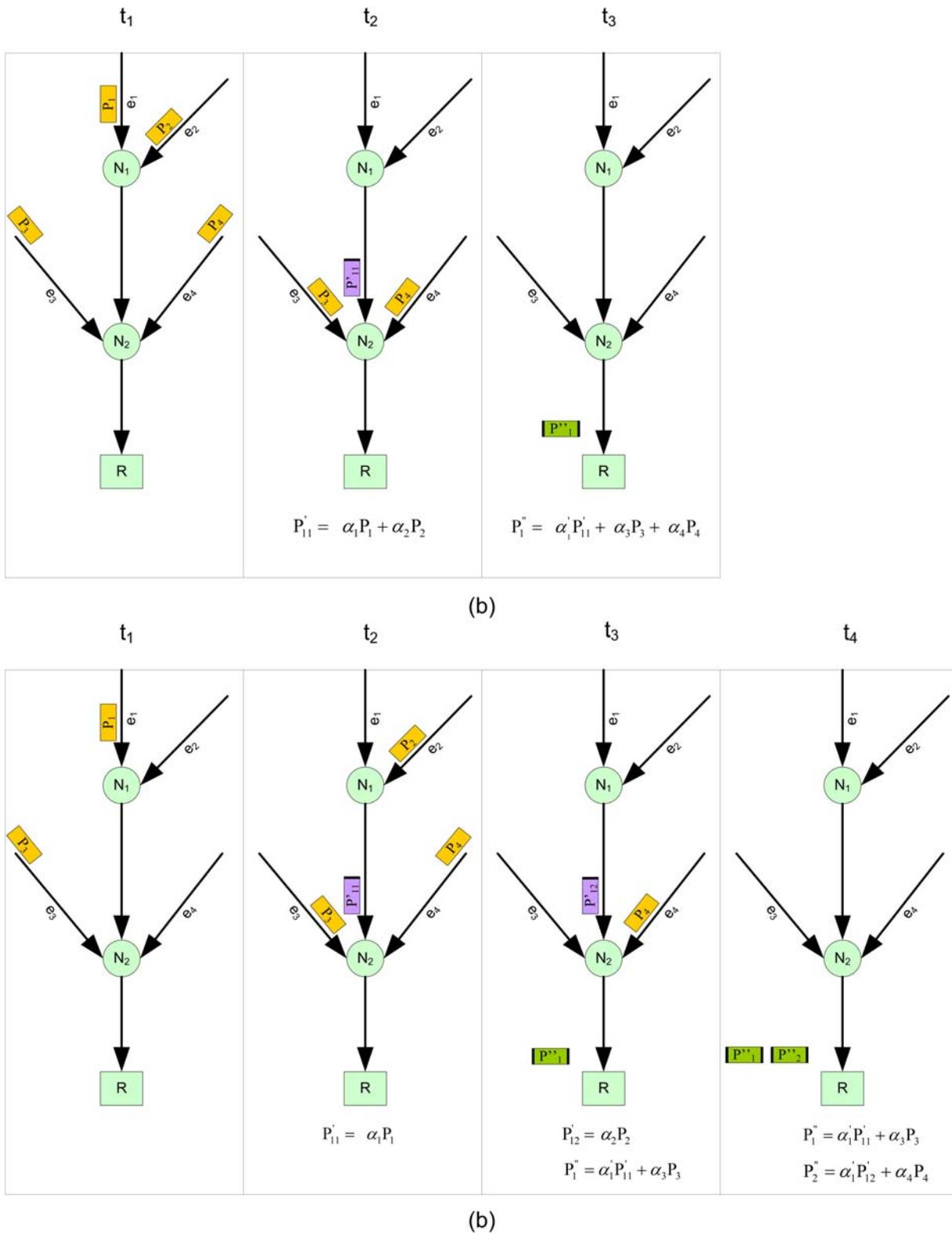


FIGURE 4.7 – Les fonctionnements de : (a)- la stratégie orientée flux. (b)- la stratégie de transfert rapide.

4.4. Stratégie de Transfert Rapide (FFS : Fast Forwarding Strategy)

Au temps t_4 , le récepteur reçoit P_2'' . Il peut alors l'additionner à P_1'' . Le résultat obtenu est alors exactement le même que si les opérations avaient été réalisées par la stratégie FOS.

Le gain de la stratégie FFS par rapport à FOS peut être observé dans le cas où aucun paquet n'est généré par les sources sur les liens e_2 (c'est-à-dire P_2) et e_4 (c'est-à-dire P_4). Avec FFS, le décodage peut être réalisé au temps t_3 dès la réception du paquet P_1'' (le processus permettant de détecter qu'un paquet reçu par le décodeur peut être utilisé par le décodeur est décrit dans la partie 4.4.6). Avec la stratégie FOS, les nœuds N_1 et N_2 auraient dû attendre la date d'arrivée maximale des paquets de cette génération pour transmettre les paquets combinés, ce qui aurait généré des délais supplémentaires.

On peut toutefois observer que la stratégie FFS utilise plus de ressources réseau que FOS. Par exemple, deux paquets du même bloc ($P'_{1,1}$ et $P'_{1,2}$) sont transmis par FFS sur le lien entre N_1 et N_2 alors que FOS n'en transmet qu'un seul. En fait, ce point n'est pas spécialement gênant car si le lien avait été saturé, cela aurait provoqué des délais d'attente des paquets dans N_1 , ce qui aurait peut-être permis à P_2 d'arriver dans N_1 avant que $P'_{1,1}$ n'ait été transmis. Il aurait alors été intégré dans la combinaison linéaire $P'_{1,1}$. De la même manière, si N_2 avait été congestionné au temps t_2 , cela aurait peut-être permis à $P'_{1,2}$ et P_4 d'atteindre le nœud avant que la combinaison linéaire P_1'' ne soit transmise. Ils auraient alors été intégrés à ce paquet.

L'avantage principal de la stratégie FFS est de combiner un certain niveau de souplesse avec une fiabilité totale. En effet, de la même manière que les stratégies de codage aléatoire classiques, elle combine tous les paquets (d'un bloc) présents dans un nœud à un instant donné, mais en plus, comme les opérations réalisées sont celles indiquées par un code réseau fixé à l'avance, elle permet d'assurer une fiabilité totale car les paquets finalement reçus par le décodeur sont les mêmes que ceux générés par la stratégie de codage classique (FOS).

La manière dont cette stratégie est implémentée dans les nœuds, et notamment la méthode permettant de gérer simultanément plusieurs générations dans un nœud, est présentée dans la partie suivante.

4.4.4 FFS au niveau d'un nœud de codage

Après avoir illustré le fonctionnement de FFS sur un exemple, décrivons-le dans le cas général.

Considérons un nœud intermédiaire avec n flux d'entrée et un flux sortant (voir la Figure 4.8). Considérons les hypothèses sur le réseau présentées dans les Sections 4.3.2 et 4.4.2. Supposons qu'un paquet d'un bloc donné X arrive dans un nœud de codage N_{n+1} à l'instant t . La stratégie de transfert rapide de ce nœud de codage est la suivante :

- Si la file d'attente est vide, le paquet est multiplié par un coefficient de corps fini déterminé par le code réseau et il est directement transmis sur le lien sortant (si ce lien n'est pas occupé par la transmission d'autre paquets qui a déjà commencé avant le temps t).
- Si la file d'attente n'est pas vide :
 - S'il n'y a aucun paquet du bloc X dans la file d'attente, le paquet est multi-

plié par son coefficient de corps fini correspondant et ajouté à la fin de la file d'attente. Par exemple, sur la Figure 4.8, le paquet P_3^1 arrivant du nœud N_1 est ajouté à la fin de la file d'attente.

- S'il y a un paquet du bloc X dans la file d'attente, le paquet arrivant est multiplié par son coefficient de corps fini correspondant et est additionné au paquet de son bloc dans la file d'attente. Par exemple, sur la Figure 4.8, le paquet P_5^2 arrivant du nœud N_2 est additionné au paquet P_5^1 déjà présent dans la file d'attente.

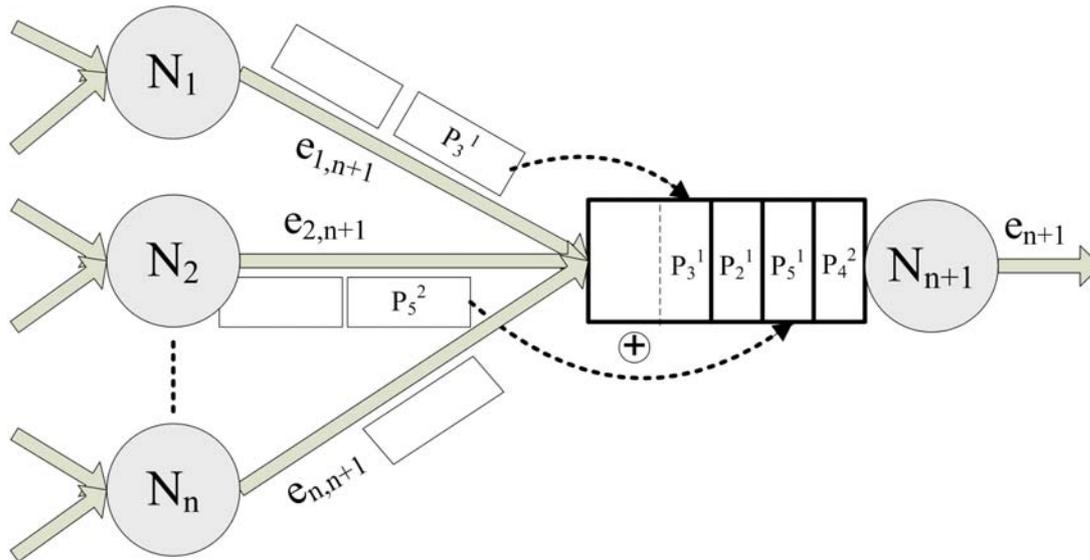


FIGURE 4.8 – Stratégie de transfert rapide

Notons que cette stratégie peut mener au déséquencement des blocs (par exemple, dans la Figure 4.8, un paquet du bloc 4 sera traité avant un paquet du bloc 2).

Pour estimer les délais de bout-en-bout et la taille de file d'attente, il faut déterminer le délai maximum subi par un paquet dans un nœud intermédiaire. De la stratégie décrite précédemment, on peut déduire que le paquet doit attendre au plus le temps nécessaire pour transmettre le nombre maximal de blocs différents qui peuvent se trouver simultanément dans le nœud intermédiaire (lorsque le paquet arrive au nœud). Le temps d'arrivée à chaque nœud intermédiaire et les temps inter-bloc, (un temps inter-bloc se définit comme la durée séparant deux blocs consécutifs dans un nœud intermédiaire), sont utilisés pour calculer ce nombre.

4.4.4.1 Temps d'arrivée à un nœud intermédiaire

Chaque bloc a des temps d'arrivée minimum et maximum prédéterminés à chaque nœud. Chaque nœud peut évaluer ces temps d'arrivée pour un paquet d'un bloc donné en demandant aux nœuds précédents les temps d'envoi minimums et maximums et en considérant des délais additionnels de transmission (donnés par les courbes de service des liens).

4.4. Stratégie de Transfert Rapide (FFS : Fast Forwarding Strategy)

Par exemple, considérons le nœud dans la Figure 4.8 comme un nœud intermédiaire de premier ordre (c.-à-d. il reçoit ses flux d'entrée directement des sources). Le temps d'arrivée d'un bloc X , qui est généré dans l'intervalle $[t, t + \Delta[$, à l'entrée de ce nœud intermédiaire N_{n+1} sera borné par :

$$t_X = [t + \min_{i=1, \dots, n} \{\omega_{i, n+1}\}, t + \Delta + \max_{i=1, \dots, n} \{T_{i, n+1} + \omega_{i, n+1}\}]$$

Notons

$$[t_X]_{\min} = t + \min_{i=1, \dots, n} \{\omega_{i, n+1}\}$$

$$[t_X]_{\max} = t + \Delta + \max_{i=1, \dots, n} \{T_{i, n+1} + \omega_{i, n+1}\}$$

Supposons maintenant que le nœud N_{n+1} soit un nœud intermédiaire à l'intérieur du réseau. Le temps d'arrivée à l'entrée de ce nœud d'un bloc X , généré dans l'intervalle $[t, t + \Delta[$, est borné par :

$$t_X = [t + \min\{\omega_{i, n+1} + [A_{i, n+1}^X]_{\min}\}, t + \Delta + \max\{T_{i, n+1} + \omega_{i, n+1} + [A_{i, n+1}^X]_{\max}\}]$$

pour $i = 1, 2, \dots, n$, où

$$[t_X]_{\min} = t + \min_{i=1, \dots, n} \{\omega_{i, n+1} + [A_{i, n+1}^X]_{\min}\}$$

$$[t_X]_{\max} = t + \Delta + \max_{i=1, \dots, n} \{T_{i, n+1} + \omega_{i, n+1} + [A_{i, n+1}^X]_{\max}\}$$

Nous définissons $[A_{i, n+1}^X]_{\min}$ et $[A_{i, n+1}^X]_{\max}$ comme les délais minimum et maximum subis par un paquet d'un bloc X de la source jusqu'à l'entrée du lien $e_{i, n+1}$.

4.4.4.2 Durée séparant deux blocs consécutifs dans un nœud intermédiaire

Nous pouvons calculer le temps qui sépare deux blocs consécutifs $\{X - 1, X\}$ en exploitant les résultats précédents.

Pour un bloc X , le temps d'arrivée minimum à un nœud N_{n+1} sur un lien $e_{i, n+1}$ (voir la Figure 4.8) est $t + \min\{\omega_{i, n+1} + [A_{i, n+1}^X]_{\min}\}$.

Sur le même lien, le temps d'arrivée minimum du bloc $X - 1$ est $t - \Delta + \min\{\omega_{i, n+1} + [A_{i, n+1}^{X-1}]_{\min}\}$.

En tenant compte du fait que $[A_{i, n+1}^X]_{\min} = [A_{i, n+1}^{X-1}]_{\min}$, nous pouvons facilement remarquer que le temps qui sépare deux temps d'arrivée minimums de deux blocs consécutifs est égal à Δ .

Nous trouvons le même type de résultat pour le temps qui sépare les temps d'arrivée maximums de deux blocs consécutifs. Celui-ci est aussi égal à Δ .

4.4.4.3 Taille de file d'attente dans un nœud intermédiaire

Le nombre maximum de paquets présentes dans la file d'attente d'un nœud intermédiaire à l'arrivée d'un paquet du bloc X détermine la taille maximum de cette file d'attente.

Chapitre 4. Nouvelles stratégies de codage réseau

En fait, avec cette stratégie, chaque bloc est présenté par au plus un paquet dans la file d'attente. Donc, le nombre maximum de paquets présents est égal au le nombre maximum de blocs présentes dans cette file d'attente. Ce nombre peut être déduit des temps d'arrivée du bloc et du temps qui sépare les blocs.

A partir des ces constatations, on peut facilement montrer que le nombre maximum de blocs est $\{[t_X]_{max} - [t_X]_{min}\} / \Delta$. La taille maximum de la file d'attente dans ce nœud intermédiaire est donc $\{[t_X]_{max} - [t_X]_{min}\} L / \Delta$.

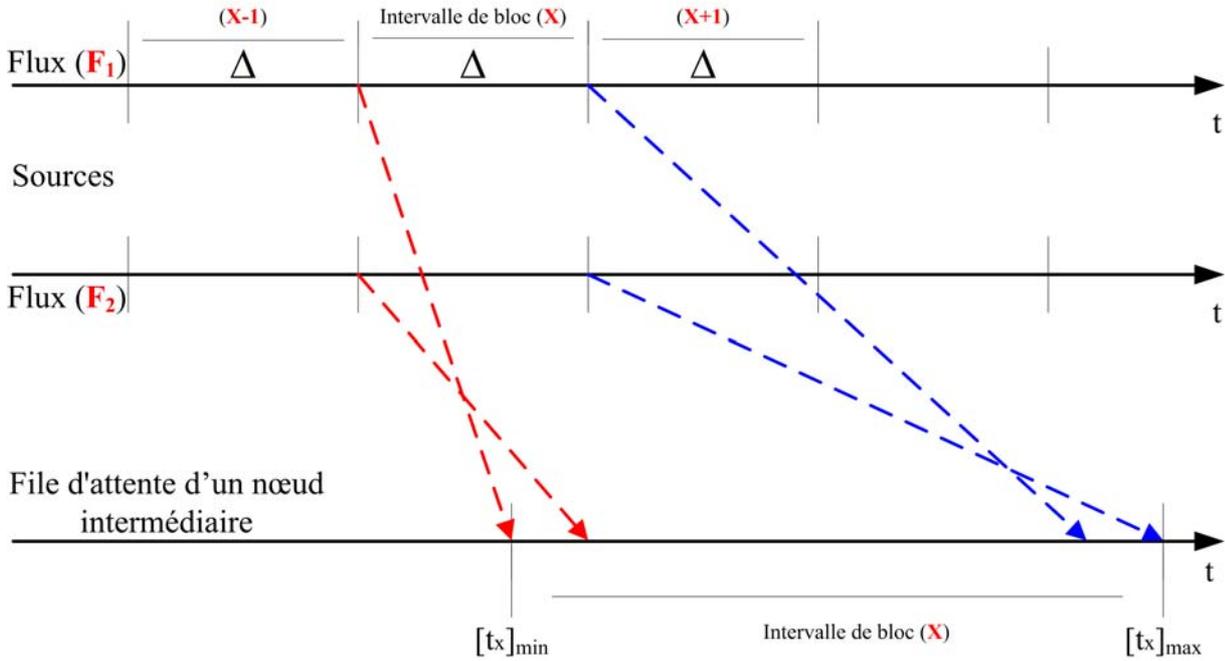


FIGURE 4.9 – Un exemple de temps d'arrivée à un nœud intermédiaire avec deux sources

On peut alors en déduire que le délai maximum subi par un paquet du bloc X dans la file d'attente d'un nœud N_{n+1} est

$$T_B^{n+1} = \frac{([t_X]_{max} - [t_X]_{min})L/\Delta}{C_{n+1}}$$

4.4.4.4 Délai maximum dans un nœud intermédiaire

Le délai maximum d'un paquet X d'un flux F_i , contraint par $\alpha_{\rho_i, \sigma_i}(t)$, dans un nœud intermédiaire N_{n+1} est présenté par l'Équation 4.2 en remplaçant $T_{B_i}^{n+1}$ par T_B^{n+1} :

$$T_i^{\text{codage}} = T_B^{n+1} + T_{lc} + \tau_{n+1} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}}$$

4.4. Stratégie de Transfert Rapide (FFS : Fast Forwarding Strategy)

4.4.5 FFS au niveau d'un réseau

Considérons l'exemple représenté dans la Figure 4.6. Nous pouvons étendre les résultats de la Section 4.3.4 à ce cas. Avec la stratégie de transfert rapide, nous obtenons le délai maximum suivant :

$$T_i^{\text{codage}} = T_{\text{tot}}^i + \frac{\sigma_i}{C_{\text{tot}}^i}$$

où T_{tot}^i et C_{tot}^i sont définies dans les équations 4.4 et 4.5 de la Section 4.3.4.

La seule différence est que la valeur de T_B^k pour un bloc X d'un flux F_i est donnée par

$$T_B^k = \frac{([t_X]_{\text{max}} - [t_X]_{\text{min}})L/\Delta}{C_{k,r}}$$

4.4.6 FFS au niveau de récepteurs

Ces modifications de la stratégie du nœud de codage n'ont aucun effet sur les processus de décodage. Nous allons illustrer ça sur un exemple simple.

Considérons le cas où trois flux seront combinés dans un nœud pour produire un flux sortant. Pour un bloc donné X , les paquets p_X^1, p_X^2 et p_X^3 (respectivement des premier, deuxième et troisième flux) doivent être combinés pour générer le paquet $p_X^{\text{out}} = \alpha_1 p_X^1 + \alpha_2 p_X^2 + \alpha_3 p_X^3$ où α_i sont des coefficients d'un corps fini. Supposons que p_X^1 et p_X^2 arrivent simultanément dans le nœud. Puisque le troisième flux peut être inactif, ces 2 paquets sont autorisés à quitter le nœud sous la forme $p_X^{\text{out},1} = \alpha_1 p_X^1 + \alpha_2 p_X^2$. Si le paquet p_X^3 arrive dans le nœud après que $p_X^{\text{out},1}$ ait été transmis, il est "encodé" et quitte le nœud sous la forme $p_X^{\text{out},2} = \alpha_3 p_X^3$. Dans un nœud suivant appartenant à un des chemins vers un récepteur, $p_X^{\text{out},2}$ peut "rattraper" $p_X^{\text{out},1}$. Comme indiqué sur la Figure 4.8, le paquet $p_X^{\text{out},1}$ est additionné au paquet $p_X^{\text{out},2}$. Le résultat de cette somme est alors $p_X^{\text{out}} = \alpha_1 p_X^1 + \alpha_2 p_X^2 + \alpha_3 p_X^3$ qui est le paquet qui aurait été généré dans le nœud de codage si les 3 paquets avaient été présent ensemble dans la file d'attente à ce moment-là.

Si le paquet $p_X^{\text{out},2}$ ne rattrape pas $p_X^{\text{out},1}$, $p_X^{\text{out},1}$ et $p_X^{\text{out},2}$ arrivent séparément au nœud récepteur, le récepteur doit être capable de vérifier si $p_X^{\text{out},1}$ est le seul paquet qui arrive de ce chemin (c'est à dire, la source qui a généré p_3 n'a pas produit de paquet dans l'intervalle du bloc X) ou si $p_X^{\text{out},2}$ arrivera.

Rappelons tout d'abord que le décodage est simplement une multiplication matrice-vecteur, où la matrice correspond à la matrice inverse du code réseau et le vecteur est l'ensemble de paquets reçus (voir par exemple [2]). Le récepteur vérifie alors la somme de contrôle des paquets obtenus. Puisque les sommes de contrôle ne sont pas linéaires dans le corps fini du code réseau, un paquet, dont la somme de contrôle est correcte, est nécessairement (ou avec une probabilité très élevée) un paquet produit par une source (voir la Section 4.4.2).

