

n° ordre 2134

ECOLE DOCTORALE INFORMATIQUE ET  
TELECOMMUNICATIONS

**Analyse et évaluation de techniques de  
commutation Ethernet pour l'interconnexion  
des systèmes avioniques**

Thèse soutenue le 24 septembre 2004 pour obtenir le grade de Docteur en Réseaux et  
Télécommunications de l'Institut National Polytechnique de Toulouse

Par

**Jérôme Grieu**

Sous la direction de

**Christian Fraboul, IRIT/ENSEEIH  
Fabrice Frances, ENSICA**

Composition du jury

**M. COURTIAT Jean-Pierre**, Directeur de recherche, LAAS  
**M. ELLOY Jean-Pierre**, Professeur des Universités, INPL  
**M. THOMESSE Jean-Pierre**, Professeur des Universités, ECN  
**M. SAINT-ETIENNE Jean-François**, Ingénieur, AIRBUS  
**M. FRABOUL Christian**, Professeur des Universités, INPT  
**M. FRANCES Fabrice**, Enseignant Chercheur, ENSICA

Président  
Rapporteur  
Rapporteur  
Examineur  
Directeur de thèse  
Codirecteur de thèse



## Remerciements

*Je remercie mes encadrants Christian Fraboul et Fabrice Frances.*

*Je remercie mes interlocuteurs chez Airbus, au premier rang desquels Lionel Loudet.*

*Merci à Julien Fasson et Fabrice Arnal qui ont sû être les confidents des moments difficiles.*

*Merci à tous les membres de TeSA que j'ai pu côtoyer et qui m'ont permis de travailler dans une ambiance chaleureuse : Sylvie, André-Luc, Manu, Béatrice, Florestan, Fred, Ridha, les filles, tout le monde !*

*Merci à tous ceux qui sont venus m'encourager et me soutenir lors de la soutenance.*

*Merci enfin à ma famille : pour tout, mais vous le savez déjà !*



# Table des matières

Introduction .....	1
1 Les systèmes avioniques : historique et évolution .....	1
2 Le problème de preuve de déterminisme .....	2
3 Organisation du mémoire .....	2
CHAPITRE 1 .....	5
Le réseau avionique étudié .....	5
1 Architecture des systèmes avioniques .....	5
1.1 Evolution des systèmes avioniques .....	5
1.2 Classification des systèmes avioniques .....	6
1.3 Architectures classiques .....	7
1.3.1 Architecture centralisée .....	7
1.3.2 Architecture fédérale .....	7
1.3.3 Architecture répartie .....	7
1.4 Avionique modulaire intégrée .....	8
2 Ethernet commuté Full Duplex .....	9
2.1 La méthode d'accès Ethernet .....	9
2.2 Réseau Ethernet commuté .....	10
2.3 Réseau Ethernet commuté Full-Duplex .....	11
3 La norme ARINC 664 .....	12
3.1 Domaine d'application .....	12
3.2 Restrictions imposées pour les réseaux profilés .....	12
3.2.1 Configuration statique .....	13
3.2.2 Isolation des erreurs .....	13
3.2.3 Trafic maîtrisé .....	13
4 Le réseau AFDX .....	14
4.1 Les objectifs du réseau .....	14
4.2 La notion de VL .....	14
4.3 Services de communication .....	15
5 La problématique de certification du réseau .....	15
CHAPITRE 2 .....	17
Méthode de preuve de déterminisme par calcul réseau .....	17
1 Les travaux fondateurs de Cruz .....	17
1.1 Modélisation des flux .....	17
1.2 Modélisation des éléments simples .....	18
1.2.1 Le lien à délai constant .....	18
1.2.2 Le tampon de réception .....	18
1.2.3 Le démultiplexeur .....	19
1.3 Modélisation d'un multiplexeur .....	19
1.3.1 Bornes sur le délai .....	20
1.3.2 Bornes sur les enveloppes en sortie .....	20
1.3.3 Bornes sur l'arrière de travail .....	20
1.3.4 Remarque sur l'utilisation de ces résultats .....	20
1.4 Méthode de calcul sur un réseau complet .....	21
2 Pourquoi le Network Calculus .....	22
2.1 Intérêt de la méthode .....	22
2.2 Applicabilité du Network Calculus .....	22