Si la somme de contrôle des paquets reconstruits n'est pas correcte, ceci signifie qu'au moins un des paquets d'entrée n'est pas correct et le récepteur doit attendre des paquets additionnels du bloc X . Quand un nouveau paquet arrive, il met à jour les

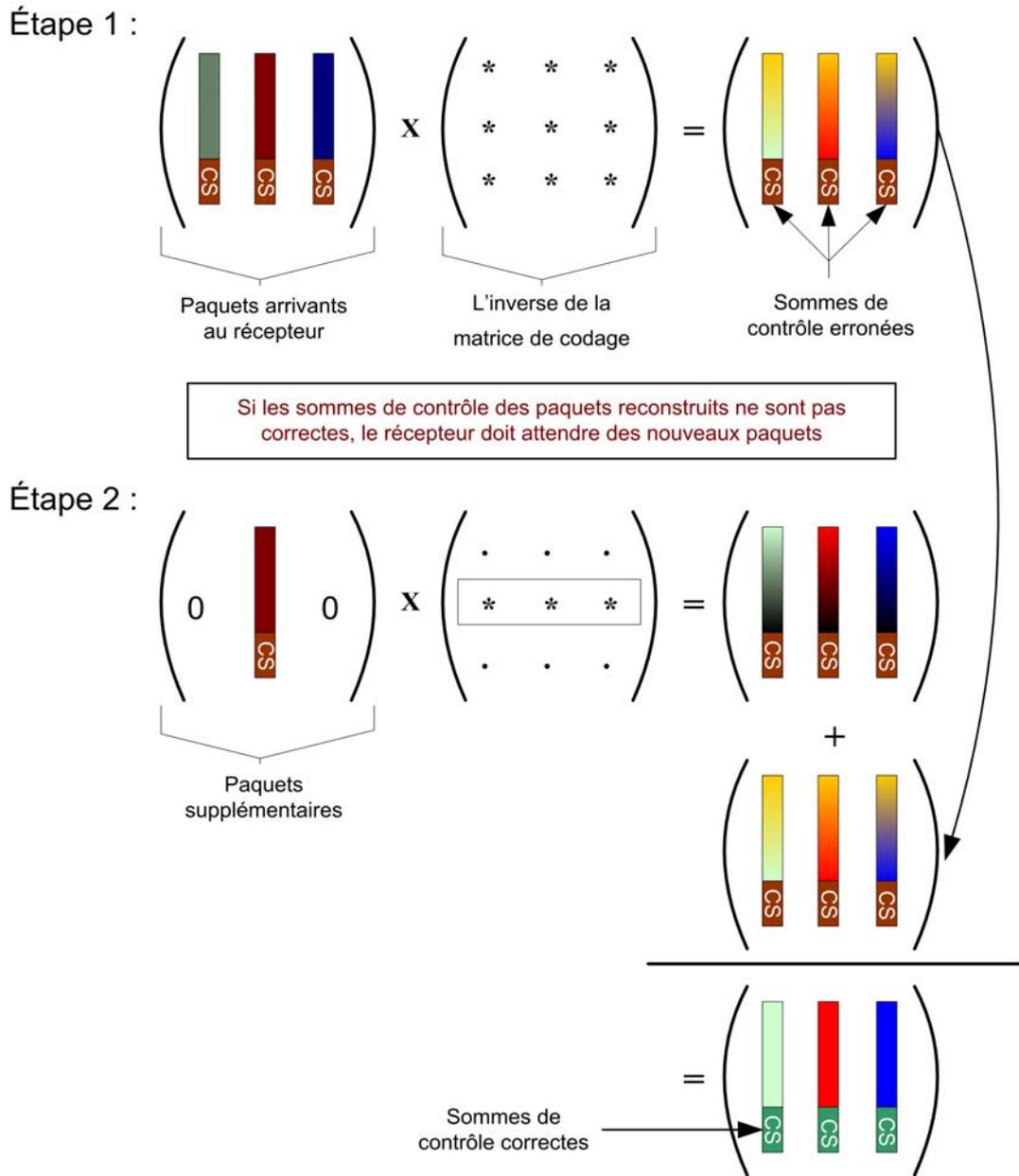


FIGURE 4.10 – FFS au niveau de récepteurs

4.4. Stratégie de Transfert Rapide (FFS : Fast Forwarding Strategy)

paquets obtenus et vérifie encore les sommes de contrôle. Notons que pour faire cette mise à jour, nous n'avons pas besoin d'une multiplication matrice-vecteur complète, mais simplement d'une multiplication scalaire de la ligne correspondante de la matrice et d'une addition de deux vecteurs. La Figure 4.10 illustre le processus de décodage.

4.4.7 Discussion et considérations pratiques

Dans la présentation du système dans la Section 4.4.2, plusieurs hypothèses ont été faites sur les propriétés du réseau et des flux. Dans cette partie, nous allons analyser ces contraintes et proposer quelques méthodes pour les implémenter.

4.4.7.1 Synchronisation des nœuds

La synchronisation des nœuds et notamment des sources est une hypothèse importante du système présenté. En effet, la désynchronisation d'une source pourrait conduire les paquets à arriver à un nœud de codage hors de l'intervalle de temps autorisé. La conséquence pourrait être une perte de paquets et, par conséquent, un mauvais décodage de l'ensemble du bloc.

Deux systèmes sont généralement utilisés pour synchroniser les ordinateurs à distance. Le plus précis est le système GPS dont la précision, liée au temps standard de GPS, est généralement inférieure à $1 \mu\text{s}$ [70].

Pour les réseaux filaires, le protocole du temps du réseau (NTP) "Network Time Protocol" [71] est utilisé par les serveurs de temps d'Internet et leurs clients pour synchroniser les horloges. Il fournit des précisions typiquement dans l'ordre d'une milliseconde sur les réseaux locaux (LANs) et jusqu'à quelques millisecondes sur des WAN grâce au temps universel coordonné (UTC) "Coordinated Universal Time" [72].

4.4.7.2 Délais de transmission

Les délais de transmission sont utilisés dans le système pour évaluer le temps d'arrivée d'un paquet dans un nœud. Un paquet qui arrive après son temps d'arrivée maximum ne peut plus être considéré puisque les autres paquets du même bloc pourraient avoir été déjà transmis par le nœud. Un tel paquet doit être alors supprimé.

Il est possible de rajouter une marge sur la borne maximum, mais celle-ci mène à une augmentation des tailles de backlog dans les nœuds et un délai additionnel sur le délai de bout-en-bout.

4.4.7.3 Mise à jour des paramètres temporels

La précision et la validité des bornes de l'intervalle de temps d'arrivée est essentielle pour le système puisque les tailles de file d'attente dans les nœuds et le délai de bout-en-bout dépendent directement de ces paramètres.

Considérons un paquet du bloc k ($k = 1, 2, 3, \dots$) transmis d'un nœud v_i vers un autre nœud v_j sur le lien $e_{i,j}$. Supposons que l'intervalle de temps d'arrivée utilisé par

Chapitre 4. Nouvelles stratégies de codage réseau

le nœud v_j soit $[t_{k-1} + L/C_{i,j}, t_k + L/C_{i,j} + T_{i,j}]$. Les erreurs sur les bornes de l'intervalle ont deux origines :

- une désynchronisation des horloges des nœuds.
- une erreur sur le délai de transmission entre les deux nœuds $\delta_{i,j}$.

Puisque le délai de transmission peut varier, il est nécessaire de pouvoir mettre à jour ces valeurs. Selon la méthode décrite précédemment, qui permet d'estimer les intervalles de temps d'arrivée de proche en proche, la mise à jour du délai de transmission et les intervalles de temps d'arrivée sont effectués de proche en proche dans le réseau à partir des sources vers les récepteurs.

L'autre intérêt de cette mise à jour est qu'elle intègre indirectement les décalages possible des horloges (voir la Figure 4.11). En effet, pour mesurer le délai de transmission, l'expéditeur v_i insère une estampille temporelle (timestamp) dans le paquet indiquant la date d'émission du paquet (sur son horloge), notée t . Le récepteur (du lien) v_j estime le délai de transmission δ_T en faisant la différence entre la date de réception (sur son horloge, notée par t') et l'estampille temporelle, c.-à-d. $t' - t$. Pour les prochains blocs, il fixe le temps d'arrivée maximum à $t_k + \delta_T$ ($k = 1, 2, 3, \dots$).

Supposons maintenant que l'horloge de l'expéditeur (source) ait un décalage δ_s , c.-à-d. $t_{source} = t_{réf} + \delta_s$, et que l'horloge du récepteur ait un décalage δ_r , c.-à-d. $t_{récepteur} = t_{réf} - \delta_r$.

Pour évaluer le délai de transmission, un paquet envoyé à t vers le récepteur (c.-à-d. à $t + \delta_s$ dans le temps de référence), est reçu à $t + \delta_s + \delta_T$ dans le temps de référence, c.-à-d. à $t' = t + \delta_s + \delta_T - \delta_r$ dans le temps de récepteur. Le délai de transmission estimé par le récepteur est alors égal à $t' - t = t + \delta_s + \delta_T - \delta_r - t = \delta_T + \delta_s - \delta_r$.

L'expéditeur informe le récepteur qu'un paquet du $k^{\text{ème}}$ bloc sera envoyé tout au plus à t_k , au temps d'expéditeur, c.-à-d. en fait à $t_k + \delta_s$ au temps de référence. Le temps d'arrivée maximum est fixé au $(t_k + \delta_r) + \delta_T + \delta_s - \delta_r = t_k + \delta_T + \delta_s$ au temps de référence, c.-à-d. en fait à $t_k + \delta_T + \delta_s - \delta_r$ au temps du récepteur.

Il s'ensuit que les paquets envoyés à $t_k + \delta_s$ arrivent au récepteur après un délai de transmission δ_T à $t_k + \delta_s + \delta_T$. Ainsi, puisque le même argument peut être appliqué à la borne inférieure de l'intervalle, l'intervalle de temps d'arrivée utilisé par le récepteur peut être correctement adapté à l'intervalle de temps utilisé par l'expéditeur (source), même si leurs horloges ne sont pas synchronisées.

Ceci montre que la mise à jour des intervalles de temps d'arrivée permet d'éviter une synchronisation stricte des nœuds de codage et des récepteurs. Si les sources ne sont pas synchronisées, la seule conséquence est une désynchronisation des blocs qui pourrait mener à une augmentation de taille des files d'attente des nœuds de codage situés après les sources.

4.5. Stratégie Classique de Routage/Multiplexage (MS : Multiplexing Strategy)

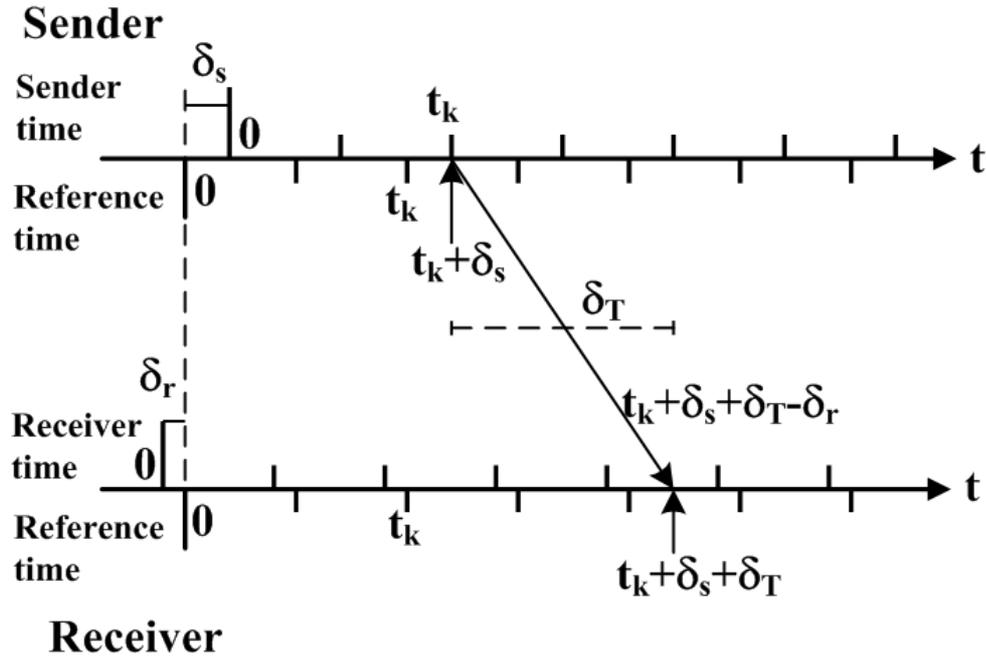


FIGURE 4.11 – Réglage d'intervalle de temps d'arrivée avec des décalages des horloges

4.5 Stratégie Classique de Routage/Multiplexage (MS : Multiplexing Strategy)

4.5.1 Introduction

Dans cette section, nous considérons une approche classique de type *Routage/Multiplexage*. Comme pour les précédentes stratégies, l'analyse sera faite en deux étapes, tout d'abord au niveau d'un nœud, puis au niveau du réseau complet.

4.5.2 MS au niveau d'un nœud

Considérons l'exemple représenté dans la Figure 4.4. Nous supposons que n nœuds sources N_1, \dots, N_n génèrent n flux F_1, \dots, F_n . Ces flux sont respectivement contraints par les courbes d'arrivée affines $\alpha_{\rho_1, \sigma_1}, \dots, \alpha_{\rho_n, \sigma_n}$. Ces flux sont multiplexés par le nœud N_{n+1} et transmis sur le lien e_{n+1} .

Les courbes de service offertes par les liens $e_{1, n+1}, \dots, e_{n, n+1}$ sont respectivement données par $\beta_{C_{1, n+1}, T_{1, n+1}}, \dots, \beta_{C_{n, n+1}, T_{n, n+1}}$. Nous supposons que le débit de n'importe quel flux ne dépasse pas la capacité du lien $\rho_i \leq C_{i, n+1}$.

La courbe d'arrivée d'un flux F_i , selon 3.4-8 et pour $i = 1, \dots, n$, à la sortie du lien $e_{i, n+1}$ est

$$\alpha_{\rho_i, \sigma_i + \rho_i T_{i, n+1}} = \alpha_{\rho_i, \sigma_i} \otimes \beta_{C_{i, n+1}, T_{i, n+1}}$$

Nous supposons qu'un nœud multiplexeur FIFO N_{n+1} offre au flux total (agrégé)

Chapitre 4. Nouvelles stratégies de codage réseau

une courbe de service $\beta_{C_{n+1}, \tau_{n+1}}$ où τ_{n+1} est le temps de service de ce flux dans ce nœud et que la capacité du lien sortant est supérieure ou égale à la somme des débits des flux d'entrée

$$\sum_{i=1}^n \rho_i \leq C_{n+1}$$

En généralisant le théorème [10, Théorème 6.2.3], nous pouvons montrer qu'un nœud multiplexeur FIFO N_{n+1} offre au $i^{\text{ème}}$ flux d'entrée la courbe de service suivante

$$\begin{aligned} \beta_{n+1}^i(t) &= \beta_{C_{n+1}, \tau_{n+1}}(t) - \sum_{k=1, k \neq i}^n \alpha_{\rho_k, \sigma_k + \rho_k T_{k, n+1}}(t) \\ &= \beta_{(C_{n+1} - \sum_{k=1, k \neq i}^n \rho_k), T_i^*} \end{aligned} \quad (4.6)$$

où

$$T_i^* = \tau_{n+1} + \frac{\sum_{k=1, k \neq i}^n (\sigma_k + \rho_k T_{k, n+1})}{C_{n+1}}$$

A partir des mêmes références, nous pouvons obtenir le délai maximum subi par un paquet du flux F_i dans un nœud de multiplexage N_{n+1} :

$$T_i^{\text{multiplexage}} = \tau_{n+1} + \frac{\sum_{k=1, k \neq i}^n (\sigma_k + \rho_k T_{k, n+1})}{C_{n+1}} + \frac{(\sigma_i + \rho_i T_{i, n+1})}{C_{n+1} - \sum_{k=1, k \neq i}^n \rho_k}$$

A partir de ces résultats, nous obtenons la courbe d'arrivée du $i^{\text{ème}}$ flux en sortie de ce multiplexeur. Cette courbe est représentée par la courbe d'arrivée affine $[\alpha_{\rho_i, \sigma_i^*}]$ où la taille maximum du "backlog" est

$$\sigma_i^* = \sigma_i + \rho_i \left(T_{i, n+1} + \tau_{n+1} + \frac{\sum_{k=1, k \neq i}^n (\sigma_k + \rho_k T_{k, n+1})}{C_{n+1}} \right)$$

4.5.3 MS au niveau d'un réseau

Supposons que F_i soit un flux d'entrée contraint par $\alpha_{\rho_i, \sigma_i}(t)$. Ce flux est transmis d'une source S_i à un récepteur R en passant par n nœuds de multiplexage FIFO et n liens.

Nous supposons que la capacité d'un lien sortant d'un nœud est supérieure ou égale à la somme des débits des flux d'entrée. Par exemple, sur la Figure 4.12, pour le nœud N_2 , on a : $\sum_{i=1}^{m_2} \rho_{i,2} \leq C_{2,3}$ où m_2 est le nombre de flux entrants au nœud N_2 .

4.5. Stratégie Classique de Routage/Multiplexage (MS : Multiplexing Strategy)

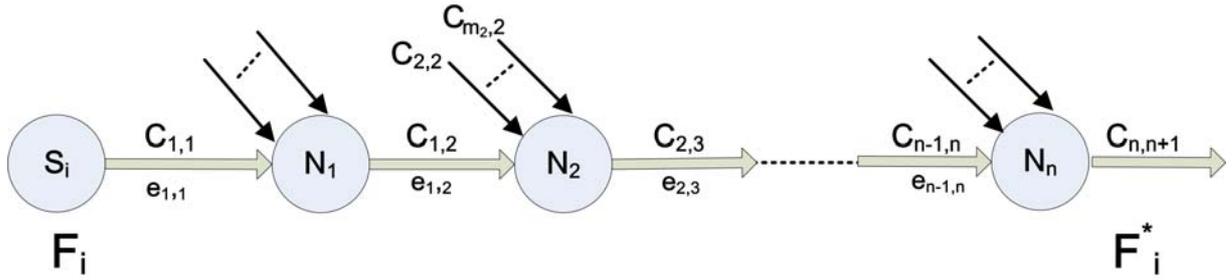


FIGURE 4.12 – Transmission d'un flux à travers n nœuds de multiplexage.

Nous savons que chaque lien $e_{i,k}$ offre une courbe de service

$$\beta_{C_{i,k}, T_{i,k}}(t) = C_{i,k}(t - T_{i,k})$$

et chaque nœud de multiplexage N_k offre au flux total une courbe de service $\beta_{C_{k,r}, \tau_k}(t)$ où $C_{k,r}$ représente la capacité du lien sortant $e_{k,r}$.

En exploitant l'équation 4.6 présentée dans la Section 4.5.2, nous pouvons obtenir la courbe de service offerte au $i^{\text{ème}}$ flux d'entrée par un nœud de multiplexage N_k

$$\beta_k^i(t) = \beta_{C_{k,r} - X_k^i, \tau_k + Y_k^i}(t) \quad (4.7)$$

où

$$X_k^i = \sum_{s=1, s \neq i}^{m_k} \rho_s$$

et

$$Y_k^i = \sum_{s=1, s \neq i}^{m_k} \frac{\sigma_s}{C_{k,r}}$$

Dans ces équations, ρ_s et σ_s représentent respectivement le débit et le "backlog" du flux F_s à l'entrée de nœud N_k .

La courbe de service totale offerte par le chemin au flux F_i est $[\beta_{C_{tot}^i, T_{tot}^i}(t)]$ où

$$T_{tot}^i = T_{1,1} + \sum_{k=1}^{n-1} T_{k,k+1} + \sum_{k=1}^n \{\tau_k + Y_k^i\}$$

et

$$C_{tot}^i = \min(C_{1,1}, C_{1,2} - X_1^i, C_{2,3} - X_2^i, \dots, C_{n,n+1} - X_n^i)$$

Le délai maximum du flux F_i de la source jusqu'à la sortie de ce chemin peut alors être obtenu :

$$T_i^{\text{multiplexage}} = T_{tot}^i + \frac{\sigma_i}{C_{tot}^i}$$

Finalement, les courbes d'arrivée du flux d'entrée F_i à la sortie de ce chemin sont égales à

$$\alpha_{\rho_i, \sigma_i^*}(t) = \alpha_{\rho_i, \sigma_i}(t) \circ \beta_{C_{tot}^i, T_{tot}^i}^i(t)$$

où la taille maximum du "backlog" est

$$\sigma_i^* = \sigma_i + \rho_i * T_{tot}^i$$

4.6 Conclusion générale

Dans ce chapitre, nous avons introduit et analysé trois stratégies de codage réseau.

Nous avons tout d'abord proposé une approche basée sur le codage aléatoire, en termes de dates d'arrivée des paquets, pour appliquer les concepts de codage réseau aux réseaux fournissant des garanties de QoS. Comme cette approche ne fournit pas une fiabilité totale, nous avons proposé, dans le reste de travail, deux autres stratégies totalement fiables basées sur le codage réseau "classique" utilisant le concept de code en bloc.

Nous avons aussi montré comment calculer les délais maximums de bout-en-bout et les tailles des files d'attente pour les différentes stratégies. Nous avons aussi fait ce travail pour une stratégie de routage/multiplexage classique.

Ces différentes études ont été réalisées dans le but d'évaluer de manière globale l'intérêt d'utiliser le codage réseau pour des réseaux à garantie de qualité de service. Les deux chapitres suivants tentent de répondre à cette problématique en comparant les deux approches fiables et l'approche de routage/multiplexage classique sur des réseaux particuliers de manière théorique, puis par simulation.

5

Évaluation du délai maximum

Sommaire

| | | |
|------------|---|------------|
| 5.1 | Introduction | 94 |
| 5.2 | Évaluation de la borne maximale du délai au niveau d'un nœud | 94 |
| 5.2.1 | Rappel des résultats obtenus | 94 |
| 5.2.2 | Stratégie de transfert rapide (FFS) vs stratégie orientée flux (FOS) | 96 |
| 5.2.3 | Stratégie de transfert rapide (FFS) vs Stratégie de routage / multiplexage (MS) | 97 |
| 5.2.4 | Stratégie orientée flux (FOS) vs Stratégie de routage / multiplexage (MS) | 99 |
| 5.2.5 | Application numérique | 102 |
| 5.3 | Évaluation de la borne maximale du délai d'un réseau de bout-en-bout | 103 |
| 5.3.1 | Cas d'Étude -1 : Réseau "Butterfly" | 104 |
| 5.3.2 | Cas d'Étude -2- Réseau avec de multiples flux entrants | 114 |
| 5.3.3 | Cas d'Étude -3- Réseau avec de multiples niveaux du codage / multiplexage | 118 |
| 5.4 | Discussion | 122 |

5.1 Introduction

Nous avons introduit trois stratégies de codage réseau dans le chapitre précédent. Dans la première stratégie, nous avons donné un exemple d'architecture de nœud capable d'implémenter le codage réseau dans le cas des flux de paquets continus, non synchronisés et dépourvus de la notion de blocs (codage réseau aléatoire en termes de dates d'arrivée des paquets).

Même si cette stratégie présente beaucoup d'avantages pour une implémentation pratique (simplicité d'implémentation, souplesse par rapport aux variations des débits des flux, ...), elle ne fournit pas une fiabilité totale en raison du caractère aléatoire du codage. Pour cette raison, nous avons introduit deux autres stratégies fournissant une fiabilité totale, nécessaire dans le contexte des réseaux à garantie de qualité de service.

Dans ce chapitre, nous allons évaluer les résultats de ces deux dernières stratégies en les comparant aux résultats de la stratégie classique du routage / multiplexage. Puis, nous allons effectuer des évaluations et des comparaisons de ces bornes maximales de délais de bout-en-bout dans plusieurs cas d'études avec des réseaux différents. Trois topologies de réseaux différentes seront proposées et étudiées dans le but de comparer les bornes de délais de transmission de bout-en-bout.

5.2 Évaluation de la borne maximale du délai au niveau d'un nœud

Le délai maximum subi par un paquet dans un nœud (de codage ou de routage / multiplexage), par définition, est la somme des délais subis par ce paquet dans les éléments qui composent ce nœud.

Les analyses des délais maximaux au niveau d'un nœud de codage pour les deux stratégies de codages réseau (la stratégie orientée flux (FOS) et la stratégie de transfert rapide (FFS)) et au niveau d'un nœud de routage pour la stratégie classique de routage/multiplexage ont été présentées dans le chapitre précédent où nous avons aussi décrit et expliqué en détails les différentes stratégies.

Dans cette partie, nous comparerons ces délais maximaux au niveau d'un nœud du premier ordre (un nœud qui reçoit ses flux directement des sources). Nous traiterons ensuite le cas d'un réseau complet.

5.2.1 Rappel des résultats obtenus

Considérons un nœud intermédiaire du premier ordre N_{n+1} avec n flux d'entrée et un flux sortant. Les délais maximums subis par un paquet appartenant à un certain bloc X d'un flux F_i , qui est contraint par $\alpha_{\rho_i, \sigma_i}(t)$, dans ce nœud intermédiaire N_{n+1} ont été donnés dans le chapitre précédent pour les différentes stratégies.

5.2. Évaluation de la borne maximale du délai au niveau d'un nœud

5.2.1.1 Stratégie orientée flux (FOS)

En prenant en compte les contraintes et les conditions imposées dans la Section 4.3.2, nous avons obtenu le délai maximum suivant :

$$T_i^{\text{codage}} = T_{B_i}^{n+1} + T_{lc} + \tau_{n+1} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}} \quad (5.1)$$

où

$$T_{B_i}^{n+1} = \Delta + \max_{\substack{j=1,\dots,n \\ j \neq i}} [T_{j,n+1} + \omega_{j,n+1}] - [\omega_{i,n+1}]$$

$$\omega_{j,n+1} = L/C_{j,n+1}$$

pour $j = 1, 2, \dots, n$.

5.2.1.2 Stratégie de transfert rapide (FFS)

Pour cette stratégie, en considérant les contraintes et les conditions introduites dans la Section 4.4.2, le délai maximum dans ce nœud intermédiaire N_{n+1} (représenté sur la Figure 4.8), est égal à :

$$T_i^{\text{codage}} = T_B^{n+1} + T_{lc} + \tau_{n+1} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}} \quad (5.2)$$

où

$$T_B^{n+1} = \frac{([t_X]_{\max} - [t_X]_{\min})L/\Delta}{C_{n+1}}$$

$$[t_X]_{\min} = t + \min\{\omega_{j,n+1}\}$$

$$[t_X]_{\max} = t + \Delta + \max\{T_{j,n+1} + \omega_{j,n+1}\}$$

$$\omega_{j,n+1} = L/C_{j,n+1}$$

pour $j = 1, 2, \dots, n$.

5.2.1.3 Stratégie de routage/multiplexage

Le délai maximum subi par un paquet d'un flux F_i qui arrive sur le lien $e_{i,n+1}$ dans un nœud du routage / multiplexage N_{n+1} est

$$T_i^{\text{multiplexage}} = \tau_{n+1} + \frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j} \quad (5.3)$$

pour $j = 1, 2, \dots, n$.

| Stratégie | Délai Maximum |
|------------|---|
| FOS | $T_{B_i}^{n+1} + T_{lc} + \tau_{n+1} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}}$ |
| FFS | $T_B^{n+1} + T_{lc} + \tau_{n+1} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}}$ |
| MS | $\tau_{n+1} + \frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j}$ |

TABLE 5.1 – Délais maximum au niveau d'un nœud

5.2.1.4 Conclusion

Un résumé de ces trois délais est présenté dans le Tableau 5.1.

Dans la suite, nous allons évaluer les résultats en trois étapes. Nous commencerons par une comparaison des deux stratégies du codage. Nous comparerons ensuite les deux stratégies du codage avec la stratégie de routage/multiplexage classique. Finalement, les délais des trois stratégies seront illustrées par des courbes montrant l'évolution des délais en fonction de débits des flux entrants.

5.2.2 Stratégie de transfert rapide (FFS) vs stratégie orientée flux (FOS)

Nous constatons que les équations (5.1) et (5.2) sont similaires. La seule différence concerne les termes $T_{B_i}^{n+1}$, T_B^{n+1} qui représentent les délais d'attente dans les files d'attente de chaque nœud.

La performance (en terme de délai) de chaque stratégie est liée directement à ces termes.

Nous pouvons récrire le délai maximum d'attente dans la file d'attente du flux F_i dans le cas de la stratégie orientée flux (FOS) comme suit :

$$T_{B_i}^{n+1} = \Delta + \max_{\substack{j=1, \dots, n \\ j \neq i}} [T_{j,n+1} + L/C_{j,n+1}] - [L/C_{i,n+1}]$$

Avec la stratégie de transfert rapide (FFS), le délai maximum d'attente pour un flux F_i dans la seule file d'attente est représenté par

$$T_B^{n+1} = \frac{L}{\Delta \cdot C_{n+1}} \{ \Delta + \max_{j=1, \dots, n} [T_{j,n+1} + L/C_{j,n+1}] - \min_{j=1, \dots, n} [L/C_{j,n+1}] \}$$

| Stratégie | Délai Maximum d'Attente |
|------------|---|
| FOS | $\Delta + \max_{\substack{j=1, \dots, n \\ j \neq i}} [T_{j,n+1} + L/C_{j,n+1}] - [L/C_{i,n+1}]$ |
| FFS | $\frac{L}{\Delta \cdot C_{n+1}} \{ \Delta + \max_{j=1, \dots, n} [T_{j,n+1} + L/C_{j,n+1}] - \min_{j=1, \dots, n} [L/C_{j,n+1}] \}$ |

TABLE 5.2 – Délais maximum d'attente : comparaison entre FOS et FFS

5.2. Évaluation de la borne maximale du délai au niveau d'un nœud

En comparant les deux équations (voir le Tableau 5.2), nous constatons deux choses principales :

- La similarité entre la première équation et le deuxième terme de la deuxième équation. En supposant que tous les liens ont la même capacité et le même temps de service, nous pouvons considérer que la première équation et le deuxième terme de la deuxième équation sont exactement identiques.
- Pour que la stratégie de transfert rapide soit plus avantageuse que la stratégie orientée flux dans un nœud de codage en termes de délai maximum (c.-à-d. elle impose un délai maximum plus petit à la rencontre des paquets), il faut nécessairement que le terme $\frac{L}{\Delta \cdot C_{n+1}}$ soit plus petit ou égal à 1.

Sachant que $\frac{L}{\Delta} = \rho$ représente le taux maximum auquel une source peut générer des paquets et que C_{n+1} est la capacité du lien sortant du nœud et en rappelant les deux contraintes exigées sur les flux et les liens $\rho_j \leq C_{j,n+1}$ et $C_{j,n+1} \leq C_{n+1}$ pour $j = 1, 2, \dots, n$, nous trouvons logiquement que $\frac{L}{\Delta \cdot C_{n+1}} \leq 1$.

Si toutes les capacités des liens en entrée du nœud sont égales, nous remarquons que la stratégie FFS est plus performante que la stratégie FOS. Sa performance dépend des valeurs de Δ et des capacités des liens. En supposant que les capacités sont fixes et que la seule variable est Δ , nous constatons le rôle important de ce paramètre comme cela a été expliqué dans la Section 4.3.5.

5.2.3 Stratégie de transfert rapide (FFS) vs Stratégie de routage / multiplexage (MS)

Comparons maintenant le cas de la stratégie de transfert rapide (FFS) avec la stratégie de routage / multiplexage (MS). Nous constatons en analysant les deux formules des délais maximums (voir le Tableau 5.3) qu'il existe certains points communs entre les deux équations. Nous remarquons, en premier lieu, que les termes finaux des équations,

| Stratégie | Délai Maximum |
|------------|---|
| FFS | $\tau_{n+1} + T_B^{n+1} + T_{lc} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}}$ |
| MS | $\tau_{n+1} + \frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j}$ |

TABLE 5.3 – Délais maximum 1 : comparaison entre **FFS** et **MS**

tions, qui représentent les délais maximums nécessaires pour expédier le backlog maximum dans la file d'attente du flux F_i sur le lien sortant e_{n+1} , sont similaires à la différence près que $(C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j)$ dans l'équation 5.3 est toujours inférieure à C_{n+1} dans l'équation 5.2. En considérant que $\rho_j \leq C_{j,n+1}$, que $C_{j,n+1} \leq C_{n+1}$ et que la capacité du lien sortant est supérieure ou égale à la somme des débits des flux d'entrée

Chapitre 5. Évaluation du délai maximum

($\sum_{j=1}^n \rho_j \leq C_{n+1}$), ce résultat reste valable quelque soit le nombre de flux à l'entrée du nœud et quels que soient les débits de ces flux.

La dernière hypothèse a été choisie pour illustrer le fait que le codage réseau obtient un gain sur les bornes de délais dans des conditions où il ne permet pas d'obtenir un gain en débit moyen.

Par conséquent, le dernier terme de l'équation 5.2 de la stratégie de transfert rapide $\frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}}$ est toujours inférieur au terme $\frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j}$ de la stratégie de routage / multiplexage (MS).

De plus, nous devons noter que le délai maximum de la stratégie de routage / multiplexage augmente en proportion directe avec les augmentations des débits des flux entrants. De son côté, la stratégie de transfert rapide supporte mieux l'augmentation du trafic car le délai maximum augmente proportionnellement au maximum des débits en entrée.

Nous considérons, en deuxième lieu, les autres termes des équations (voir le Tableau 5.3). Nous supposons que le délai ou le temps de service offert au flux total par un nœud multiplexeur FIFO est égal au le temps de service offert au flux codé par un nœud intermédiaire (délai de service pour transmettre des paquets). Donc nous pouvons effacer le terme commun τ_{n+1} des deux équations.

Nous devons alors comparer les termes

$$T_{lc} + \frac{L}{\Delta \cdot C_{n+1}} \left[\Delta + \max_{j=1, \dots, n} \{T_{j,n+1} + L/C_{j,n+1}\} - \min_{j=1, \dots, n} \{L/C_{j,n+1}\} \right]$$

et

$$\frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}}$$

Comme il est difficile de les comparer dans le cas général, nous allons nous focaliser sur le cas particulier où

- Tous les liens ont la même capacité C .
- Les liens ont le même temps de service $T = T_{1,n+1} = T_{2,n+1} = \dots = T_{n,n+1}$.

Nous avons les mêmes capacités des liens, en conséquence :

$$\max_{j=1, \dots, n} \{L/C_{j,n+1}\} = \min_{j=1, \dots, n} \{L/C_{j,n+1}\}$$

Nous obtenons :

$$T_{lc} + \frac{L}{\Delta \cdot C_{n+1}} \left[\Delta + \max_{j=1, \dots, n} \{T_{j,n+1} + L/C_{j,n+1}\} - \min_{j=1, \dots, n} \{L/C_{j,n+1}\} \right] = T_{lc} + \frac{L}{\Delta \cdot C} [\Delta + T]$$

$$\frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}} = \frac{1}{C} \left[\sum_{j=1, j \neq i}^n (\sigma_j) + \sum_{j=1, j \neq i}^n (\rho_j T_{j,n+1}) \right]$$

5.2. Évaluation de la borne maximale du délai au niveau d'un nœud

Sachant que $\rho_j = \rho = \frac{L}{\Delta}$ et que $\{\sigma_j \geq L : \forall j \in \{1, 2, \dots, n\}\}$ (voir la Section 4.3.2), nous trouvons que quelle que soit la valeur de n , les équations suivantes sont toujours vérifiées :

$$L \leq \sum_{j=1, j \neq i}^n (\sigma_j)$$

$$\rho.T \leq n.\rho.T$$

Et donc

$$\frac{L}{\Delta.C} [\Delta + T] \leq \frac{1}{C} \left[\sum_{j=1, j \neq i}^n (\sigma_j) + \sum_{j=1, j \neq i}^n (\rho_j T_{j,n+1}) \right]$$

Sachant que le terme T_{lc} , correspondant à la combinaison des paquets, est considéré comme négligeable devant les autres délais, nous pouvons écrire que

$$T_{lc} + \frac{L}{\Delta.C_{n+1}} \left[\Delta + \max_{j=1, \dots, n} \left\{ T_{j,n+1} + \frac{L}{C_{j,n+1}} \right\} - \min_{j=1, \dots, n} \left\{ \frac{L}{C_{j,n+1}} \right\} \right] \leq \frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}}$$

Finalement, nous concluons que le délai maximum subi par un paquet avec la stratégie de transfert rapide (FFS) est clairement plus petit de celui de la stratégie de routage / multiplexage (MS) et surtout quand la charge du nœud est élevée.

$$\tau_{n+1} + T_B^{n+1} + T_{lc} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}} < \tau_{n+1} + \frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j}$$

5.2.4 Stratégie orientée flux (FOS) vs Stratégie de routage / multiplexage (MS)

Nous allons comparer dans cette partie la stratégie orientée flux (FOS) et la stratégie de routage / multiplexage (MS) en analysant les deux équations correspondant aux délais maximums (5.1) et (5.3) (voir le Tableau 5.4).

Comme dans les comparaisons précédentes, nous supposons que τ_{n+1} a la même valeur dans les deux équations. Nous supposons aussi que $\rho_j \leq C_{j,n+1}$, $C_{j,n+1} \leq C_{n+1}$ et $\sum_{j=1}^n \rho_j \leq C_{n+1}$. Dans le cas de la stratégie orientée flux (FOS), le délai maximum d'attente dans la file d'attente du flux F_i est

$$T_{B_i}^{n+1} = \Delta + \max_{\substack{j=1, \dots, n \\ j \neq i}} [T_{j,n+1} + L/C_{j,n+1}] - [L/C_{i,n+1}]$$

Chapitre 5. Évaluation du délai maximum

| Stratégie | Délai Maximum |
|------------|---|
| FOS | $\tau_{n+1} + T_{B_i}^{n+1} + T_{lc} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}}$ |
| MS | $\tau_{n+1} + \frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j}$ |

TABLE 5.4 – Délais maximum : comparaison entre les stratégies FOS et MS

Donc le délai maximum de la stratégie orientée flux sera plus petit que le délai maximum de la stratégie de routage/multiplexage si

$$T_{lc} + T_{B_i}^{n+1} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}} < \frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j}$$

En considérant les deux conditions présentées dans les parties précédentes (des liens qui ont la même capacité C et le même temps de service $T = T_{1,n+1} = T_{2,n+1} = \dots = T_{n,n+1}$, nous trouvons

$$\Delta + \max_{\substack{j=1, \dots, n \\ j \neq i}} [T_{j,n+1} + L/C_{j,n+1}] - [L/C_{i,n+1}] = \Delta + T$$

$$T_{lc} + T_{B_i}^{n+1} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}} = T_{lc} + \Delta + T + \frac{\sigma_i + \rho_i T}{C}$$

$$\frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j} = \frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T)}{C} + \frac{\sigma_i + \rho_i T}{C - \sum_{j=1, j \neq i}^n \rho_j}$$

Notons que $\{(\rho = \rho_j = \frac{L}{\Delta}) \text{ et } (\sigma = \sigma_j) : \forall j \in \{1, 2, \dots, n\}\}$ où ρ et σ représente le débit et le backlog maximums des sources (voir la Section 4.3.2), donc, nous avons

$$T_{lc} + T_{B_i}^{n+1} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1}} = T_{lc} + \Delta + T + \frac{\sigma + \rho T}{C}$$

$$\frac{\sum_{j=1, j \neq i}^n (\sigma_j + \rho_j T_{j,n+1})}{C_{n+1}} + \frac{\sigma_i + \rho_i T_{i,n+1}}{C_{n+1} - \sum_{j=1, j \neq i}^n \rho_j} = \frac{(n-1)(\sigma + \rho T)}{C} + \frac{\sigma + \rho T}{C - (n-1)\rho}$$

Nous remarquons que $\frac{\sigma + \rho T}{C} \leq \frac{\sigma + \rho T}{C - (n-1)\rho}$ et ce résultat reste toujours valable quelque soit les valeurs de n et de ρ .

5.2. Évaluation de la borne maximale du délai au niveau d'un nœud

Pour des débits des flux entrants très faibles, c'est à dire pour des petites valeurs de ρ et des grandes valeurs de Δ , la stratégie FOS est clairement moins performante que MS. Par contre, lorsque le débit augmente (Δ diminue et ρ augmente), la stratégie MS semble plus impactée que FOS.

Dans la suite, nous allons chercher à déterminer le point de croisement entre les deux stratégies pour une valeur de n donnée. Le point de croisement entre les 2 courbes présentant les deux stratégies doit vérifier la condition suivante :

$$T_{lc} + \Delta + T + \frac{\sigma + \rho T}{C} - \frac{(n-1)(\sigma + \rho T)}{C} - \frac{\sigma + \rho T}{C - (n-1)\rho} = 0$$

$$(C - (n-1)\rho)(T_{lc} + \frac{L}{\rho} + T) + (C - (n-1)\rho)(\frac{\sigma + \rho T}{C})(2-n) - (\sigma + \rho T) = 0$$

Nous obtenons finalement l'équation de degré 3 suivante :

$$[(n-1)(n-2)\frac{T}{C}]\rho^3 +$$

$$[(1-n)T_{lc} + (2-2n)T + (n-1)(n-2)\frac{\sigma}{C}]\rho^2 +$$

$$[(1-n)\sigma + (1-n)L + C.T + C.T_{lc}]\rho + C.L = 0$$

Le délai maximum des combinaisons des paquets T_{lc} est assez petit et négligeable devant les autres délais.

En prenant $L = 1$ (paquets), une solution (dans l'intervalle des valeurs de débit possibles) pour cette équation donnera la valeur de débit de flux à partir de laquelle la stratégie orientée flux (FOS) devient plus avantageuse (voir la Figure 5.1).

Par exemple pour $n = 2$, nous avons

$$(0)\rho^3 + (-2T)\rho^2 + (-\sigma - 1 + C.T)\rho + C = 0$$

Les solutions de cette équation sont :

$$\rho_1, \rho_2 = \frac{-(-\sigma - 1 + C.T) \mp \sqrt{(-\sigma - 1 + C.T)^2 + 8C.T}}{-4T}$$

Pour $n = 3$, nous avons

$$(2\frac{T}{C})\rho^3 + (-4T + 2\frac{\sigma}{C})\rho^2 + (-2\sigma - 2 + C.T)\rho + C = 0$$

Pour $n = 4$, nous avons

$$(6\frac{T}{C})\rho^3 + (-6T + 6\frac{\sigma}{C})\rho^2 + (-3\sigma - 3 + C.T)\rho + C = 0$$

Les racines de ces équations peuvent être obtenues formellement, mais leurs expressions sont trop complexes pour être données ici.

5.2.5 Application numérique

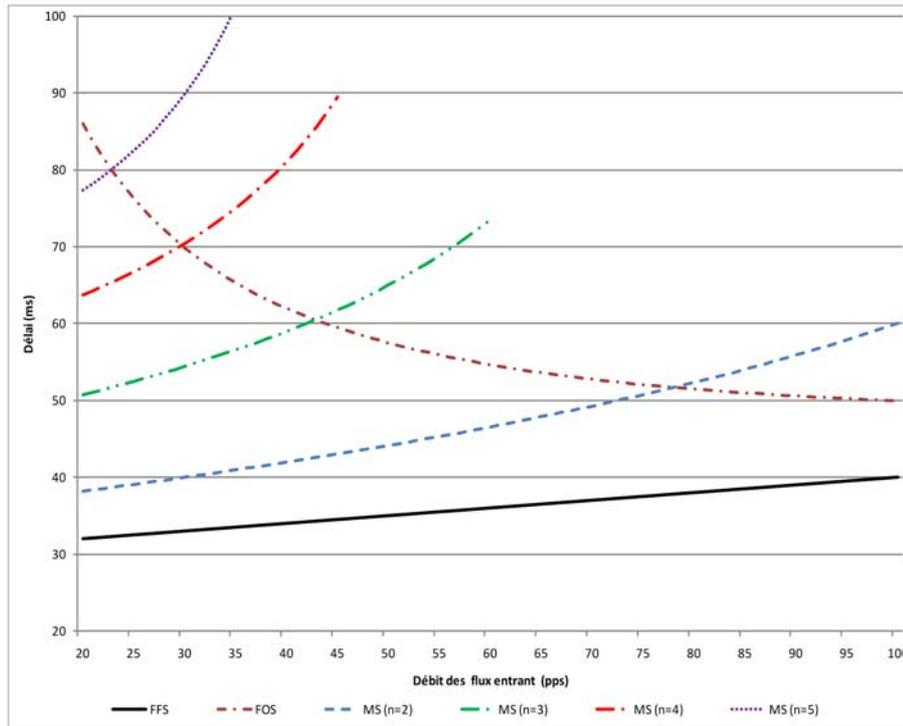


FIGURE 5.1 – Délais maximums dans un nœud

Considérons un nœud intermédiaire du premier ordre avec n liens entrants et un seul lien à la sortie. Dans ce nœud, nous supposons que $\tau_{n+1} = 15$ ms, $T_{lc} = 0$ ms, $\sigma = 2$ paquets.

Nous supposons que tous les flux entrants ont les mêmes propriétés et que $\forall i \leq n : T_{i,n+1} = 10$ ms et $C_{i,n+1} = C_{n+1} = 200$ paquets par seconde (pps). En conséquence, $L/C_{i,n+1} = L/C_{n+1} = 1/200 = 5$ ms.

Supposons que $\rho_1 = \rho_2 = \dots = \rho_n = \rho$. Comme $\sum_{i=1}^n \rho_i \leq C_{n+1}$, la valeur de ρ doit être $\rho \leq \frac{C_{n+1}}{n} = \frac{200}{n}$ pps. Nous supposons donc que ρ varie dans $[20, 100]$ (pps), ce qui correspond à Δ qui varie dans l'intervalle $[10, 50]$ ms).

Pour $n = 2$ liens entrants, $C_1 = C_2 = C_3 = 200$ pps et $\rho_1 = \rho_2 = \rho \leq \frac{200}{2} = 100$ pps.

Si nous avons $n = 3$ liens entrants, $C_1 = C_2 = C_3 = C_4 = 200$ pps et $\rho_1 = \rho_2 = \rho_3 = \rho \leq \frac{200}{3} = 66$ pps.

Appliquons maintenant ces valeurs numériques sur les équations qui caractérisent les délais maximums des trois stratégies (voir les Équations 5.1, 5.2 et 5.3). Les courbes dans la Figure 5.1 représentent les valeurs des délais maximums des trois stratégies lorsque les débits des flux entrants varient dans les limites autorisées selon le nombre de liens entrants n .