2.2.1	Construction d'un modèle .....	22
2.2.2	Flux d'entrée .....	23
2.2.3	Topologie du réseau et routage des flux.....	23
3	Application du Network Calculus au réseau étudié .....	24
3.1	Description du cas d'étude .....	24
3.2	Modélisation du trafic .....	26
3.2.1	Enveloppe initiale.....	26
3.2.2	Modélisation du caractère multicast des VL.....	26
3.3	Modélisation des éléments du réseau .....	26
3.3.1	L'abonné.....	26
3.3.2	Les liens.....	27
3.3.3	Le commutateur.....	28
3.4	Méthode de calcul mise en oeuvre .....	31
3.5	Description de l'outil réalisé .....	32
4	Etude du réseau au moyen de l'outil développé.....	34
4.1	Calcul avec une politique FIFO .....	34
4.2	Comparaison avec d'autres politiques de service .....	36
4.2.1	Politique conservatrice générique .....	36
4.2.2	Politique à deux niveaux de priorité.....	37
4.3	Interprétation physique de ces résultats .....	40
5	Conclusion.....	40
	CHAPITRE 3 .....	41
	Extensions de la méthode de preuve du déterminisme .....	41
1	Network Calculus généralisé.....	41
1.1	Introduction de la notion de courbe de service .....	41
1.2	Le formalisme de Le Boudec .....	42
1.2.1	Caractérisation des éléments par la notion de courbe de service.....	42
1.2.2	Opérateurs pour le calcul réseau .....	43
1.2.3	Reformulation des théorèmes fondamentaux du calcul réseau .....	44
1.2.4	Courbe de service maximale .....	45
1.3	Application au réseau étudié .....	46
1.3.1	Description des flux .....	46
1.3.2	Description des éléments.....	46
1.3.3	Méthode de calcul .....	46
2	Proposition de la notion de groupe.....	47
2.1	Motivation .....	47
2.2	Caractérisation des groupes.....	48
2.2.1	Formation d'un groupe.....	48
2.2.2	Enveloppe de groupe .....	48
2.2.3	Lien avec la notion de courbe de service maximale.....	49
2.3	Intégration de la notion de groupe dans le calcul.....	50
2.3.1	Propagation de l'enveloppe d'un groupe .....	50
2.3.2	Fin de la propagation.....	50
2.3.3	Multiplexeurs et groupes.....	50
2.3.4	Pratique du calcul .....	51
2.4	Gains apportés .....	51
3	Analyse en trames .....	52
3.1	Mode de calcul en trames.....	53
3.1.1	Enveloppes en trames.....	53
3.1.2	Courbe de service en trames.....	56