Nous remarquons que les délais maximums des stratégies de codage ne varient pas en fonction de n contrairement à la stratégie de routage / multiplexage (MS). La stra-

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

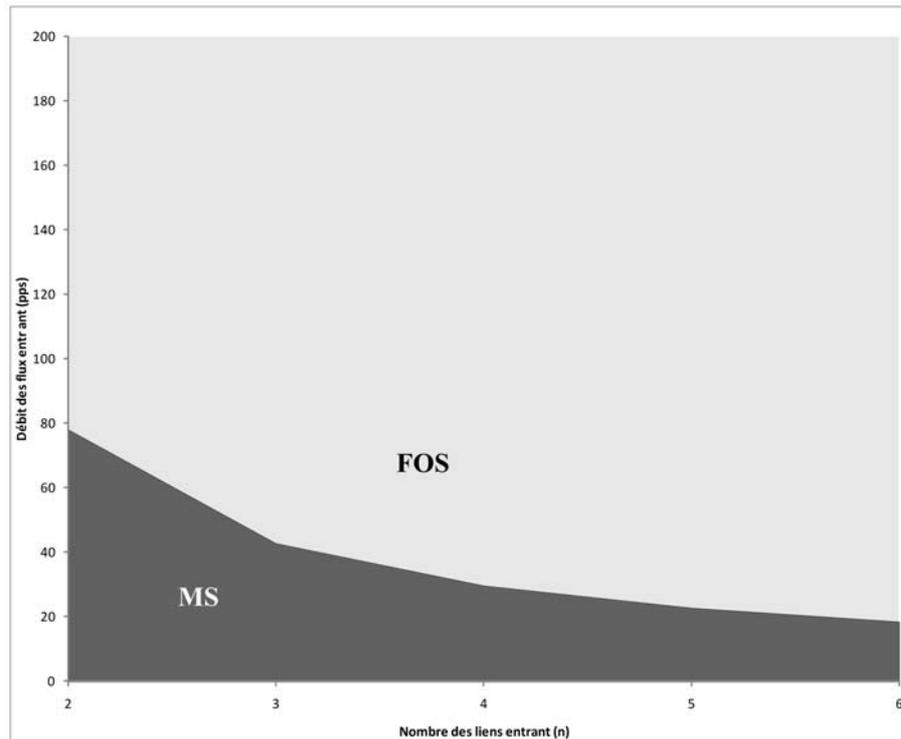


FIGURE 5.2 – Régions favorables pour les stratégies FOS et MS

tégie orientée flux (FOS) dépend directement de la valeur de Δ et par conséquent des débits des flux entrants. Cette stratégie n'est pas toujours avantageuse en comparant à la stratégie de routage / multiplexage (MS). La Figure 5.2 représente les régions où chaque stratégie a l'avantage en fonction des valeurs de n et des débits des flux.

5.3 Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

Dans cette partie, nous allons rappeler sommairement les résultats obtenus dans le chapitre précédent sur les délais maximums des stratégies de codage et de routage / multiplexage au niveau d'un réseau.

Comme il nous semble très difficile de comparer ces trois stratégies sur des réseaux complètement théoriques, nous allons nous focaliser sur trois réseaux particuliers. Pour chacun de ces réseaux, nous calculerons les délais maximums de bout-en-bout pour les trois stratégies. Puis, nous comparerons ces délais pour voir laquelle de ces stratégies est la plus performante et dans quelles conditions.

Le choix de ces cas d'études a été fait en tenant compte aux aspects suivants :

1. Le premier cas d'étude est sur le réseau "butterfly", qui est le premier réseau sur lequel le codage réseau a été appliqué. Ce réseau représente le cas témoin.

Chapitre 5. Évaluation du délai maximum

2. Dans le deuxième cas d'étude, un réseau semblable au réseau "butterfly" sera proposé. L'intérêt de ce réseau vient du fait qu'il contient plus de liens entrants dans le nœud de codage/routage potentiellement engorgé. Ce choix nous permet de comparer ses résultats avec les résultats de premier cas d'étude et d'observer les changements possibles dû à l'augmentation de charge du trafic.
3. Dans le troisième cas d'étude, nous allons considérer un réseau plus complexe avec plusieurs niveaux du codage / multiplexage. Nous avons choisi ce réseau pour observer l'effet apporté sur les délais par ces multiples niveaux de codage/multiplexage.

Rappelons qu'avec les stratégies de codage (FOS et FFS), le délai maximum d'un paquet d'un bloc X du flux F_i entre la source S_i et jusqu'à un certain récepteur en passant par n nœuds intermédiaires égale

$$T_i^{\text{codage}} = T_{tot}^i + \frac{\sigma_i}{C_{tot}^i} \quad (5.4)$$

où

$$T_{tot}^i = T_{1,1} + \sum_{k=1}^{n-1} T_{k,k+1} + \sum_{k=1}^n \{T_{B_i}^k + T_{lc} + \tau_k\} \quad \text{et} \quad C_{tot}^i = \min(C_{1,1}, C_{1,2}, \dots, C_{n-1,n})$$

Comme nous l'avons vu au début de ce chapitre, la seule différence entre les deux stratégies réside dans les valeurs des temps d'attente dans les files d'attente $T_{B_i}^k$ et T_B^k .

Nous rappelons aussi que le délai maximum d'un paquet du flux F_i , calculé entre la source S_i et un certain récepteur, en passant par n nœuds implémentant la stratégie de routage / multiplexage est égal à :

$$T_i^{\text{multiplexage}} = T_{tot}^i + \frac{\sigma_i}{C_{tot}^i} \quad (5.5)$$

où

$$T_{tot}^i = T_{1,1} + \sum_{k=1}^{n-1} T_{k,k+1} + \sum_{k=1}^n \{\tau_k + Y_k^i\}$$

$$C_{tot}^i = \min(C_{1,1}, C_{1,2} - X_1^i, C_{2,3} - X_2^i, \dots, C_{n,n+1} - X_n^i)$$

et

$$X_k^i = \sum_{s=1, s \neq i}^{m_k} \rho_s \quad Y_k^i = \sum_{s=1, s \neq i}^{m_k} \frac{\sigma_s}{C_{k,r}}$$

Un résumé de ces résultats est dans le Tableau 5.5.

5.3.1 Cas d'Étude -1 : Réseau "Butterfly"

Considérons le réseau "Butterfly" représenté dans la Figure 5.3.

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

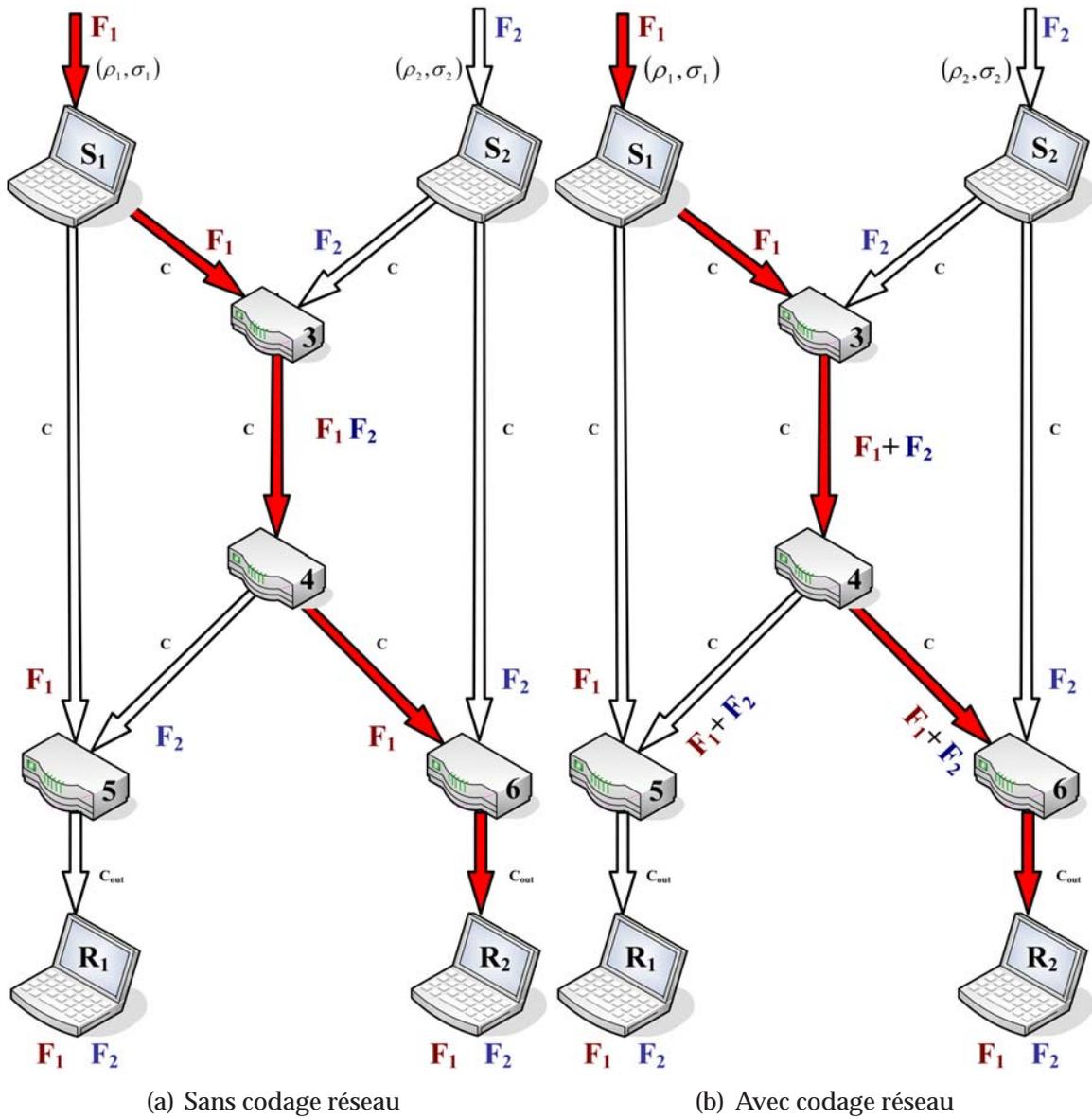


FIGURE 5.3 – Cas d'étude -1- Réseau "Butterfly"

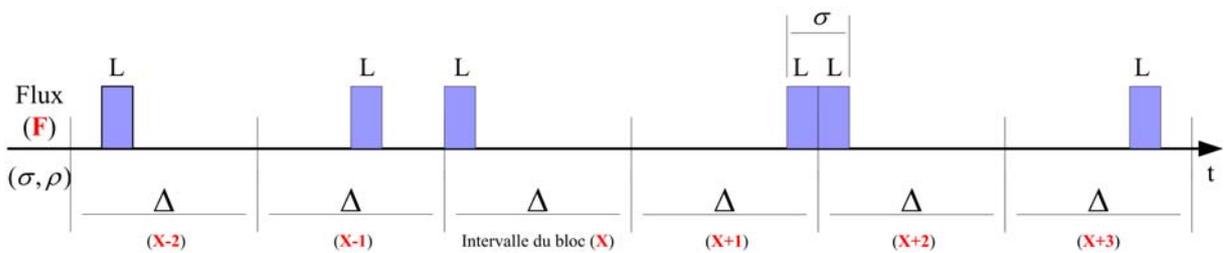


FIGURE 5.4 – Exemple d'un flux étudié

| Stratégie | Délai Maximum | Délai et Capacité Totaux |
|------------|--|---|
| FOS | $T_i^{\text{codage}} = T_{tot}^i + \frac{\sigma_i}{C_{tot}^i}$ | $T_{tot}^i = T_{1,1} + \sum_{k=1}^{n-1} T_{k,k+1} + \sum_{k=1}^n \{T_{B_i}^k + T_{lc} + \tau_k\}$ |
| FFS | | $C_{tot}^i = \min(C_{1,1}, C_{1,2}, \dots, C_{n-1,n})$ |
| MS | $T_i^{\text{multiplexage}} = T_{tot}^i + \frac{\sigma_i}{C_{tot}^i}$ | $T_{tot}^i = T_{1,1} + \sum_{k=1}^{n-1} T_{k,k+1} + \sum_{k=1}^n \{\tau_k + Y_k^i\}$ |
| | | $C_{tot}^i = \min(C_{1,1}, C_{1,2} - X_1^i, \dots, C_{n,n+1} - X_n^i)$ |

TABLE 5.5 – Délais maximum de bout-en-bout

Dans ce réseau, nous supposons que les sources S_1 et S_2 multicast les flux F_1 et F_2 vers deux récepteurs R_1 et R_2 . Supposons que F_1 et F_2 sont contraints par la même courbe d'arrivée affine $\alpha_{\rho,\sigma}(t)$.

Nous allons considérer que tous les liens ont les mêmes capacités C exceptés les liens entrants dans les récepteurs qui ont une capacité C_{out} . Nous avons $\sigma = 2$ paquets et $\rho = \frac{L}{\Delta} \leq C$ (voir la Figure 5.4). Tous les liens ont aussi les mêmes délais de service T . Donc chaque lien $e_{i,j}$ fournit une courbe de service $\beta_{i,j} = \beta_{C,T}(t)$ ou $\beta_{C_{out},T}(t)$. Nous considérons que, pour chaque lien $e_{i,j}$, le délai de transmission maximum d'un paquet de L bits est connu et égal à $L/C + T = \omega + T$.

Nous supposons aussi que chaque nœud du routage / multiplexage N_k offre une courbe de service $\beta_{C,\tau_k}(t)$ ou $\beta_{C_{out},\tau_k}(t)$ où τ_k est le délai de service offert au flux total.

De la même manière, chaque nœud intermédiaire N_k offre une courbe de service

$$\beta_{C,T_B^k+T_{lc}+\tau_k}(t) = C(t - T_B^k - T_{lc} - \tau_k) \text{ ou } \beta_{C_{out},T_B^k+T_{lc}+\tau_k}(t) = C_{out}(t - T_B^k - T_{lc} - \tau_k)$$

où T_B^k ($T_{B_i}^k$ pour FOS) est le temps maximum subi par un paquet dans les files d'attente en attendant les paquets correspondants d'autres flux. Nous avons déjà défini T_{lc} comme le temps maximum nécessaire pour réaliser une combinaison linéaire de paquets. τ_k est le délai de service de flux combiné dans ce nœud.

En considérant les conditions et les hypothèses décrites auparavant, nous remarquons que les délais des pires cas avec les approches du multiplexage et du codage sont obtenus pour les chemins qui traversent le maximum de nœuds, c.-à-d. les chemins qui croisent 3 nœuds sur la Figure 5.3.

Puisque les deux chemins avec 3 nœuds ont la même propriété, nous pouvons choisir un d'entre eux et étudier ses délais des pires cas pour les trois stratégies. Nous allons choisir le chemin suivi par le flux F_1 de S_1 à R_2 qui traverse les nœuds 3, 4 et 6 (chemin marqué en rouge sur la Figure 5.3).

5.3.1.1 Délai maximum de bout-en-bout - stratégie orientée flux (FOS)

Le calcul du délai maximum entre la source S_1 et le récepteur R_2 se fait en trois étapes :

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

1. calcul des courbes de service offertes par chaque nœud et par chaque lien sur ce chemin et pour ce flux, i.e. $\beta_{1,3}$, $\beta_{2,3}$, $\beta_{3,4}$, $\beta_{4,6}$, $\beta_{2,6}$, β_4^1 , β_3^1 , et β_6^1 .
2. calcul des délais d'attente maximums dans les files d'attente $T_{B_1}^3$ et $T_{B_1}^6$ pour le flux F_1 .
3. calcul de la courbe de service totale pour ce chemin $\beta_{C_{tot}, T_{tot}}$.

Nous commençons par calculer les courbes de service déjà connues. Les courbes de service des liens $e_{1,3}$, $e_{3,4}$, $e_{4,6}$ et $e_{2,3}$ sont $\beta_{1,3} = \beta_{2,3} = \beta_{3,4} = \beta_{4,6} = \beta_{2,6} = \beta_{C,T}$. La courbe de service offerte par le nœud N_4 au flux F_1 est $\beta_4^1 = \beta_{C,\tau_4}$.

Les courbes de service offertes par les nœuds N_3 et N_4 au flux F_1 sont respectivement $\beta_3^1 = \beta_{C,\tau_3+T_{lc}+T_{B_1}^3}$ et $\beta_6^1 = \beta_{C_{out},\tau_6+T_{lc}+T_{B_1}^6}$.

Puis, il faut déterminer la valeur de $T_{B_1}^3$. Rappelons que $T_{B_1}^3$ correspond au temps d'attente maximum d'un paquet provenant du flux F_1 dans le nœud N_3 . Ce temps d'attente maximum est atteint lorsque le paquet atteint le nœud avec un délai minimum et lorsqu'il attend la date d'arrivée limite du paquet du même bloc avec lequel il doit être combiné.

Calculons tout d'abord la date d'arrivée minimale d'un paquet du bloc X dans le nœud N_3 .

Pour trouver cette valeur, nous allons envisager le pire des scénarios comme suit : un paquet d'un bloc X , produit par la source S_1 à l'instant $t + (X - 1)\Delta$ (c.-à-d. au début de l'intervalle réservé à son bloc - voir la Figure 5.4), reçoit un service minimum sur le chemin qui le conduit jusqu'à la file d'attente B_1 .

Supposons que ce paquet, qui sera servi seulement par le lien $e_{1,3}$, se trouve dans le plus petit backlog possible (en d'autres termes, nous avons un seul paquet a traité et une courbe d'arrivée $\alpha_{\rho_1,L}$ pour le flux F_1). Notons que le backlog maximum à la source est $\sigma = 2$ paquets (voir la Figure 5.4).

La plus petite courbe de service possible offerte par $e_{1,3}$ sera $\beta_{C_{1,3},0}$. Ainsi, le flux F_1 à l'entrée de cette file d'attente sera représenté par la courbe d'arrivée $\alpha_{\rho_1,L} \otimes \beta_{C_{1,3},0}$. Le délai minimum sera $T_{min} = \frac{\min[\sigma]}{C_{1,3}} = \frac{L}{C_{1,3}} = \omega_{1,3}$.

En conséquence, le paquet qui est produit à l'instant $t + (X - 1)\Delta$ arrivera à la file d'attente B_1 à l'instant $t + (X - 1)\Delta + \omega_{1,3} = t + (X - 1)\Delta + \frac{L}{C_{1,3}}$. Ce paquet doit attendre l'autre paquet du même bloc X qui vient de la source S_2 sur le chemin (le lien) $e_{2,3}$. Le pire scénario ici est que l'autre paquet soit généré à la fin de l'intervalle réservé au bloc X dans la source S_2 , $t + X.\Delta$, et qu'il reçoive un service maximum tout au long de son chemin (lien $e_{2,3}$) jusqu'à sa file d'attente B_2 au nœud N_3 . Le service maximum offert par le lien $e_{2,3}$ est $\beta_{C_{2,3},T_{2,3}}$. Nous supposons aussi que ce paquet fait une partie d'un backlog maximum $\sigma = 2$ paquets.

La courbe d'arrivée du flux F_2 à l'entrée de la file d'attente B_2 sera $\alpha_{\rho_2,2L} \otimes \beta_{C_{2,3},T_{2,3}}$. Le délai maximum pour un paquet de flux F_2 sera $T_{max} = T_{2,3} + \frac{\max[\sigma]}{C_{2,3}} = T_{2,3} + \frac{2L}{C_{2,3}} = T + 2\omega$. Ainsi, ce paquet arrivera à la file d'attente B_2 à l'instant $t + X.\Delta + T_{2,3} + 2.\omega_{2,3} = t + X.\Delta + T_{2,3} + 2.\frac{L}{C_{2,3}}$ au maximum.

Chapitre 5. Évaluation du délai maximum

Donc, nous pouvons calculer $T_{B_1}^3$ comme suit :

$$\begin{aligned} T_{B_1}^3 &= \{t + X.\Delta + T_{2,3} + 2.\frac{L}{C_{2,3}}\} - \{t + (X - 1)\Delta + \frac{L}{C_{1,3}}\} = \Delta + T_{2,3} + (2.\frac{L}{C_{2,3}} - \frac{L}{C_{1,3}}) \\ &= \Delta + T + \frac{L}{C} = \Delta + T + \omega \end{aligned}$$

De façon similaire, $T_{B_2}^3 = \Delta + T + \frac{L}{C}$.

De la même manière, nous allons nous intéresser à $T_{B_1}^6$. Le calcul de ce délai se résume à calculer la différence entre les temps d'arrivée de deux paquets du même bloc, le premier paquet arrivé étant transmis de S_1 et N_6 et le deuxième paquet arrivé étant transmis de S_2 vers N_6 . Entre S_2 et N_6 , le temps d'arrivée maximum d'un paquet du bloc X est obtenu lorsque le paquet est produit par la source S_2 à la fin de son bloc et lorsqu'il subit un délai maximum de transmission. Entre S_1 et N_6 , le délai minimum est atteint lorsque le paquet est produit par la source S_1 au début du bloc X et lorsqu'il subit un délai minimum lors de la transmission. La différence entre les temps d'arrivés de ces paquets aux files d'attente du nœud N_6 donne la valeur de $T_{B_1}^6$.

Plus précisément, nous trouvons tout d'abord le T_{max} entre S_2 et N_6 et le T_{min} entre S_1 et N_6 . Supposons qu'un paquet du bloc X subisse un délai minimum entre la source S_1 et l'entrée de la file d'attente B_1 du nœud N_6 . Cela peut être traduit par une courbe de service représentant le meilleur service possible :

$$\beta_{C_{1,3,0}} \otimes \beta_3^1 \otimes \beta_{C_{3,4,0}} \otimes \beta_4^1 \otimes \beta_{C_{4,6,0}} = \beta_{C,0} \otimes \beta_{C,\tau_3+T_{lc}+T_{B_1}^3} \otimes \beta_{C,0} \otimes \beta_{C,\tau_4} \otimes \beta_{C,0}$$

Pour calculer le délai minimum, nous supposons qu'il n'y a pas de temps d'attente dans la file d'attente ou un temps d'attente minimum, i.e. $T_{B_1}^3 = 0$. Alors, la courbe de service sera

$$\beta_{C,0} \otimes \beta_{C,\tau_3+T_{lc}} \otimes \beta_{C,0} \otimes \beta_{C,\tau_4} \otimes \beta_{C,0} = \beta_{C,\tau_3+\tau_4+T_{lc}}$$

Par suite, le délai minimum T_{min} entre S_1 et N_6 sera $T_{min} = \tau_3 + \tau_4 + T_{lc} + \frac{L}{C} = 2\tau + T_{lc} + \omega$. Le temps d'arrivée minimum de ce paquet dans le nœud 6 sera

$$t + (X - 1)\Delta + T_{min} = t + (X - 1)\Delta + 2\tau + T_{lc} + \omega$$

Pour calculer le temps d'attente maximum de ce paquet dans le nœud 6, nous devons calculer la date d'arrivée maximale du paquet du même bloc transmis de S_2 vers N_6 . La courbe d'arrivée du flux F_2 à l'entrée de la file d'attente B_2 du nœud N_6 est égale à $\alpha_{\rho_2,2L} \otimes \beta_{C_{2,6},T_{2,6}}$. Le délai maximum T_{max} entre S_2 et N_6 est donc $T_{max} = T_{2,6} + 2.\frac{L}{C_{2,6}} = T_{i,j} + 2\omega_{i,j}$. La date d'arrivée maximum de ce paquet sera

$$t + X.\Delta + T_{max} = t + X.\Delta + T + 2\omega$$

Le délai d'attente, noté ϕ , est alors égal à

$$\begin{aligned} \phi &= \{t + X.\Delta + T + 2\omega\} - \{t + (X - 1)\Delta + 2\tau + T_{lc} + \omega\} \\ &= \{T + \omega + \Delta\} - \{2\tau + T_{lc}\} \end{aligned}$$

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

La valeur de ϕ est positive si

$$T + \omega + \Delta > 2\tau + T_{lc}$$

$$\Delta = L/\rho > 2\tau + T_{lc} - T - \omega$$

Pour $L = 1$, ceci est équivalent à :

$$\rho < \frac{1}{2\tau + T_{lc} - T - \omega}$$

Finalement, le temps d'attente maximum dans la file d'attente B_1 de nœud N_6 est :

$$T_{B_1}^6 = \begin{cases} \phi & \text{si } \rho < \frac{1}{2\tau + T_{lc} - T - \omega} \\ 0 & \text{si } \rho \geq \frac{1}{2\tau + T_{lc} - T - \omega} \end{cases}$$

La dernière étape consiste à calculer la courbe de service totale pour le chemin allant de S_1 et jusqu'à la sortie du nœud N_6

$$\begin{aligned} \beta_{C_{tot}, T_{tot}} &= \beta_{1,3} \otimes \beta_3^1 \otimes \beta_{3,4} \otimes \beta_4^1 \otimes \beta_{4,6} \otimes \beta_6^1 \\ &= \beta_{C,T} \otimes \beta_{C,\tau_3+T_{lc}+T_{B_1}^3} \otimes \beta_{C,T} \otimes \beta_{C,\tau_4} \otimes \beta_{C,T} \otimes \beta_{C_{out},\tau_6+T_{lc}+T_{B_1}^6} \\ &= \beta_{\min[C, C_{out}], 3T+3\tau+2T_{lc}+T_{B_1}^3+T_{B_1}^6} \end{aligned}$$

Nous trouvons donc

$$C_{tot} = \min[C, C_{out}]$$

$$T_{tot} = 3T + 3\tau + 2T_{lc} + T_{B_1}^3 + T_{B_1}^6$$

$$T_{tot} = \begin{cases} 5T + \tau + T_{lc} + 2\Delta + 2\omega & \text{si } \rho < \frac{1}{2\tau + T_{lc} - T - \omega} \\ 4T + 3\tau + 2T_{lc} + \Delta + \omega & \text{si } \rho \geq \frac{1}{2\tau + T_{lc} - T - \omega} \end{cases}$$

Le délai maximum entre S_1 et R_2 - selon l'Équation 5.4 - est

$$T_{S_1, R_2}^{\text{codage}} = T_{tot} + \frac{2L}{\min[C, C_{out}]}$$

5.3.1.2 Délai maximum de bout-en-bout - stratégie de transfert rapide (FFS)

Dans cette partie, nous allons tout d'abord calculer la courbe de service totale offerte à un flux F_1 tout au long du chemin entre S_1 et R_2 . Puis, nous calculerons le délai maximum entre cette source et ce récepteur pour un paquet d'un certain bloc X .

Chapitre 5. Évaluation du délai maximum

Le calcul du délai maximum avec cette stratégie se fait de la même manière que la précédente en passant par les mêmes étapes. Les seules différences seront les valeurs des T_B^3 et T_B^6 .

Le calcul de T_B^3 et T_B^6 se fait avec la même méthode que pour la section précédente, mais en appliquant la stratégie de transfert rapide (FFS)(voir la Section 4.4.5).

D'après les résultats obtenus dans la Section 4.4.4 pour trouver les valeurs des délais d'attente dans les files d'attente,

$$T_B^3 = \frac{([t_X]_{max} - [t_X]_{min})\frac{L}{\Delta}}{C}$$

Sachant que la valeur de $[t_X]_{max} - [t_X]_{min}$ est équivalente à la valeur du délai d'attente maximum calculé dans la stratégie précédente pour ce nœud.

$$\begin{aligned} [t_X]_{max} - [t_X]_{min} &= \{t + X.\Delta + T_{2,3} + 2.\frac{L}{C_{2,3}}\} - \{t + (X - 1)\Delta + \frac{L}{C_{1,3}}\} \\ &= \Delta + T + \frac{L}{C} \end{aligned}$$

Donc

$$T_B^3 = \frac{(\Delta + T + \omega)\frac{L}{\Delta}}{C} = \left(\frac{L}{\rho} + T + \omega\right)\frac{\rho}{C}$$

De façon similaire, nous trouvons T_B^6

$$T_B^6 = \frac{([t_X]_{max} - [t_X]_{min})\frac{L}{\Delta}}{C_{out}}$$

où,

- $[t_X]_{min}$: est le délai minimum entre S_2 et N_6 calculé dans la Section 5.3.1.1.

$$[t_X]_{min} = t + (X - 1)\Delta + \omega$$

- $[t_X]_{max}$: représente le délai maximum entre S_1 et N_6 . Pour calculer $[t_X]_{max}$, nous supposons qu'un paquet produit par la source S_1 au début du bloc X , subit un délai minimum entre S_1 et N_3 puis un délai d'attente maximum $T_B^3 = \left(\frac{L}{\rho} + T + \omega\right)\frac{\rho}{C}$ dans le nœud N_3 . Il subit ensuite des délais maximums jusqu'à N_6 . Donc

$$\begin{aligned} [t_X]_{max} &= t + (X - 1)\Delta + \tau_3 + T_{lc} + T_B^3 + T_{3,4} + \tau_4 + T_{4,6} + \omega \\ &= t + (X - 1)\Delta + \tau_3 + T_{lc} + \left(\frac{L}{\rho} + T + \omega\right)\frac{\rho}{C} + T + \tau_4 + T + \omega \end{aligned}$$

Nous obtenons alors

$$[t_X]_{max} - [t_X]_{min} = 2\tau + T_{lc} + 2T + \left(\frac{L}{\rho} + T + \omega\right)\frac{\rho}{C}$$

Nous en déduisons le temps d'attente maximum dans la seule file d'attente B de nœud N_6 :

$$T_B^6 = \left\{2\tau + T_{lc} + 2T + \left(\frac{L}{\rho} + T + \omega\right)\frac{\rho}{C}\right\}\frac{\rho}{C_{out}}$$

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

Enfin, la courbe de service totale entre S_1 et la sortie du nœud N_6 est

$$\begin{aligned}\beta_{C_{tot}, T_{tot}} &= \beta_{1,3} \otimes \beta_3^1 \otimes \beta_{3,4} \otimes \beta_4^1 \otimes \beta_{4,6} \otimes \beta_6^1 \\ &= \beta_{C,T} \otimes \beta_{C,\tau_3+T_{lc}+T_B^3} \otimes \beta_{C,T} \otimes \beta_{C,\tau_4} \otimes \beta_{C,T} \otimes \beta_{C_{out},\tau_6+T_{lc}+T_B^6} \\ &= \beta_{\min[C, C_{out}], 3T+3\tau+2T_{lc}+T_B^3+T_B^6}\end{aligned}$$

Nous trouvons donc

$$C_{tot} = \min[C, C_{out}]$$

$$T_{tot} = 3T + 3\tau + 2T_{lc} + T_B^3 + T_B^6$$

$$\begin{aligned}T_{tot} &= 3T + 3\tau + 2T_{lc} + (\Delta + T + \omega) \frac{\rho}{C} \\ &\quad + \left\{ 2\tau + T_{lc} + 2T + \left(\frac{L}{\rho} + T + \omega \right) \frac{\rho}{C} \right\} \frac{\rho}{C_{out}}\end{aligned}$$

Finalement, le délai maximum entre S_1 et R_2 est

$$T_{S_1, R_2}^{\text{codage}} = T_{tot} + \frac{2L}{\min[C, C_{out}]}$$

5.3.1.3 Délai maximum de bout-en-bout - stratégie de routage / multiplexage (MS)

Nous supposons que la stratégie de routage / multiplexage (MS) est appliquée au réseau présenté sur la Figure 5.3(a). Les nœuds N_3 et N_6 sont des nœuds de multiplexage FIFO. Nous considérons les mêmes contraintes et conditions imposées auparavant. Chaque flux F_i est représenté par sa courbe d'arrivée $\alpha_{\rho_i, \sigma_i}$, le débit d'un flux sur un lien est toujours inférieur ou égal à la capacité de ce lien $\rho_i \leq C$, la somme des débits des flux entrants un nœud est inférieure ou égale à la capacité du lien sortant $\sum_{i=1}^n \rho_i \leq C_{out}$.

La courbe de service totale entre S_1 et la sortie du nœud N_6 est donnée par la relation utilisée précédemment

$$\beta_{C_{tot}, T_{tot}} = \beta_{1,3} \otimes \beta_3^1 \otimes \beta_{3,4} \otimes \beta_4^1 \otimes \beta_{4,6} \otimes \beta_6^1$$

Toutes les courbes de service sont connues mis à part β_3^1 et β_6^1 . Ces courbes de service peuvent être trouvées grâce à l'Équation 4.6. Pour trouver β_3^1 et β_6^1 , nous avons donc

$$\begin{aligned}\beta_3^1(t) &= \beta_{C_{3,4}, \tau_3}(t) - \alpha_{\rho_2, \sigma_2 + \rho_2 T_{2,3}}(t) = \beta_{(C_{3,4} - \rho_2), \tau_3 + \frac{\sigma_2 + \rho_2 T_{2,3}}{C_{3,4}}} \\ &= \beta_{(C - \rho), (\tau + \frac{2L + \rho T}{C})}\end{aligned}$$

$$\begin{aligned}\beta_6^1(t) &= \beta_{C_{out}, \tau_6}(t) - \alpha_{\rho_2, \sigma_2 + \rho_2 T_{2,6}}(t) = \beta_{(C_{out} - \rho_2), \tau_6 + \frac{\sigma_2 + \rho_2 T_{2,6}}{C_{out}}} \\ &= \beta_{(C_{out} - \rho), (\tau + \frac{2L + \rho T}{C_{out}})}\end{aligned}$$

Chapitre 5. Évaluation du délai maximum

La courbe de service totale est alors

$$\begin{aligned}
 \beta_{C_{tot}, T_{tot}} &= \beta_{C_{1,3}, T_{1,3}} \otimes \beta_{(C_{3,4}-\rho), \tau_3 + \frac{\sigma_2 + \rho_2 T_{2,3}}{C_{3,4}}} \otimes \beta_{C_{3,4}, T_{3,4}} \otimes \beta_{C_{4,6}, \tau_4} \otimes \beta_{C_{4,6}, T_{4,6}} \\
 &\quad \otimes \beta_{(C_{out}-\rho), \tau_6 + \frac{\sigma_2 + \rho_2 T_{2,6}}{C_{out}}} \\
 &= \beta_{C, T} \otimes \beta_{(C-\rho), \tau + \frac{2L + \rho T}{C}} \otimes \beta_{C, T} \otimes \beta_{C, \tau} \otimes \beta_{C, T} \\
 &\quad \otimes \beta_{(C_{out}-\rho), \tau + \frac{2L + \rho T}{C_{out}}} \\
 &= \beta_{\min[C, C-\rho, C_{out}-\rho], 3T + 3\tau + \frac{2L + \rho T}{C} + \frac{2L + \rho T}{C_{out}}}
 \end{aligned}$$

Nous trouvons donc

$$C_{tot} = \min[C, C - \rho, C_{out} - \rho] = \min[C - \rho, C_{out} - \rho]$$

$$T_{tot} = 3T + 3\tau + \frac{2L + \rho T}{C} + \frac{2L + \rho T}{C_{out}}$$

$$T_{tot} = 3T + 3\tau + \frac{2L}{C} + \frac{2L}{C_{out}} + \frac{T}{C}\rho + \frac{T}{C_{out}}\rho$$

Le délai maximum entre S_1 et R_2 sera

$$T_{S_1, R_2}^{\text{multiplexage}} = T_{tot} + \frac{2L}{\min[C - \rho, C_{out} - \rho]}$$

Les Tableaux 5.6 et 5.7 résument les résultats obtenus sur ce réseau avec les trois stratégies.

| Stratégie | Capacité Totale |
|------------|--|
| FOS | $C_{tot} = \min[C, C_{out}]$ |
| FFS | $C_{tot} = \min[C, C_{out}]$ |
| MS | $C_{tot} = \min[C - \rho, C_{out} - \rho]$ |

TABLE 5.6 – Capacité totale : réseau "butterfly"

5.3.1.4 Application numérique

Dans cette partie, nous proposons une application numériques des résultats théoriques obtenus précédemment.

Nous supposons que $\tau_i = 15$ ms, $T_{lc} = 0$ ms, $\sigma = 2$ paquets. Nous supposons aussi que tous les flux entrants ont les mêmes propriétés et que $\forall i, j : T = 10$ ms et $C = C_{out} = 200$ paquets par seconde (pps). En conséquence, $L/C = L/C_{out} = 1/200 = 5$ ms.

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

| Stratégie | Délai maximum |
|------------|---|
| FOS | $\begin{cases} 5T + \tau + T_{lc} + 2\Delta + 2\omega + \frac{2L}{\min[C, C_{out}]} & \text{si } \rho < \frac{1}{2\tau + T_{lc} - T - \omega} \\ 4T + 3\tau + 2T_{lc} + \Delta + \omega + \frac{2L}{\min[C, C_{out}]} & \text{si } \rho \geq \frac{1}{2\tau + T_{lc} - T - \omega} \end{cases}$ |
| FFS | $3T + 3\tau + 2T_{lc} + (\Delta + T + \omega) \frac{\rho}{C} + \{2\tau + T_{lc} + 2T + (\frac{L}{\rho} + T + \omega) \frac{\rho}{C}\} \frac{\rho}{C_{out}} + \frac{2L}{\min[C, C_{out}]}$ |
| MS | $3T + 3\tau + \frac{2L}{C} + \frac{2L}{C_{out}} + \frac{T}{C} \rho + \frac{T}{C_{out}} \rho + \frac{2L}{\min[C - \rho, C_{out} - \rho]}$ |

TABLE 5.7 – Délais maximums de bout-en-bout : réseau "butterfly"

Nous supposons aussi que $\rho_1 = \rho_2 = \rho$. Sachant que $\sum_{i=1}^n \rho_i \leq C$, alors la valeur de ρ doit vérifier la relation $\rho \leq \frac{C}{n} = \frac{200}{n}$ pps. Nous supposons donc que ρ varie dans l'intervalle $[20, 100]$ (pps) (ce qui correspond à Δ variant dans l'intervalle $[10, 50]$ ms). Pour $n = 2$ liens entrants, $\rho_1 = \rho_2 = \rho \leq \frac{200}{2} = 100$ (pps).

1. Considérons la **stratégie orientée flux (FOS)**. En appliquant les valeurs numériques aux formules trouvées précédemment, le délai du pire cas est égal à

$$T_{S_1, R_2}^{\text{codage, FOS}} = \begin{cases} 5 \frac{10}{1000} + \frac{15}{1000} + 2 \frac{5}{1000} + \frac{2L}{200} + \frac{2L}{\rho} = \frac{85}{1000} + \frac{2L}{\rho} & \text{si } \rho < 66.66 \\ 4 \frac{10}{1000} + 3 \frac{15}{1000} + \frac{5}{1000} + \frac{2L}{200} + \frac{L}{\rho} = \frac{100}{1000} + \frac{L}{\rho} & \text{si } \rho \geq 66.66 \end{cases}$$

2. En appliquant la **stratégie de transfert rapide (FFS)**, nous constatons que le délai du pire cas est égal à

$$\begin{aligned} T_{S_1, R_2}^{\text{codage, FFS}} &= 3 \frac{10}{1000} + 3 \frac{15}{1000} + (\frac{L}{\rho} + \frac{10}{1000} + \frac{5}{1000}) \frac{\rho}{200} + \{2 \frac{10}{1000} + 2 \frac{15}{1000} + \\ &\quad (\frac{L}{\rho} + \frac{10}{1000} + \frac{5}{1000}) \frac{\rho}{200}\} \frac{\rho}{200} + \frac{2L}{200} \\ &= \frac{90}{1000} + \frac{70}{1000} \frac{\rho}{200} + \frac{15}{1000} \frac{\rho}{200} \frac{\rho}{200} \end{aligned}$$

3. Considérons maintenant la **stratégie de routage / multiplexage (MS)**, le délai maximum du flux F_1 à la sortie du routeur 6 est

$$\begin{aligned} T_{S_1, R_2}^{\text{multiplexage}} &= 3 \frac{10}{1000} + 3 \frac{15}{1000} + 4 \frac{5}{1000} + 2 \frac{10}{1000} \frac{\rho}{200} + \frac{2L}{200 - \rho} \\ &= \frac{95}{1000} + \frac{1}{1000} \frac{\rho}{10} + \frac{2L}{200 - \rho} \end{aligned}$$

Chapitre 5. Évaluation du délai maximum

Pour trouver le débit à partir duquel la stratégie (FOS) est plus performante que (MS), nous devons trouver les valeurs de ρ telles que

$$T_{S_1, R_2}^{\text{codage, FOS}} - T_{S_1, R_2}^{\text{multiplexage}} = 0$$

Ces valeurs sont les racines des polynômes suivants :

$$\text{Pour } \rho < 66.66 : (-1/10000)\rho^3 + (10/1000)\rho^2 + 6\rho - 400 = 0$$

$$\text{Pour } \rho \geq 66.66 : (-1/10000)\rho^3 + (25/1000)\rho^2 + 2\rho - 200 = 0$$

Les solutions sont

$$\rho = -232.34, \rho = 64.2, \rho = 268.13$$

$$\rho = -107.77, \rho = 62.94, \rho = 294.82$$

La seule solution acceptable parmi ces solutions est $\rho = 64.2$ (voir la Figure 5.5).

En reprenant la même approche, les courbes FFS et FOS se croisent lorsque :

$$T_{S_1, R_2}^{\text{codage, FOS}} - T_{S_1, R_2}^{\text{FFS}} = 0$$

c'est à dire sur la seule racine acceptable des équations :

$$\text{Pour } \rho < 66.66 : -(3/8000000)\rho^3 - (7/20000)\rho^2 - (1/200)\rho + 2 = 0$$

$$\text{Pour } \rho \geq 66.66 : -(3/8000000)\rho^3 - (7/20000)\rho^2 - (1/100)\rho + 1 = 0$$

Cette racine est égale à $\rho = 66.66$ (pps)

Dans le cas d'étude suivant, nous allons étudier un réseau un peu plus complexe. Par rapport au réseau étudié dans cette partie, il contient un plus grand nombre de sources et de liens entrants dans le routeur "congestionné". Nous allons trouver les délais maximums des trois stratégies et comparer ces délais avec les délais obtenus avec le réseau "butterfly".

5.3.2 Cas d'Étude -2- Réseau avec de multiples flux entrants

Nous considérons le réseau représenté dans la Figure 5.6. Les sources S_1 , S_2 et S_3 transmettent en multicast les flux F_1 , F_2 et F_3 vers les trois récepteurs R_1, R_2 et R_3 . Les flux F_1 , F_2 et F_3 sont contraints par la même courbe d'arrivée affine $\alpha_{\rho, \sigma}(t)$. Nous allons considérer les mêmes conditions et contraintes que celles imposées dans le premier cas d'étude.

Les délais maximums seront calculés pour les trois stratégies pour un chemin qui traversent le maximum de nœuds. Nous allons choisir le chemin de S_1 à R_3 qui traverse les nœuds 4, 5 et 8 (chemin marqué en rouge sur la Figure 5.6).

Les résultats seront donnés directement dans le Tableau 5.9. Ils sont calculés de façon similaire à celle faite dans le premier cas d'étude.