3.1.3	Enveloppes de groupe en trames .....	57
3.1.4	Applicabilité des théorèmes .....	57
3.1.5	Stabilité des éléments dans le domaine des trames .....	57
3.2	Proposition d'une méthode de calcul en trames .....	58
3.2.1	Calcul direct à partir des bornes en bits .....	58
3.2.2	Deuxième méthode .....	59
4	Prise en compte des priorités .....	60
4.1	Définition de la courbe de service .....	60
4.2	Impact sur le calcul .....	60
4.2.1	Propagation de l'enveloppe .....	60
4.2.2	Influence sur les groupes .....	61
4.3	Application .....	61
	Conclusion .....	62
	CHAPITRE 4 .....	63
	Méthodes d'optimisation du réseau .....	63
1	Affectation des priorités aux VL .....	64
1.1	Complexité du problème .....	64
1.2	Algorithmes classiques .....	65
1.2.1	Une variation sur l'algorithme Alpha bêta .....	65
1.2.2	Une variation sur la méthode du gradient .....	67
1.3	Heuristiques d'affectation .....	68
1.3.1	Heuristique « $S_{min}$ » .....	68
1.3.2	Autres heuristiques envisagées .....	74
1.3.3	Améliorations de l'heuristique $S_{min}$ .....	75
1.3.4	Synthèse .....	76
1.4	Algorithmes d'évolution .....	76
1.4.1	Généralités sur les algorithmes d'évolution [45] .....	76
1.4.2	Généralités sur les algorithmes multicritères [70] .....	80
1.4.3	Application des algorithmes génétiques multicritères au problème étudié .....	82
1.4.4	Résultats obtenus sans choix de la population initiale .....	85
1.4.5	Résultats obtenus avec population initiale choisie .....	89
1.4.6	Apport des algorithmes génétiques au problème .....	91
1.5	Synthèse sur l'affectation des priorités aux VL .....	91
2	Optimisation de l'architecture matérielle du réseau .....	92
2.1	Mode d'interconnexion des commutateurs .....	92
2.1.1	Motivations .....	92
2.1.2	Réalisation .....	92
2.1.3	Résultats locaux .....	93
2.1.4	Influence sur le reste du réseau .....	94
2.1.5	Synthèse .....	95
2.2	Politique de service des ports de sortie des commutateurs .....	95
2.2.1	Motivations .....	95
2.2.2	Première approche : régulation atténuée .....	96
2.2.3	Deuxième approche : régulation adaptative .....	98
2.2.4	Synthèse .....	100
2.3	Synthèse sur l'optimisation de l'architecture matérielle .....	100
	Conclusion .....	101
	CHAPITRE 5 .....	103
	Une approche combinée déterministe stochastique .....	103
1	Motivations .....	103

1.1	Le caractère pessimiste des bornes déterministes .....	103
1.2	Apport des méthodes stochastiques.....	105
2	Principe d'une approche combinée déterministe stochastique.....	106
2.1	Etat de l'art sur les garanties statistiques .....	106
2.1.1	Hypothèse des sources on-off .....	106
2.1.2	Inégalités de Hoeffding .....	106
2.1.3	Enveloppes à $\varepsilon$ près .....	107
2.2	Présentation des travaux de Vojnović .....	107
2.2.1	Contexte .....	107
2.2.2	Principales hypothèses .....	108
2.2.3	Méthode.....	109
2.2.4	Principaux résultats .....	109
2.2.5	Autres résultats.....	111
3	Etude des bornes obtenues par Vojnović .....	112
3.1	Etude des facteurs influents de ces bornes .....	112
3.1.1	Influence du nombre de pas de subdivision .....	113
3.1.2	Influence du nombre de sources.....	113
3.1.3	Influence de la charge des sources .....	115
3.1.4	Influence de la dispersion des sources .....	116
3.2	Confrontation des bornes de Vojnović à une simulation .....	116
4	Application de l'approche combinée au réseau étudié.....	117
4.1	Vérification des hypothèses de la méthode .....	118
4.1.1	Indépendance des flux .....	118
4.1.2	Régulation des flux.....	118
4.1.3	Débit à long terme .....	119
4.1.4	Caractérisation des éléments .....	119
4.2	Propositions de méthodes d'application des bornes au réseau complet.....	119
4.3	Résultats obtenus.....	119
4.3.1	Première méthode.....	119
4.3.2	Deuxième méthode.....	120
4.4	Obtention de bornes en trames .....	122
4.5	Obtention de bornes sur le délai .....	123
4.6	Illustration de l'intérêt pratique de ces bornes .....	124
4.6.1	Premier exemple.....	124
4.6.2	Deuxième exemple .....	125
	Conclusion.....	125
	CONCLUSION .....	127
1	Résumé des travaux.....	127
2	Synthèse de la démarche .....	128
3	Perspectives de recherche.....	129
	BIBLIOGRAPHIE .....	131