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

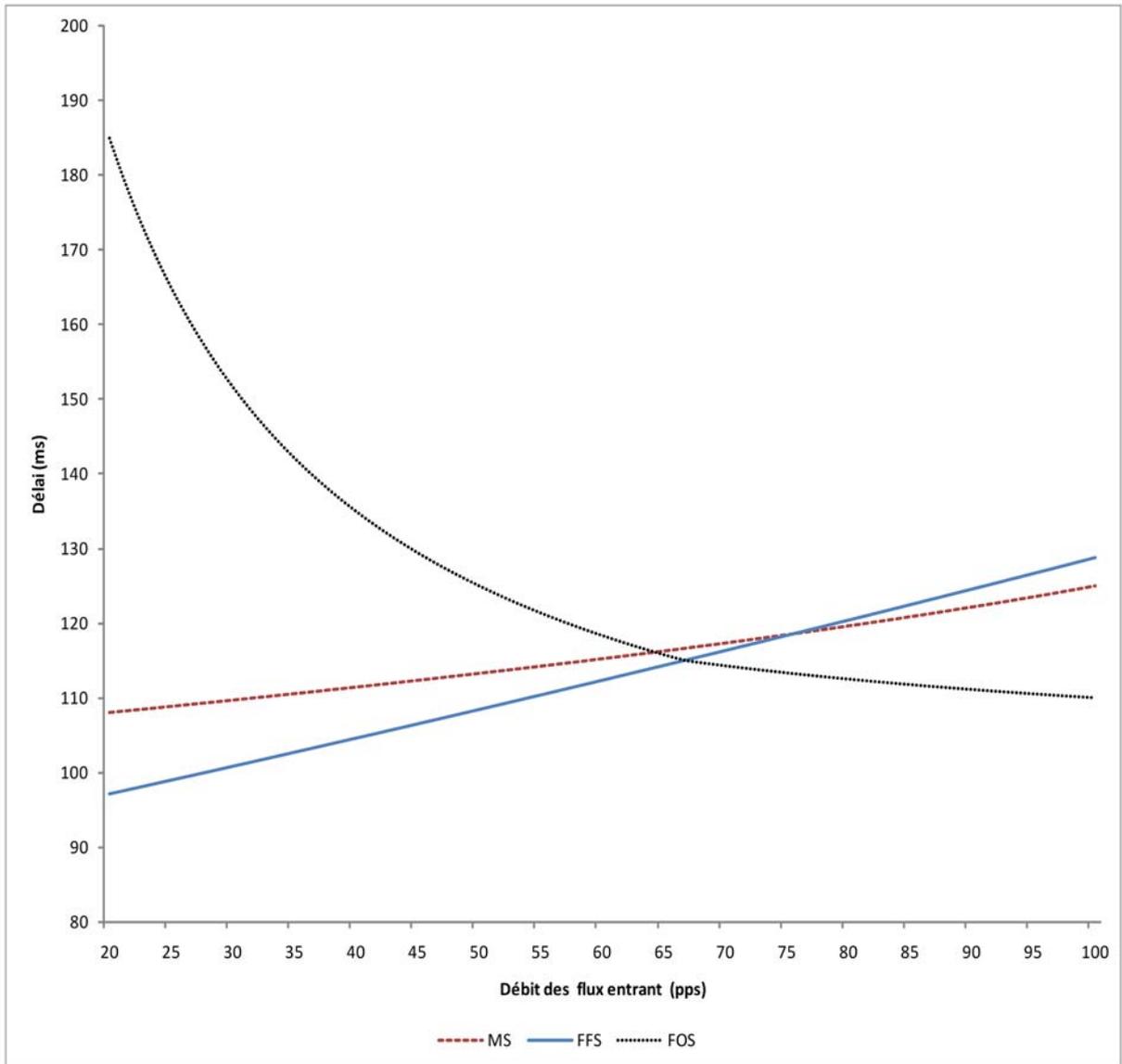


FIGURE 5.5 – Délais maximums : réseau butterfly

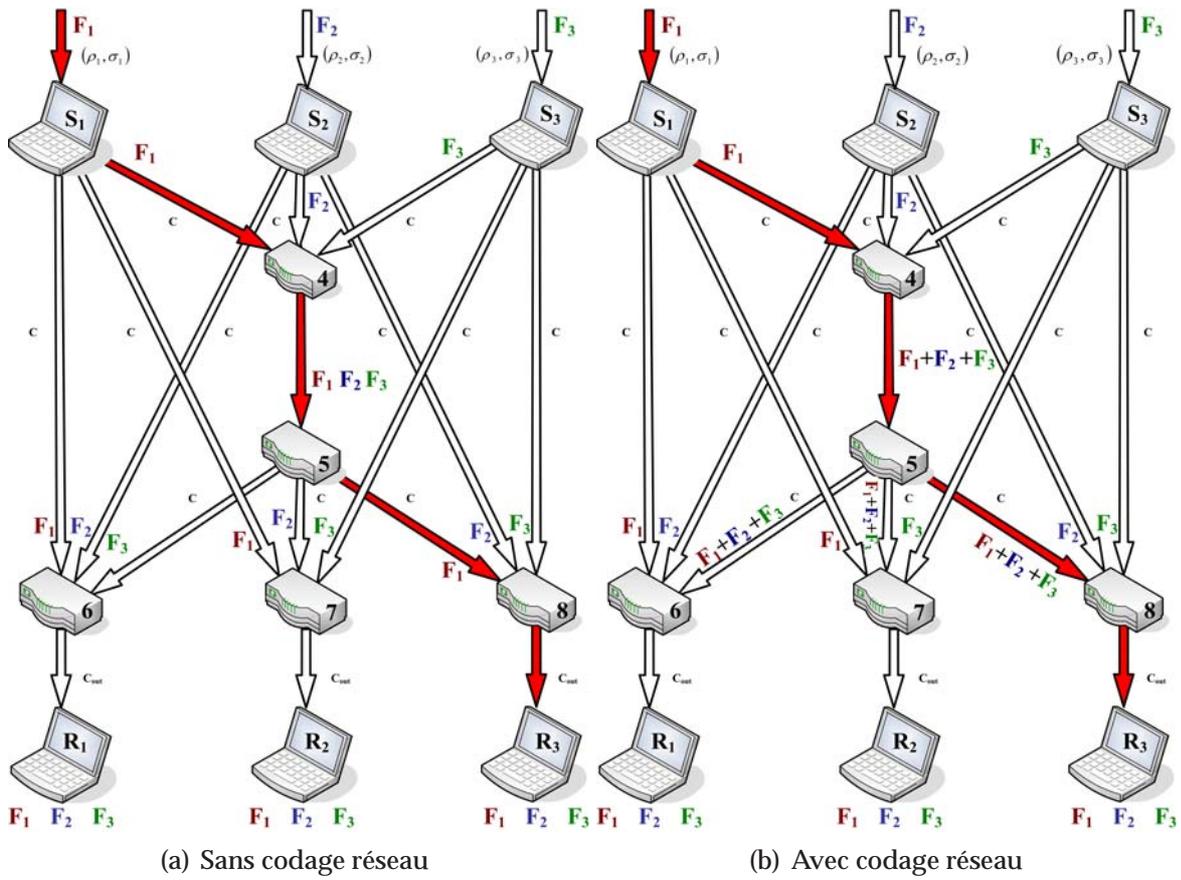


FIGURE 5.6 – Cas d'étude -2- Réseau avec multiples flux entrants

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

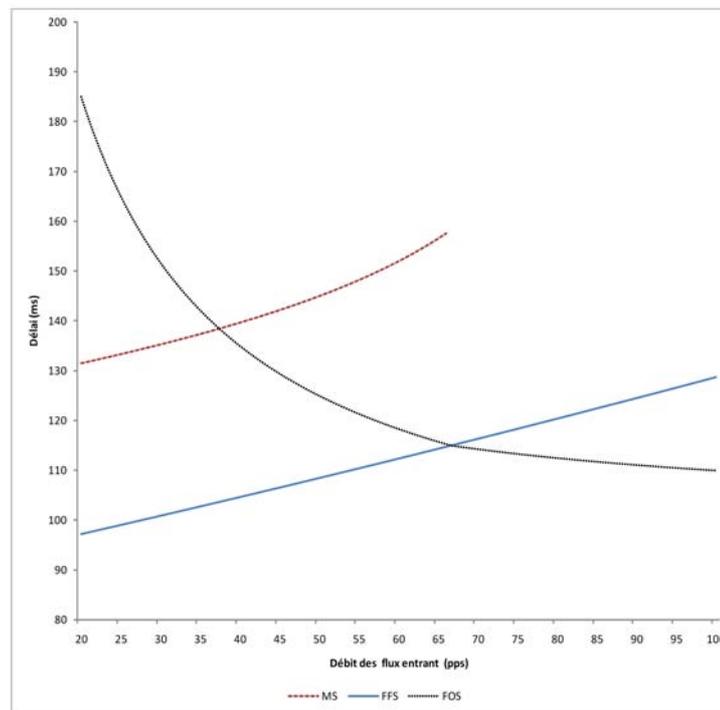


FIGURE 5.7 – Délais maximums : réseau avec multiples flux entrants

| Stratégie | Capacité Totale |
|------------|--|
| FOS | $C_{tot} = \min[C, C_{out}]$ |
| FFS | $C_{tot} = \min[C, C_{out}]$ |
| MS | $C_{tot} = \min[C - 2\rho, C_{out} - 2\rho]$ |

TABLE 5.8 – Capacité totale : réseau avec multiples flux entrants

5.3.2.1 Application numérique

Nous adoptons les mêmes valeurs numériques que celles présentées dans la partie précédente. Nous avons $\rho_1 = \rho_2 = \rho_3 = \rho \leq \frac{200}{3}$ (pps) car le nombre de liens entrants est égal à $n = 3$.

En observant la Figure 5.7 et en la comparant avec la Figure 5.5, nous constatons que les délais de codage n'ont pas changé. Ceci était prévisible car les chemins entre les sources et les récepteurs n'ont pas été modifiés dans les deux réseaux car seul le nombre de flux entrants a changé.

La principale information apportée par ce cas d'étude est que, par rapport au réseau "Butterfly", les stratégies du codage supportent un nombre supérieur de flux sans augmenter les délais contrairement à la stratégie de routage/multiplexage. En effet, même si le nombre de paquets à combiner augmente, les délais n'augmentent qu'à cause de l'augmentation de la complexité du codage. En revanche, plus le nombre de paquets à

| Stratégie | Délai maximum |
|------------|---|
| FOS | $\begin{cases} 5T + \tau + T_{lc} + 2\Delta + 2\omega + \frac{2L}{\min[C, C_{out}]} & \text{si } \rho < \frac{1}{2\tau + T_{lc} - T - \omega} \\ 4T + 3\tau + 2T_{lc} + \Delta + \omega + \frac{2L}{\min[C, C_{out}]} & \text{si } \rho \geq \frac{1}{2\tau + T_{lc} - T - \omega} \end{cases}$ |
| FFS | $\begin{cases} 3T + 3\tau + 2T_{lc} + (\Delta + T + \omega)\frac{\rho}{C} \\ + \{2\tau + T_{lc} + 2T + (\frac{L}{\rho} + T + \omega)\frac{\rho}{C}\}\frac{\rho}{C_{out}} + \frac{2L}{\min[C, C_{out}]} \end{cases}$ |
| MS | $3T + 3\tau + \frac{4L}{C} + \frac{4L}{C_{out}} + \frac{2T}{C}\rho + \frac{2T}{C_{out}}\rho + \frac{2L}{\min[C - 2\rho, C_{out} - 2\rho]}$ |

TABLE 5.9 – Délais maximums de bout-en-bout : réseau avec multiples flux entrants

traiter sur les liens entrants est grand, plus le délai de multiplexage est grand. Ce cas d'étude montre bien l'avantage du codage réseau par rapport à la stratégie classique.

Dans le prochain cas d'étude, nous allons considérer le troisième réseau avec plusieurs niveaux du codage / multiplexage.

5.3.3 Cas d'Étude -3- Réseau avec de multiples niveaux du codage / multiplexage

Considérons le réseau représenté sur la Figure 5.8. Les sources S_1 , S_2 et S_3 transmettent en multicast les flux F_1 , F_2 et F_3 vers deux récepteurs R_1 et R_2 . Considérons les mêmes contraintes sur les flux et les éléments du réseau que celles mentionnées auparavant. Par rapport aux réseaux précédents, ce réseau contient trois niveaux du codage / multiplexage. Chaque niveau du codage / multiplexage a un impact sur les délais maximums de bout-en-bout. Les résultats obtenus illustreront l'avantage des stratégies du codage dans des réseaux qui contiennent plusieurs flux de débit importants et plusieurs niveaux de traitement des paquets.

| Stratégie | Capacité Totale |
|------------|---|
| FOS | $C_{tot} = \min[C, C_{out}]$ |
| FFS | $C_{tot} = \min[C, C_{out}]$ |
| MS | $C_{tot} = \min[C - \rho, C_{out} - 2\rho]$ |

TABLE 5.10 – Capacité totale : réseau avec multiples niveaux du codage / multiplexage

Nous allons directement présenter les résultats obtenus sur les délais maximums

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

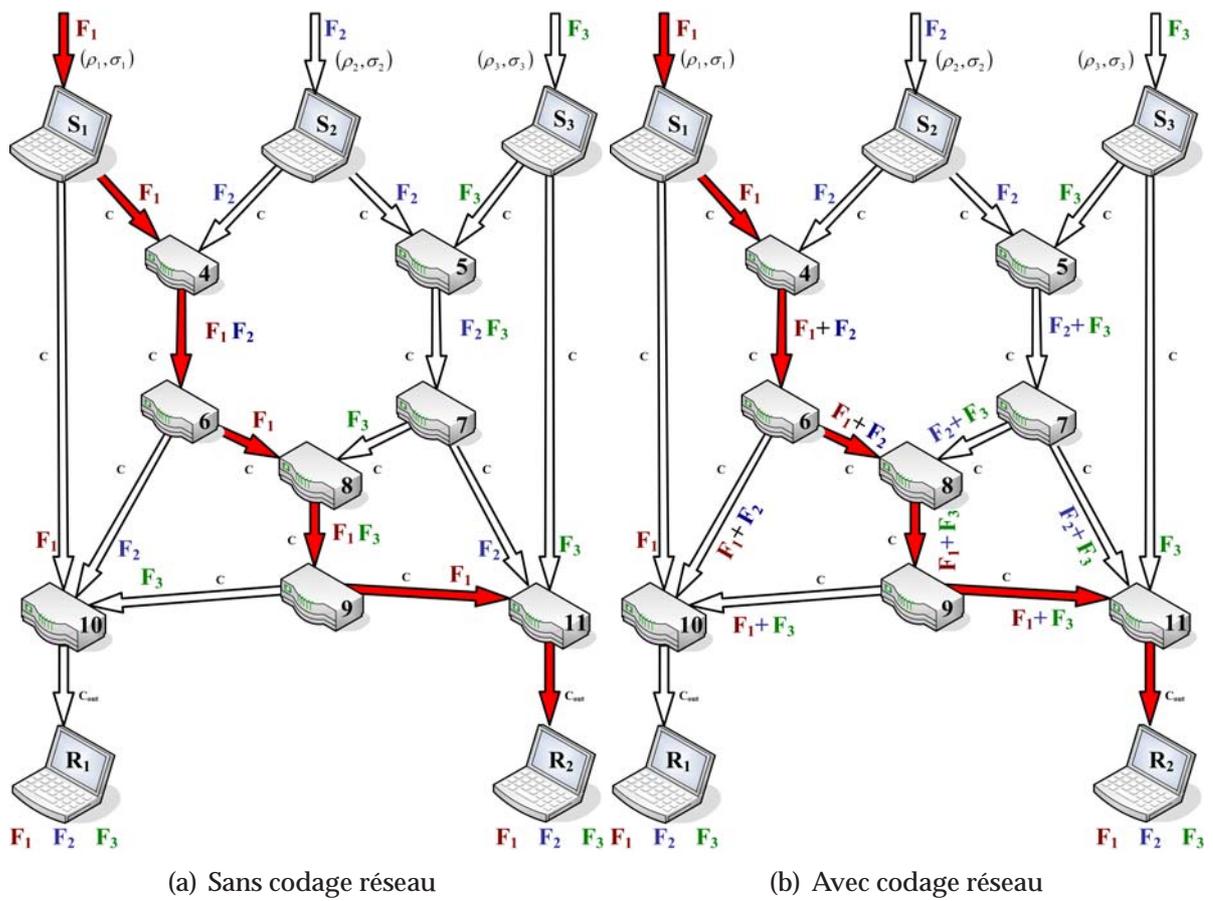


FIGURE 5.8 – Cas d'étude -3- Réseau avec multiples niveaux du codage / multiplexage

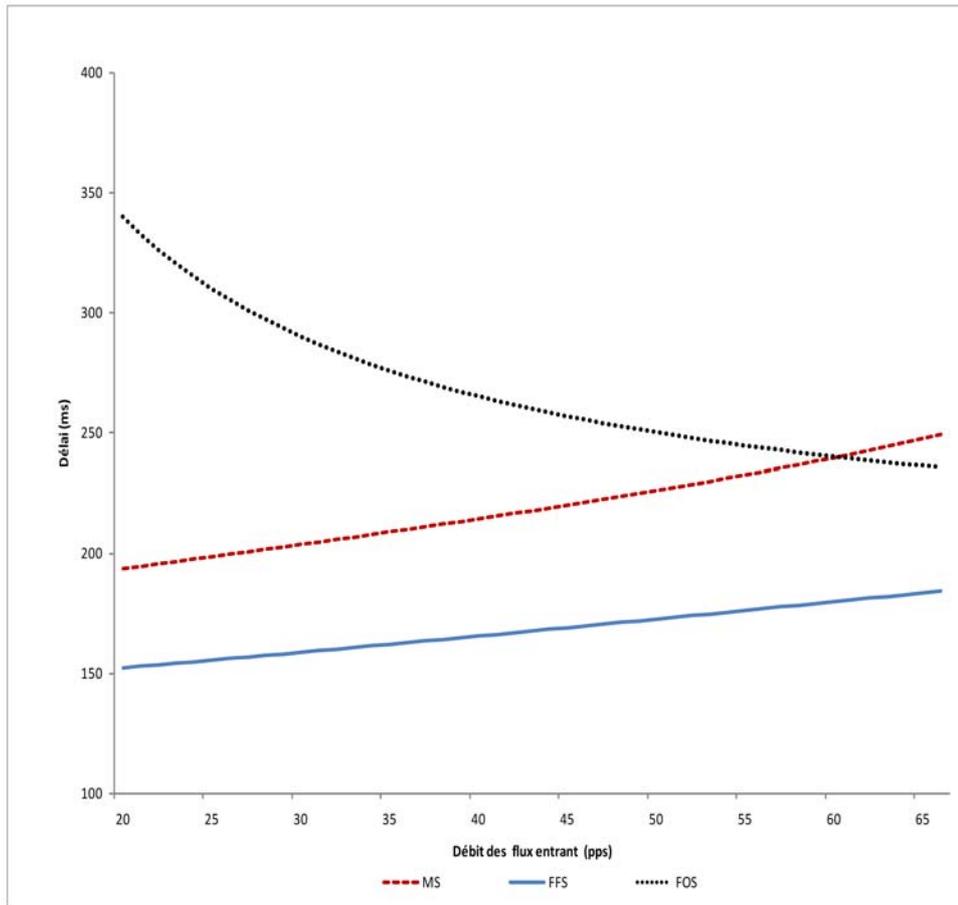


FIGURE 5.9 – Délais maximums : réseau avec multiples niveaux du codage / multiplexage

5.3. Évaluation de la borne maximale du délai d'un réseau de bout-en-bout

de bout-en-bout. Les résultats peuvent être calculés de façon similaire à celle faite dans le premier cas d'étude.

Le Tableau 5.11 résume les délais maximums calculés sur le chemin allant de S_1 à R_2 qui traverse les nœuds 4, 6, 8, 9 et 11 (chemin marqué en rouge sur la Figure 5.8.

| Stratégie | Délai maximum |
|------------|--|
| FOS | $\begin{cases} 12T + 3\tau + 2T_{lc} + 3\Delta + 3\omega + \frac{2L}{\min[C, C_{out}]} & \text{si } \rho < \frac{1}{2\tau + T_{lc} - 3T - \omega} \\ 9T + 5\tau + 3T_{lc} + 2\Delta + 2\omega + \frac{2L}{\min[C, C_{out}]} & \text{si } \rho \geq \frac{1}{2\tau + T_{lc} - 3T - \omega} \end{cases}$ |
| FFS | $\begin{cases} 5T + 5\tau + 3T_{lc} + \frac{\rho}{C}(\Delta + T + \omega)(1 + \frac{\rho}{C} + \frac{\rho}{C} \frac{\rho}{C_{out}}) \\ + 2T \frac{\rho}{C}(1 + \frac{\rho}{C_{out}}) + \frac{\rho}{C_{out}}(2T + 4\tau + 2T_{lc}) + \frac{2L}{\min[C, C_{out}]} \end{cases}$ |
| MS | $\begin{cases} 5T + 5\tau + \frac{4L}{C} + \frac{4L}{C_{out}} + (\frac{\rho}{C} + \frac{\rho}{C_{out}})(4T + 2\tau + \frac{2L}{C} + \frac{T}{C}\rho) \\ + \frac{2L}{\min[C - \rho, C_{out} - 2\rho]} \end{cases}$ |

TABLE 5.11 – Délais maximums de bout-en-bout : réseau avec multiples niveaux du codage / multiplexage

5.3.3.1 Application numérique

Comme dans les cas d'étude précédents, nous évaluons les formules théoriques obtenues pour les différentes stratégies sur un ensemble de valeurs numériques. Ces valeurs sont identiques à celles choisies dans les parties précédentes.

Les débits des flux doivent toujours vérifier la condition suivante : $\rho_1 = \rho_2 = \rho_3 = \rho \leq \frac{200}{3}$ (pps) car le nombre maximum de liens entrants à un nœud est égal à $n = 3$.

En examinant la Figure 5.9 nous constatons que la stratégie (FFS) offre une meilleure performance par rapport aux autres stratégies dans les limites des débits imposés.

Dans cette figure, nous remarquons deux choses :

- nous apercevons clairement le gain de la stratégie (FFS) par rapport à la stratégie (MS). Ce gain augmente avec le débit. Rappelons aussi que nous avons choisi de comparer ces stratégies pour des débits supportables par les 3 stratégies et que les stratégies du codage peuvent supporter des débits proches des valeurs des capacités des liens, ce qui n'est pas le cas de la stratégie MS.
- nous observons des zones non-favorables à l'application de la stratégie FOS. Ces régions correspondent à des valeurs de débits relativement petites. Ce phénomène peut être expliqué par le fait que cette stratégie est contrainte d'imposer

un certain délai d'attente afin de réaliser les combinaisons de paquets. Ces délais d'attente sont liés bien sûr aux débits des flux ("plus les débits sont petits, plus le temps d'attente est grand").

5.4 Discussion

Les comparaisons des délais maximums des trois stratégies, soit au niveau d'un nœud ou au niveau d'un réseau, montrent que la *stratégie de transfert rapide* offre généralement des délais de bout-en-bout meilleurs que ceux offerts par les autres stratégies.

En général, la performance de la *stratégie de transfert rapide* ne dépend pas fortement du nombre de flux ou de débits à l'inverse des deux autres stratégies.

Les conclusions de cette comparaison sont les suivantes :

- Indépendamment des valeurs des débits des flux, la stratégie de transfert rapide est la meilleure dans quasiment tous les cas de figure.
- Plus le réseau est chargé plus l'avantage des stratégies de codage augmente par rapport à la stratégie de routage / multiplexage.
- Lorsque le débit augmente, les délais maximums de la stratégie de transfert rapide augmentent de façon régulière mais moins significative que les délais de la stratégie de routage / multiplexage. Dans le même temps, les délais de la stratégie orientée flux décroissent.
- Le fait que la stratégie orientée flux combine tous les paquets impose un certain délai supplémentaire à ajouter. Ce délai est liée directement à la durée de l'intervalle des blocs et donc inversement aux débits des flux. Par conséquent, il faut éviter d'appliquer cette stratégie quand les paquets sont largement séparés, autrement dit, quand les débits sont petits. En contrepartie, quand le débit augmente, cette stratégie peut se révéler plus performante que la stratégie de transfert rapide.

6

Évaluation et analyse du délai maximum par simulation

Sommaire

| | | |
|------------|---|------------|
| 6.1 | Introduction | 124 |
| 6.2 | Simulation | 124 |
| 6.2.1 | Présentation de simulateur | 124 |
| 6.2.2 | Modélisation des trafics et des délais | 126 |
| 6.3 | Application, analyse et évaluation du délai maximum d'un réseau de bout-en-bout par simulation | 126 |
| 6.4 | Stratégie de transfert rapide (FFS) | 128 |
| 6.5 | Stratégie orientée flux (FOS) | 128 |
| 6.6 | Stratégie de Routage / Multiplexage (MS) | 130 |
| 6.7 | Comparaisons des résultats de simulation obtenus par les trois stra- tégies | 132 |

6.1 Introduction

Une méthode permettant l'évaluation théorique des bornes maximales des délais de bout en bout a été présentée dans les chapitres précédents. Ces bornes représentent les limites maximales calculées pour les pires scénarios possibles.

Dans ce chapitre, nous cherchons à évaluer la différence entre les bornes théoriques et les valeurs de délai maximales rencontrées dans la pratique. Pour cela, nous simulons le fonctionnement des trois réseaux présentés dans le chapitre précédent. Les conditions et les contraintes exigées sur les flux et les éléments des réseaux sont strictement respectées pour que nous puissions avoir des résultats comparables.

Dans la première partie de ce chapitre, nous présenterons le simulateur développé et dans un deuxième temps, nous analyserons les résultats obtenus et nous comparerons les résultats obtenus par les trois stratégies avec les bornes théoriques, puis nous les comparerons entre elles.

6.2 Simulation

6.2.1 Présentation de simulateur

Le simulateur que nous avons implémenté est basé sur le simulateur Java présenté dans [73].

Ce simulateur [73] est une implémentation du codage réseau aléatoire construit à la base pour les réseaux mobile sans fil ou plus généralement pour les réseaux qui fonctionnent dans des environnements extrêmes. Ce simulateur est implémenté en Java et réalise des combinaisons linéaires aléatoires.

Nous avons choisi ce simulateur dans notre travail, car il s'agit d'une structure légère adaptée pour mettre en œuvre et déployer des applications réelles et pour évaluer leur performance dans des conditions réalistes. L'avantage de l'implémentation en Java est sa portabilité et sa capacité à être exécutée indépendamment de la plateforme. Un des ses points forts concerne l'utilisation de la modularité de composition de pile de protocole. Ainsi, dans cette architecture, le codage réseau est présenté comme une couche protocolaire. Il suffit donc de changer le code de cette couche pour modifier le codage. Finalement, il est l'un des rares simulateurs de codage réseau existants disponible pour gratuitement.

D'autres implémentations de codage réseau ont été développées, comme par exemple, le système de Microsoft "avalanche" utilisé pour les réseaux pair-à-pair (P2P) de distribution ou de diffusion de contenu [21]. D'autres exemples sont le système COPE présenté par le MIT pour les réseaux maillés sans fil [26] et le simulateurs C++ présenté dans [53] pour évaluer le performance de codage réseau pour des applications de partage de fichiers dans des réseaux maillés sans fil. D'autres logiciels liés au codage réseau développés pour différents scénarios de communication sont disponibles sur [74].

Dans les parties suivantes, nous allons présenter cette implémentation de codage réseau en décrivant les modifications apportées sur les différentes étapes.

- * Les paquets transmis de chaque nœud du codage sont des combinaisons linéaires des paquets présents dans les files d'attente. Ces combinaisons ne sont pas aléatoires, mais elles sont choisies selon les stratégies de codage FOS et FFS introduites dans le Chapitre 4 en fonction du code réseau défini à priori.
- * Les coefficients utilisés pour former un paquet transmis sont choisis dans un corps fini et ils sont stockés dans l'entête de chaque paquet codé. D'autres identificateurs sont stockés dans l'entête de chaque paquet : une estampille temporelle indiquant la date du bloc du paquet, le numéro du bloc, l'adresse, ... etc. Les identificateurs rendent chaque paquet unique et permettent de le distinguer des autres paquets.

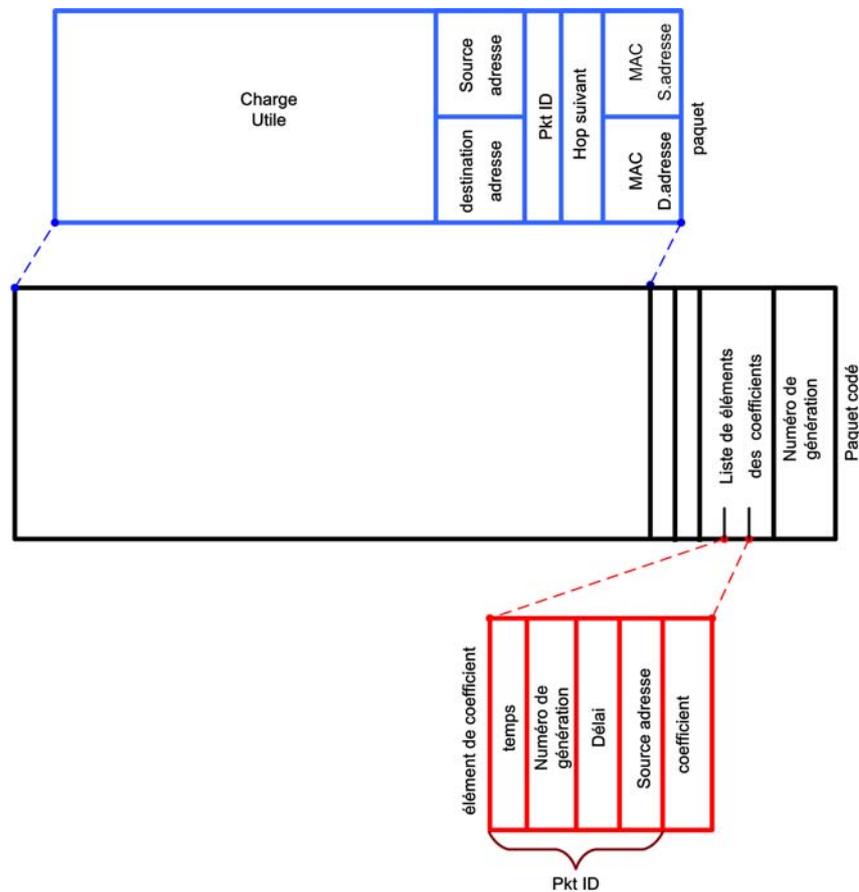


FIGURE 6.1 – Paquet codé

- * Un paquet combiné se compose d'une charge utile codée et d'une liste qui contient les coefficients et les identificateurs des paquets générés par les sources qui sont à l'origine de ce paquet.
- * Quand un nouveau paquet combiné arrive au récepteur, la liste des coefficients qu'il contient est extraite et un pivot de Gauss est appliqué sur la matrice correspondant aux coefficients de codage. Les paquets décodés après chaque étape de réduction sont expédiés à la couche application, si ce récepteur est leur destination finale.

- * Le corps fini utilisé est \mathbb{F}_{2^8} : le principal avantage de ce corps est qu'il constitue un bon compromis car il contient un nombre "suffisant" d'éléments tout en maintenant une vitesse des opérations très rapide. En effet, les éléments de ce corps sont considérés comme des polynômes sur le corps binaire \mathbb{F}_2 et peuvent donc être représentés par des octets. Les additions sont de simples OU exclusifs (eXclusive OR : XOR) entre ces octets. Les multiplications sont réalisées grâce à une représentation des éléments sous la forme de puissance d'un élément primitif. Le passage entre ces différentes représentations se fait en utilisant deux tableaux de 256 octets chacun.

6.2.2 Modélisation des trafics et des délais

Les trafics utilisés dans cette implémentation, sont générés par des sources représentées par des générateurs aléatoires de paquets.

Rappelons que chaque bloc est associé à un intervalle unique $[t, t + \Delta[$. Pour chaque bloc, la source choisit l'instant où un paquet sera produit suivant un tirage aléatoire uniforme dans l'intervalle associé au bloc. Nous rappelons que la source n'est pas autorisée à générer plus d'un paquet dans chaque intervalle (voir la Figure 5.4). Dans cette simulation, pour être en conformité avec l'approche théorique, nous générons exactement un paquet par bloc et par source (alors que les stratégies introduites sont capable de traiter le cas où aucun paquet n'est généré pendant un intervalle du bloc donné).

Les délais subis sur les liens et les délais de service des flux dans les nœuds sont aussi générés suivant une loi uniforme.

6.3 Application, analyse et évaluation du délai maximum d'un réseau de bout-en-bout par simulation

Les résultats obtenus sur réseaux simulés sont représentés sur les Figures 5.3, 5.6 et 5.8.

Chaque source transmet en multicast un fichier de 1000 paquets vers tous les récepteurs. Les sources partagent la même horloge, mais ils ne produisent pas leurs paquets simultanément.

La durée d'un bloc, Δ , varie de 10 ms à 50 ms. Ainsi, les débits des flux varient de 20 à 100 pps (paquets par seconde).

Tous les liens dans les réseaux ont la même capacité C qui est égale à 200 pps et le délai subi par un paquet sur un lien $e_{i,j}$ est compris entre L/C et $L/C + T_{i,j}$. La valeur de $T_{i,j}$, correspondant au délai de transmission d'un paquet sur un lien, est tiré aléatoirement suivant une loi uniforme dans l'intervalle $[0, 10]$ ms.

Tous les nœuds dans les réseaux ont le même délai de service τ qui suit une loi uniforme dans l'intervalle $[0, 15]$. Ce délai est donc en moyenne égal à 7.5 ms.

T_{lc} qui désigne le temps maximal nécessaire pour parvenir à une combinaison linéaire de paquets est considéré comme très petit et négligeable par rapport aux autres

6.3. Application, analyse et évaluation du délai maximum d'un réseau de bout-en-bout par simulation

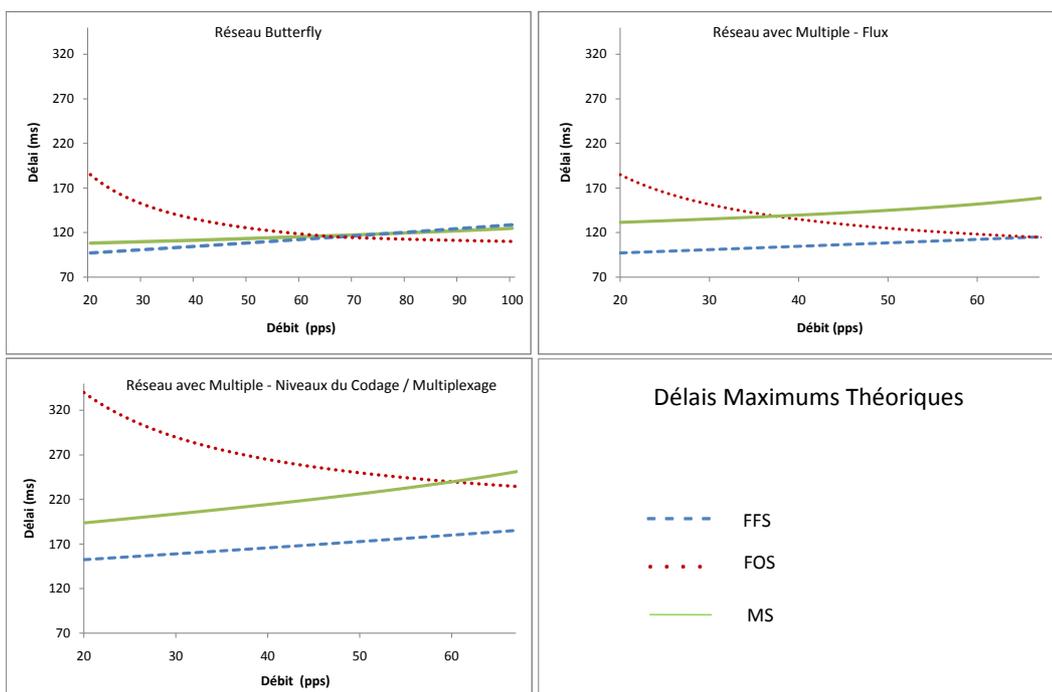


FIGURE 6.2 – Délais maximums théoriques

délais.

Dans le simulateur, les stratégies FOS, FFS et MS ont été implémentées telles que décrites dans le chapitre 4.

La Figure 6.2 rappelle les résultats obtenus dans le Chapitre 5 sur les délais maximums théoriques de bout-en-bout pour les trois réseaux.

Notre but dans ce chapitre est d'estimer et de comparer les délais de bout-en-bout obtenus par simulation avec les résultats théoriques.

Nous allons calculer les délais maximums pour les cas des codages et de multiplexage sur le chemin qui traverse un maximum de nœuds comme nous avons déjà expliqué dans le Chapitre 5.

Pour chaque valeur de débit, qui varie entre 20 et 100 pps, un fichier de 1000 paquets est envoyé de chaque source vers tous les récepteurs. À la fin du chemin maximum choisi, nous calculons les délais maximums subis par les paquets sur ce chemin. L'expérience est répétée 10 fois pour chaque valeur de débit. Pour chaque stratégie et chaque débit, nous indiquons sur la figure les valeurs :

- du maximum des maximums
- de la moyenne des maximums
- du minimum des maximums

Les Figures 6.3, 6.4 et 6.5 montrent les distributions des délais maximums pratiques autour des moyennes pour chaque valeur de débits des flux et les délais maximums théorique pour chaque réseau et chaque stratégie.

6.4 Stratégie de transfert rapide (FFS)

En observant la Figure 6.3 qui présente les distributions des délais maximums pratiques de la stratégie transfert rapide (FFS), nous constatons que

- elles ne dépassent jamais les bornes des délais maximums théoriques.
- les délais sont concentrés autour des moyennes.
- les courbes des délais maximums pratiques et théoriques sont croissantes mais l'écart entre les deux courbes augmente avec le débit.
- plus le réseau est complexe, plus l'écart entre les résultats pratiques et théoriques est grand.

6.5 Stratégie orientée flux (FOS)

Nous observons globalement les même comportement que pour la stratégie précédente. Toutefois :

- l'écart entre les deux courbes diminue lorsque le débit augmente.
- en observant les courbes de la stratégie orientée flux (FOS), nous constatons que les distributions des délais maximums autour de la moyenne sont légèrement plus étalées que les autres stratégies et plus variantes d'un réseau à l'autre.

6.5. Stratégie orientée flux (FOS)

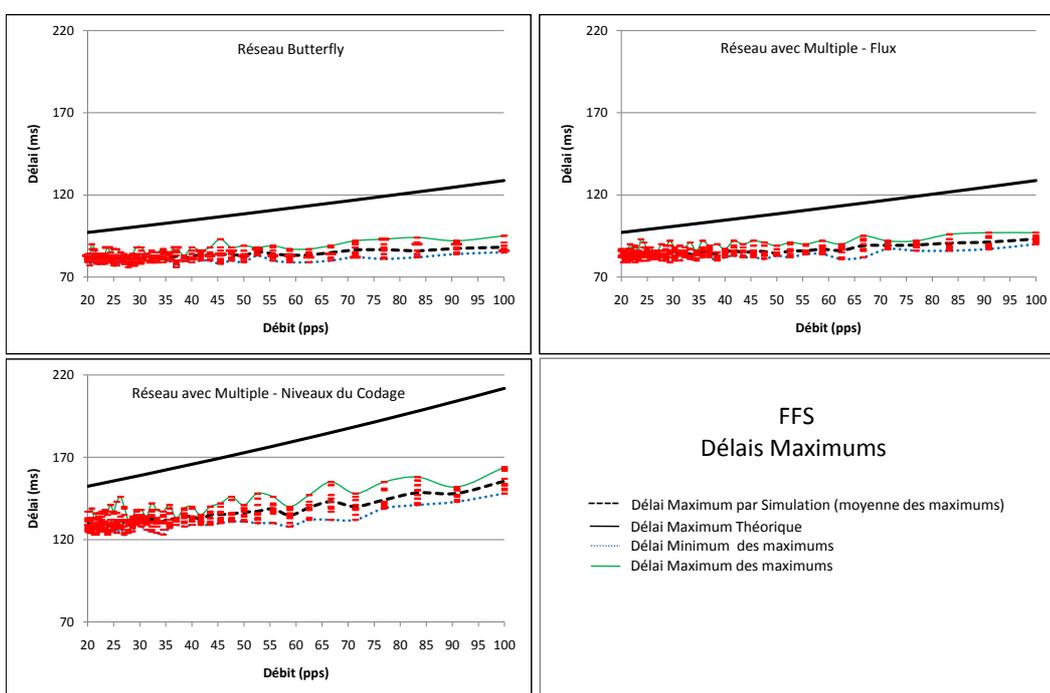


FIGURE 6.3 – Stratégie FFS : Délais Maximums

Chapitre 6. Évaluation et analyse du délai maximum par simulation

Ce dernier point peut s'expliquer par le fonctionnement de la stratégie FOS elle-même. En fait, la condition imposée dans FOS, est "un paquet d'un bloc doit attendre dans la file d'attente l'arrivée de tous les paquets de ce bloc ou la fin de la date-limite du temps d'arrivée du bloc avant qu'il puisse sortir".

En conséquence, un paquet ne peut pas quitter sa file d'attente, même si le lien sortant n'est pas occupé, et ce tant que la condition précédente n'a pas vérifiée. Il faut noter aussi que plus le nombre de flux est grand plus la probabilité que le temps d'attente dans la file d'attente s'approche de la date-limite du temps d'arrivée du bloc, est grande.

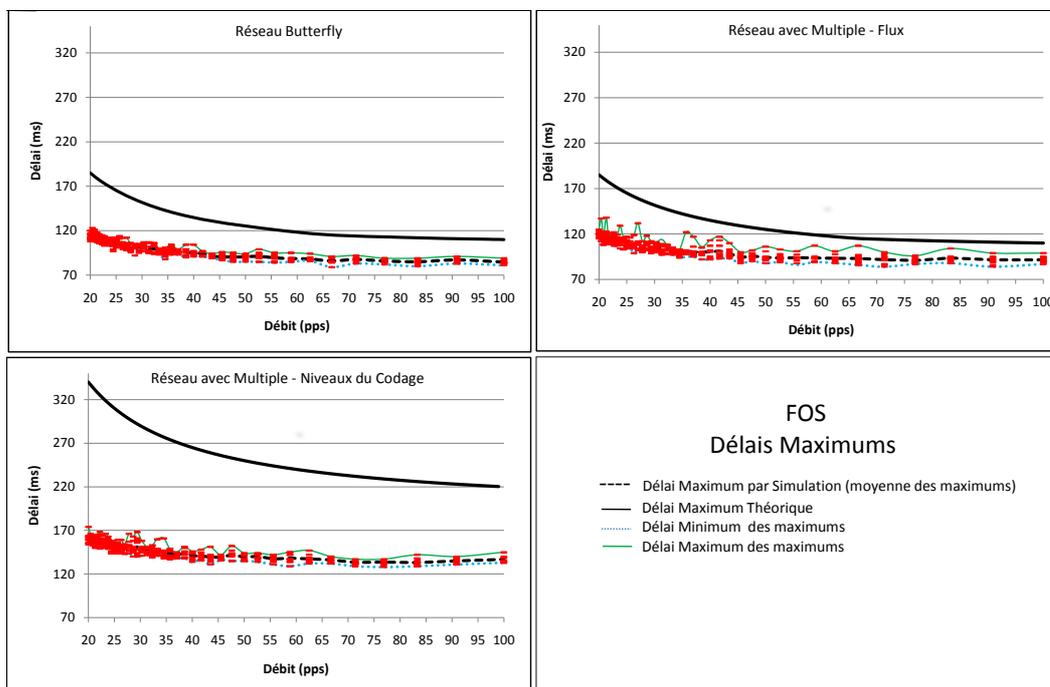


FIGURE 6.4 – Stratégie FOS : Délais Maximums

6.6 Stratégie de Routage / Multiplexage (MS)

Les mêmes types de remarques notées auparavant sont applicables sur les courbes de la stratégie de Routage / Multiplexage (MS). En revanche, nous pouvons observer que les courbes de cette stratégie sont légèrement croissantes jusqu'à un seuil inférieur de débit à partir duquel la croissance est plus forte. Ce seuil correspond à une situation où la congestion commence à apparaître dans les nœuds.

6.6. Stratégie de Routage / Multiplexage (MS)

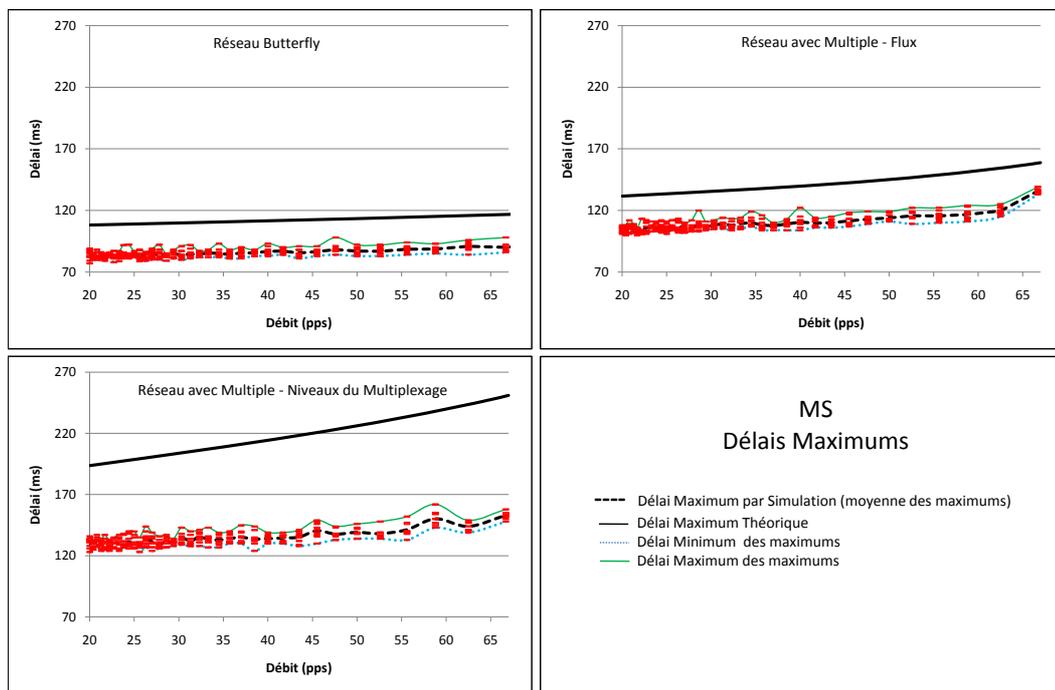


FIGURE 6.5 – Stratégie MS : Délais Maximums

6.7 Comparaisons des résultats de simulation obtenus par les trois stratégies

Les Figures 6.6 présente les moyennes des délais maximums pratiques de bout-en-bout des paquets sur les chemins les plus longs entre la source et le récepteur pour les trois réseaux quand Δ varie de 10 à 50 ms. Nous pouvons tout d'abord remarquer que,

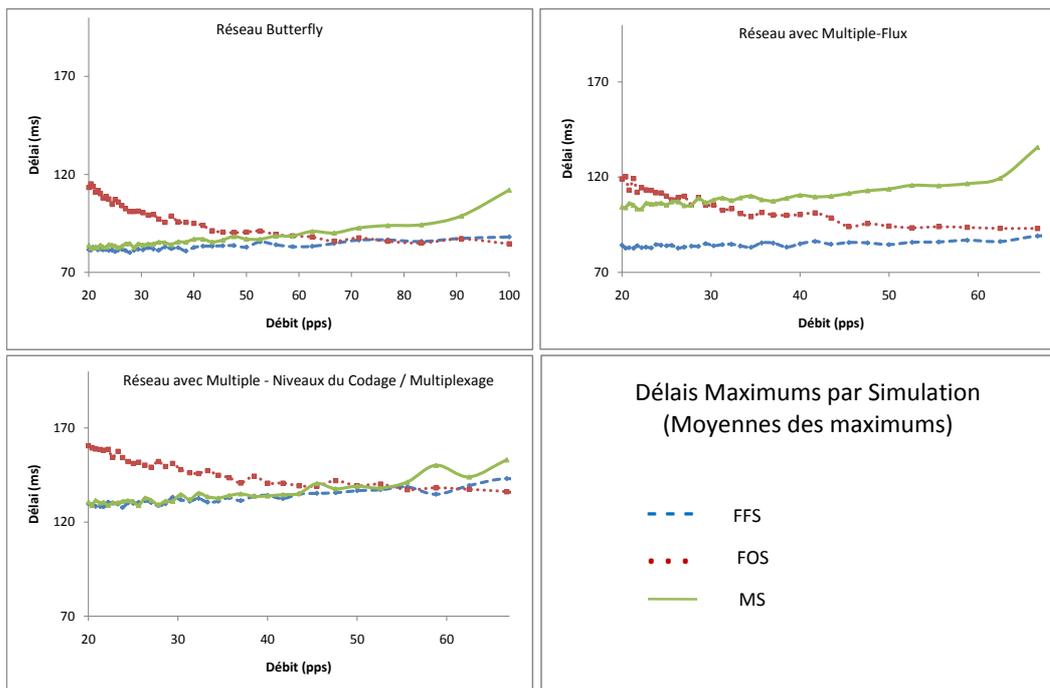


FIGURE 6.6 – Délais maximums par simulation

dans le pire cas, les résultats théoriques seront, au maximum, 2 fois plus grands que les résultats pratiques. Ça peut être expliqué du fait que les bornes sont calculées pour des pires cas avec le calcul réseau qui impose les pires des scénarios dans chaque élément des réseaux (en l'occurrence, des délais maximums). Notons que ce résultat est relativement positif car le calcul réseau fournit souvent des bornes théoriques beaucoup plus éloignés des résultats pratiques.

En observant les Figures 6.6, nous trouvons que

La *stratégie de transfert rapide* (FFS) présente les meilleurs résultats lorsque les débits ne sont pas trop importants par rapport à la capacité des liens. Ainsi, il est préférable d'utiliser la *stratégie de transfert rapide* lorsque les débits ne sont pas très importants (moins de $C/2$) ou lorsque nous traitons des flux à taux ou à régularité variable.

6.7. Comparaisons des résultats de simulation obtenus par les trois stratégies

En ce concerne la stratégie orientée flux (FOS), les délais augmentent lorsque les débits ne sont pas trop importants, mais d'un autre côté, la charge totale du réseau est considérablement réduite comparée aux autres stratégies (taux de combinaison maximum).

lorsque les débits augmentent, la stratégie FOS obtient d'excellents résultats. Elle est donc conseillée pour des flux dont les débits sont suffisamment grands (plus grand que $C/2$).

Dans la stratégie de Routage / Multiplexage (MS), nous remarquons que les délais de bout-en-bout augmentent quand les débits des flux dépassent les seuils. Autrement dit, quand la somme des débits des flux entrants devient plus grande que la capacité des liens sortants.

Comme prévu, ce seuil est atteint au point égal approximativement à 100 pps pour le réseau "butterfly" et 67 pps les deux autres réseaux.

Pour les stratégies à base de codage, les seuils équivalents apparaissent à des valeurs proches de la capacité des liens avant que la congestion commence à apparaître.

Cela signifie que les stratégies de codage sont capables de supporter des débits très près de la valeur maximale. Ce point présente l'un des avantages le plus important du codage réseau.

Enfin, il faut noter que le fait de considérer le délai de combinaison T_{lc} ne change pas les tendances des courbes des stratégies de codage (T_{lc} supposé être petit, donc son effet sur les résultats n'est pas énorme). Il suffit, pour le prendre en compte, de décaler les courbes vers le haut d'une certaine valeur équivalente à T_{lc} .

Chapitre 6. Évaluation et analyse du délai maximum par simulation

7

Conclusion

La contribution principale de ce travail est la définition de nouvelles solutions dont le but est d'introduire le codage réseau dans le contexte des réseaux à qualité de service garantie.

Pour établir les hypothèses spécifiques de travail, une exploration préalable de la problématique de ce domaine a été nécessaire. Même si plusieurs approches introduisant le codage réseau dans différents contextes et domaines d'applications ont été proposées (voir Section 2.2.1), le fait d'utiliser le codage réseau pour des applications avec la contrainte d'offrir un certain niveau de qualité de service à chaque paquets n'a pas été traité jusqu'à présent (à notre connaissance).

Ceci nous a motivé pour orienter nos recherches vers ce domaine avec des stratégies et des hypothèses assez spécifiques. A titre d'exemple quand nous nous sommes intéressés aux stratégies de codage sur les nœuds intermédiaires, plusieurs questions se posaient, notamment par rapport à la stratégie de codage à utiliser, à la manière de considérer les flux et les autres éléments des réseaux, à la façon de gérer la synchronisation des paquets. . . etc.

L'objectif initial a été globalement atteint. Rappelons que celui-ci consistait à montrer que le codage réseau peut permettre de réduire les bornes de délais de bout en bout dans des réseaux à qualité de service, y compris dans des réseaux où le codage réseau n'apporte pas de gain de débit.

Résumons maintenant l'ensemble de ce travail, du point de vue des thématiques abordées et des perspectives qu'il permet d'envisager.

- **Stratégie Orientée Réseau (NOS) :**

Nous avons introduit cette stratégie avec comme objectif de réduire au minimum les temps d'attente dans les nœuds de codage. Pour cette stratégie, qui est basée sur du codage aléatoire en termes de dates d'arrivée des paquets, nous avons présenté une méthode permettant d'obtenir les courbes de service globales du réseau. Cette méthode est basée sur une matrice de transfert dont les entrées sont

des courbes de service.

- **Stratégies FOS - FFS - MS :**

En raison de la fiabilité partielle de l'approche précédente, peu compatible avec des réseaux à QoS garantie, nous avons proposé deux autres stratégies à fiabilité totale utilisant le concept de code en bloc. La première, stratégie orientée flot (FOS) est une implémentation directe du codage réseau utilisant le concept de code en bloc, et la seconde, stratégie de transfert rapide (FFS) permet aux nœuds intermédiaires de transférer les paquets même si certains paquets à combiner ne sont pas encore arrivés.

Les délais maximums de bout-en-bout et les tailles des files d'attente ont été déterminées dans les pires cas pour ces deux dernières stratégies de codage réseau ainsi que pour la stratégie de routage / multiplexage classique dans des réseaux fournissant des QoS garanties.

- **Évaluations et résultats FOS - FFS - MS :**

Les résultats sur les délais maximums de bout-en-bout des deux stratégies fiables et de la stratégie de routage / multiplexage classique ont été évalués de deux manières sur trois topologies de réseaux : via l'approche théorique basé sur le calcul réseau et ensuite au moyen de simulations.

Nous avons montré qu'avec la stratégie de transfert rapide, les bornes obtenues sur le délai de bout-en-bout sont meilleures que les bornes obtenues avec la stratégie orientée flux tant que les débits de flux restent relativement petits. Par contre, cet avantage bascule et devient légèrement favorable à la stratégie orientée flux lorsque le débit augmente et est régulier. Par rapport à ces deux stratégies, les performances de la stratégie de multiplexage sont légèrement moins bonnes que celles de la stratégie de transfert rapide à très bas débit, par contre, l'écart se creuse lorsque les débits, et donc les congestions, augmentent.

Indépendamment des valeurs des débits des flux, la stratégie de transfert rapide (FFS) présente un meilleur compromis entre le délai et le débit et offre un meilleur service que les deux autres stratégies.

Tous ces résultats théoriques ont été confirmés par les résultats obtenus par des simulations. En effet, les courbes théoriques et de simulation suivent globalement les mêmes pentes et nous observons un facteur inférieur à 2 entre les bornes maximales théoriques et celles obtenues par simulation.

Au vu des résultats obtenus, nous souhaiterons approfondir nos études dans plusieurs directions.

Le première piste consiste à poursuivre les études des stratégies introduites dans le cadre des réseaux à garanties de QoS. Il nous semble intéressant d'intégrer des contraintes liés a certains réseaux spécifiques. Un premier type de réseaux concerne les réseaux avioniques, et notamment le réseaux AFDX, qui a motivé notre travail au début de cette étude. L'application de nos stratégies sur une topologie réaliste en intégrant les définitions des flux multicast utilisés dans ce contexte nous semble particulièrement intéressante. Une autre piste concerne les réseaux à garantie de service de type MPLS. En effet, le codage réseau est déjà

étudié dans ce contexte dans le cadre du projet européen FP7 4WARD [75][76]. L'autre direction de recherche concerne l'utilisation des stratégies que nous avons introduites, notamment la stratégie de transfert rapide, dans d'autres types de réseaux que des réseaux à QoS. En effet, si dans cette thèse, nous nous sommes focalisés sur les bornes maximums des délais de transmission, nous avons observé que la stratégie de transfert rapide obtient les mêmes gains en terme de délai moyen. Les résultats de simulations sur le réseau "butterfly" sont présentés sur la Figure 7.1.

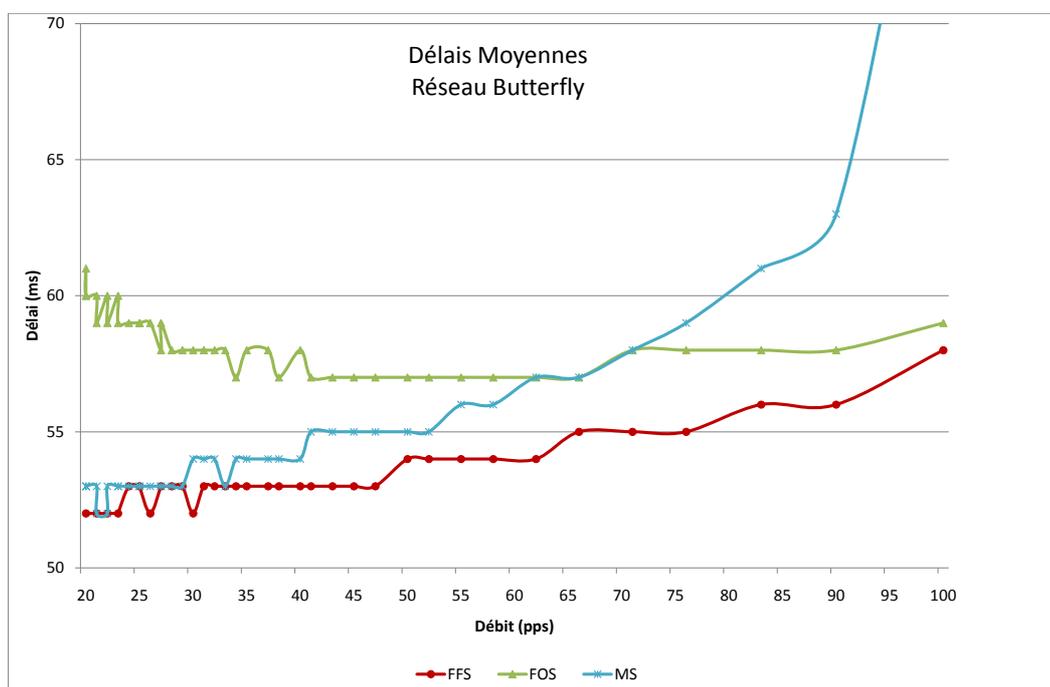


FIGURE 7.1 – Réseau "butterfly"- délais moyens

Cette stratégie peut être utilisée dans tous les réseaux où le code réseau est fixe. Elle permet d'obtenir une fiabilité totale grâce à l'utilisation du concept de code en bloc tout en s'affranchissant de délais d'attente inutiles sur les nœuds intermédiaires.

Chapitre 7. Conclusion

Bibliographie

- [1] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, vol. 46, pp. 1204-1216, July 2000.
- [2] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 5, pp. 782-795, October 2003.
- [3] R. Koetter and M. Médard. Beyond routing : An algebraic approach to network coding. In *Proceedings of the 21th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, vol. 1, pp. 122-130, June 2002.
- [4] Y. Zhu, B. Li, and J. Guo. Multicast with network coding in application-layer overlay networks. *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 107-120, January 2004.
- [5] P. A. Chou, Y. Wu, and K. Jain. Practical network coding. In *Proceedings of the 41st Allerton Conference on Communication, Control, and Computing*, October 2003.
- [6] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger. On randomized network coding. In *Proceedings of the 41st Allerton Conference on Communication, Control, and Computing*, October 2003.
- [7] Y. Wu, P. A. Chou, and S-Y. Kung. Information exchange in wireless networks with network coding and physical-layer broadcast. In *Proceedings of the 39th Conference on Information Sciences and Systems (CISS)*, March 2005.
- [8] Z. Li and B. Li. Network coding : The case of multiple unicast sessions. In *Proceedings of the 42st Allerton Conference on Communication, Control, and Computing*, October 2004.
- [9] A. Eryilmaz, A. Ozdaglar, and M. Medard. On delay performance gains from network coding. In *proceedings of the 40st Conference on Information Sciences and Systems (CISS)*, March 2006.
- [10] J-Y. Le Boudec and P. Thiran. *Network calculus : a theory of deterministic queuing systems for the internet*. Lecture Notes in Computer Science, Vol. 2050. Springer-Verlag New York, Inc., New York, NY, USA, 2001.
- [11] 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.
- [12] R. L. Cruz. A calculus for network delay, part II : Network analysis. *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 132-141, January 1991.

Bibliographie

- [13] R. W. Yeung, S-Y. R. Li, N. Cai, and Z. Zhang. *Network coding theory*. Foundation and Trends in Communications and Information Theory, vol. 2, no. 4 and 5, pp. 241-381, Now Publishers Inc., Hanover, MA, USA, 2005.
- [14] C. Fragouli and E. Soljanin. *Network coding fundamentals*. Foundation and Trends in Networking, vol.2, no. 1, pp. 1-133, Now Publishers Inc., Hanover, MA, USA, 2007.
- [15] C. Fragouli and E. Soljanin. *Network coding applications*. Foundation and Trends in Networking, vol.2, no. 2, pp. 135-269, Now Publishers Inc., Hanover, MA, USA, 2007.
- [16] R. Koetter. The network coding home page. <http://www.ifp.uiuc.edu/~koetter/NWC/>.
- [17] S-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371-381, 2003.
- [18] P. Sanders, S. Egner, and L. Tolhuizen. Polynomial time algorithms for network information flow. In *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms (SPAA), San Diego, California, USA*, pp. 286-294, June 2003.
- [19] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973-1982, June 2005.
- [20] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *Proceedings of IEEE International Symposium on Information Theory*, pp. 442+, July 2003.
- [21] C. Gkantsidis and P. Rodriguez. Network coding for large scale content distribution. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05), vol. 4*, pp. 2235-2245, March 2005.
- [22] Y. Wu, P. A. Chou, and S-Y. Kung. Minimum-energy multicast in mobile ad hoc networks using network coding. *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1906-1918, November 2005.
- [23] J. Widmer and J-Y. Le Boudec. Network coding for efficient communication in extreme networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN'05)*, pp. 284-291, August 2005.
- [24] Z. Li, B. Li, D. Jiang, and L. C. Lau. On achieving optimal throughput with network coding. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05), vol. 3*, pp. 2184-2194, March 2005.
- [25] Y. Ma, W. Li, P. Fan, and X. Liu. Queuing model and delay analysis on network coding. *IEEE International Symposium on Communications and Information Technology (ISCIT) Vol. 1*, pp. 112-115, October 2005.
- [26] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Médard. The importance of being opportunistic : Practical network coding for wireless environments. In *Proceedings of the 43st Allerton Conference on Communication, Control, and Computing*, September 2005.

- [27] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air : practical wireless network coding. *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 243-254, October 2006.
- [28] S. Zhang, S-C. Liew, and P. P. Lam. Physical layer network coding. *Computing Research Repository (CoRR)*, vol. abs/0704.2475, April 2007.
- [29] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference : Analog network coding. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '07)*, pp. 397-408. ACM, 2007.
- [30] Y. Hao, D. Goeckel, Z. Ding, D. Towsley, and K. K. Leung. Achievable rates of physical layer network coding schemes on the exchange channel. *IEEE Military Communications Conference (MILCOM'07)*, October 2007.
- [31] O. H. Abdelrahman and E. Gelenbe. Queueing performance under network coding. In *Proceedings of IEEE Information Theory Workshop (ITW'09)*, pp. 135-139, June 2009.
- [32] R. Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer-Verlag New York, LLC, New York, NY, USA, August 2005.
- [33] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. Toward a random operation of networks. *IEEE Transactions on Information Theory*, vol. 2001, pp. 1-8, 2004.
- [34] C. Fragouli, J-Y. Le Boudec, and J. Widmer. Network coding : an instant primer. *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63-68, January 2006.
- [35] T. Ho, R. Kotter, M. Médard, M. Effros, J. Shi, and D. Karger. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, vol. 52, pp. 4413-4430, October 2006.
- [36] T. Ho, B. Leong, M. Médard, R. Kotter, Y. H. Chang, and M. Effros. On the utility of network coding in dynamic environments. *International workshop on wireless Ad hoc networks (IWWAN)*, 2004.
- [37] D.S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter. An analysis of finite-memory random linear coding on packet streams. In *proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 1-6, April 2006.
- [38] B. Shrader and A. Ephremides. On the queueing delay of a multicast erasure channel. *IEEE Information Theory Workshop, ITW '06 Chengdu*, pp. 423-427, October 2006.
- [39] B. Shrader and A. Ephremides. A queueing model for random linear coding. *IEEE Military Communications Conference (MILCOM'07)*, October 2007.
- [40] C. K. Ngai and R. W. Yeung. Network coding gain of combination networks. *IEEE Information Theory Workshop*, pp. 283-287, October 2004.

Bibliographie

- [41] T. Noguchi, T. Matsuda, and M. Yamamoto. Performance evaluation of new multicast architecture with network coding, 2003.
- [42] Y. Wu. Network coding for multicasting. *Ph.D. Dissertation, Princeton University*, November 2005.
- [43] Y. Wu, P. Chou, Q. Zhang, K. Jain, W. Zhu, and S-Y. Kung. Network planning in wireless ad hoc networks : a cross-layer approach. *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 136-150, January 2005.
- [44] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft. Internet RFC 3453 : The use of forward error correction (FEC) in reliable multicast, December 2002.
- [45] C. Fragouli, D. Lun, M. Médard, and P. Pakzad. On feedback for network coding. In *proceedings of the 41st Conference on Information Sciences and Systems (CISS)*, March 2007.
- [46] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 56-67, October 1998.
- [47] M. Luby. Lt codes. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'02)*, pp. 271-280, 2002.
- [48] D. S. Lun, M. Médard, R. Koetter, and M. Effros. On coding for reliable communication over packet networks. *Physical Communication*, vol. 1, no. 1, pp. 3-20, March 2008.
- [49] A. Albanese, J. Blomer, J. Edmonds, and M. Luby. Priority encoding transmission. *IEEE Transactions on Information Theory*, vol. 42, pp. 1737-1744, 1996.
- [50] P. A. Chou and Y. Wu. Network coding for the internet and wireless networks. *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 77-85, September 2007.
- [51] S. Deb, M. Effros, T. Ho, D. R. Karger, R. Koetter, D. S. Lun, M. Médard, and N. Ratnakar. Network coding for wireless applications : A brief tutorial. In *Proceedings of International Workshop on Wireless Ad-hoc Networks (IWVAN)*, May 2005.
- [52] Y. Chen, S. Kishore, and J. Li. Wireless diversity through network coding. In *Proceeding of IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, pp. 1681-1686, April 2006.
- [53] A. Al Hamra, C. Barakat, and T. Turletti. Network coding for wireless mesh networks : A case study. In *Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 103-114, 2006.
- [54] A. Ponnappan, L. Yang, R. Pillai, and P. Braun. A policy based qos management system for the intserv/diffserv based internet. In *Proceedings of the Third IEEE International Workshop on Policies for Distributed Systems and Networks*, vol. 0, pp. 159-168, 2002.
- [55] P. Ferguson and G. Huston. *Quality of service : delivering QoS on the Internet and in corporate networks*. John Wiley & Sons, Inc., New York, NY, USA, 1998.

- [56] V. Ræisaenen. *Implementing Service Quality in IP Networks*. John Wiley & Sons, Inc., New York, NY, USA, 2003.
- [57] D. McDysan. *QoS and traffic management in IP and ATM networks*. McGraw-Hill, Inc., New York, NY, USA, 2000.
- [58] J. Grieu. Study and evaluation of the switched ethernet technology for avionic systems interconnection. *Ph.D. Dissertation, Institut National Polytechnique de Toulouse*, September 2004.
- [59] Airlines Electronic Engineering Committee. *ARINC Specification 664 : Aircraft data network, part 7 : Avionics Full Duplex Switched Ethernet (AFDX) Network*. ARINC Aeronautical Radio Inc., May 2003.
- [60] Airlines Electronic Engineering Committee. *ARINC Specification 664 : Aircraft Data Network, parti 1 : Systems Concepts and Overview*. ARINC Aeronautical Radio Inc., January 2002.
- [61] C. S. Chang. Stability, queue length and delay of deterministic and stochastic queueing networks. *IEEE Transactions on Automatic Control*, vol. 39, no. 5, pp. 913-931, May 1994.
- [62] F. Ciucu, A. Burchard, and J. Liebeherr. A network service curve approach for the stochastic analysis of networks. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS'05)*, vol. 33, no. 1, pp. 279-290, 2005.
- [63] A. Mahmino, J. Lacan, and C. Fraboul. Calculus of service guarantees for network coding. In *Proceedings of the 2006 International Symposium on Information Theory and its Applications (ISITA'06)*, October 2006.
- [64] A. Mahmino, J. Lacan, and C. Fraboul. Enhancing guaranteed delays with network coding. In *Proceedings of the 6th International IFIP-TC6 Networking Conference (NETWORKING'07)*, vol. 4497, pp. 1229-1232, Lecture Notes in Computer Science. Springer, May 2007.
- [65] A. Mahmino, J. Lacan, and C. Fraboul. Guaranteed packet delays with network coding. In *the 5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops (SECON Workshops'08)*, pp. 1-6, June 2008.
- [66] J. Kahn and J. Komlós. Singularity probabilities for random matrices over finite fields. *Combinatorics, Probability and Computing*, vol. 10, no. 2, pp. 137-157, October 2001.
- [67] Z. Guo, B. Wang, J-H. Cui, P. Xie, and W. Zeng. Efficient error recovery with network coding in underwater sensor networks. *Ad Hoc Networks*, vol. 7, no. 4, pp. 791-802, June 2009.
- [68] J. Postel. RFC 768 : User datagram protocol. <http://www.faqs.org/rfcs/rfc768.html>, August 1980.
- [69] J. Stone, R. Stewart, and D. Otis. RFC 3309 : Stream control transmission protocol (SCTP) checksum change. <http://www.faqs.org/rfcs/rfc3309.html>, September 2002.

Bibliographie

- [70] P. J. Mumford. Timing characteristics of the 1pps output pulse of three gps receivers. In *Proceedings of the 6th International Symposium on Satellite Navigation Technology Including Mobile Positioning and Location Services*, July 2003.
- [71] D. L. Mills. RFC 1305 : Network time protocol (version 3) specification, implementation. <http://www.faqs.org/rfcs/rfc1305.html>, March 1992.
- [72] D. L. Mills. Precision synchronization of computer network clocks. *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 2, pp. 28-43, April 1994.
- [73] A. El Fawal, K. Salamatian, D. Cavin, Y. Sasson, and J-Y. Le Boudec. A framework for network coding in challenged wireless network. <http://icawww1.epfl.ch/netcod>, 2006.
- [74] R. Koetter. Software related to network coding. <http://www.ifp.uiuc.edu/~koetter/NWC/Software.html>.
- [75] The fp7 4ward project. <http://www.4ward-project.eu>.
- [76] The fp7 4ward project : D-5.2.0 description of generic path mechanism. <http://www.4ward-project.eu/index.php?id=65>.



Université
de Toulouse



Institut Supérieur de l'Aéronautique et de l'Espace



AUTEUR : MAHMINO Ali

TITRE : Application du Codage Réseau aux Architectures à Garanties de
Qualité de Service (QoS)

DIRECTEUR DE THESE : J. LACAN et C. FRABOUL

LIEU ET DATE DE SOUTENANCE : ISAE (Campus ENSICA), Toulouse.
Le 12 Novembre 2009

RESUME EN FRANÇAIS :

L'intérêt du codage réseau (network coding) pour améliorer le débit ou optimiser l'utilisation de la capacité du réseau a été clairement démontré dans différents contextes. Certains travaux ont notamment montré que le codage réseau permet de diminuer le délai (maximal et moyen) de transmission de bout-en-bout d'un paquet. Ceci est dû au fait que le traitement simultané de plusieurs paquets dans un noeud de codage permet de réduire le temps passé par les paquets dans les files d'attente par rapport au routage classique. Dans cette thèse, nous considérons l'application du codage réseau dans le contexte des réseaux proposant des garanties de qualité de service (QoS).

Notre principale contribution est la proposition de trois stratégies de codage réseau assurant un niveau de QoS garantie exprimé en termes de délai de bout-en-bout. La première stratégie, appelée "stratégie orientée réseau" est une stratégie de codage aléatoire, en termes de dates d'arrivée des paquets, permettant de réduire au maximum le temps passé par les paquets dans les files d'attente des routeurs. Le point faible de cette approche, comme toute approche aléatoire, est qu'elle n'est pas totalement fiable. Les deux autres stratégies proposées implémentent une stratégie fiable en utilisant le concept de code en bloc. La première, appelée "stratégie orientée flux" est basée sur la définition classique du codage réseau alors que la seconde, appelée "stratégie de transfert rapide", permet de réduire les temps d'attente des paquets dans les files d'attente en les transférant sans attendre tous les paquets du même bloc. Les délais maximums engendrés par les différentes stratégies ont été évalués au niveau d'un noeud de codage en utilisant le calcul réseau (network calculus). Les bornes de délais de bout-de-bout ont ensuite été calculées pour plusieurs types de réseaux. Dans la plupart des cas, ces bornes sont meilleures que celles obtenues pour le routage classique. Les stratégies de codage réseau fiables et la stratégie de routage ont été implémentées et évaluées par simulation sur les réseaux étudiés précédemment. Les résultats obtenus montrent que les pires cas de délais de bout-en-bout observés ont les mêmes comportements que les bornes maximales théoriques calculées, validant ainsi les stratégies proposées.

MOTS-CLES : Codage Réseau, Délai, Stratégie de Codage, Noeud de Codage, Routage, Calcul Réseau.

DISCIPLINE : Informatique, Réseaux

ISAE (Institut Supérieur de l'Aéronautique et de l'Espace) :

10 avenue Edouard Belin - BP 54032 - 31055 Toulouse cedex 4

Tél. : (33) 05 61 33 80 80 - Fax : (33) 05 61 33 83 30