## Table des illustrations

Figure 1 Relais d'une trame .....	10
Figure 2 Le réseau de test.....	25
Figure 3 Modélisation d'un abonné .....	27
Figure 4 Format d'une trame Ethernet.....	28
Figure 5 Chaîne de traitement d'une trame.....	29
Figure 6 Modèle du commutateur .....	30
Figure 7 Extrait de réseau.....	33
Figure 8 Le VL A6-4.....	33
Figure 9 Commutateurs 1,3,5,7 avec politique FIFO.....	35
Figure 10 Commutateurs 1,3,5,7 avec GMUX .....	37
Figure 11 Commutateurs 1,3,5,7 avec double FIFO .....	39
Figure 12 Deux éléments consécutifs.....	43
Figure 13 Courbe de service cumulée .....	44
Figure 14 Déviation horizontale entre deux courbes .....	44
Figure 15 Déviation verticale entre deux courbes.....	45
Figure 16 Exemple de l'utilité des groupes .....	47
Figure 17 Modifications des courbes d'arrivée par les groupes.....	49
Figure 18 Nombre de flux et de groupes utilisés par le calcul dans chaque port.....	51
Figure 19 Comparaison analyse avec ou sans groupes .....	52
Figure 20 Schéma d'arrivée des trames .....	53
Figure 21 Second jeu d'hypothèses .....	54
Figure 22 Schéma d'arrivée pire cas en termes de trames.....	55
Figure 23 Schéma d'arrivée pire cas en termes de bits.....	56
Figure 24 Courbe de service en trames d'un multiplexeur .....	57
Figure 25 Illustration de l'instabilité .....	58
Figure 26 Résultats obtenus à partir des bornes en bits et de la taille minimale.....	59
Figure 27 Comparaison des résultats des deux méthodes .....	59
Figure 28 Exemple d'affectation de priorités .....	61
Figure 29 Cas d'une autre affectation de priorités.....	62
Figure 30 Evolution de la taille max durant l'heuristique .....	69
Figure 31 Evolution du nombre de groupes durant l'heuristique.....	70
Figure 32 Evolution de la taille min pour les groupes de priorité basse durant l'heuristique ..	70
Figure 33 Evolution de somme des Smin durant l'heuristique.....	71
Figure 34 Evolution de la taille max en bits durant l'heuristique.....	72
Figure 35 Evolution du nombre total de groupes durant l'heuristique .....	72
Figure 36 Evolution de la moyenne des bornes sur les tailles durant l'heuristique.....	73
Figure 37 Evolution de l'espérance du délai durant l'heuristique.....	73
Figure 38 Evolution de la taille max durant l'heuristique utilisant l'effet de groupe .....	74
Figure 39 Evolution de la taille max durant l'heuristique Smax .....	75
Figure 40 Evolution de la taille max pour d'autres heuristiques .....	75
Figure 41 Plan des objectifs, notion de « mieux » .....	80
Figure 42 Evocation du front de Pareto .....	81
Figure 43 Architecture PISA.....	83
Figure 44 Evolution de la population au cours de l'algorithme.....	86
Figure 45 Evolution de la dispersion de la population au cours de l'algorithme .....	86
Figure 46 Evolution de la valeur moyenne des individus au cours de l'algorithme.....	87

Figure 47 Evolution de la population au cours de l'algorithme, nouveaux paramètres .....	88
Figure 48 Evolution de la valeur moyenne des individus avec ces nouveaux paramètres.....	88
Figure 49 Evolution de la population au cours de l'algorithme mono-objectif.....	89
Figure 50 Evolution de la population au cours de l'algorithme, initialisation choisie .....	90
Figure 51 Zoom sur la population finale .....	90
Figure 52 Heuristique Smin en utilisant les flux passant par le port doublé.....	93
Figure 53 Heuristique Smin utilisant les groupes passant par le port doublé .....	93
Figure 54 Maximum des bornes sur SW-4 au cours de l'heuristique déplaçant les flux.....	94
Figure 55 Maximum des bornes sur SW-4 au cours de l'heuristique déplaçant les groupes ...	94
Figure 56 Evolution des bornes en fonction du seuil de régulation .....	97
Figure 57 Evolution des bornes en fonction de la régulation, trafic triplé.....	98
Figure 58 Evolution des bornes en fonction du seuil de régulation .....	99
Figure 59 Augmentation de l'ordonnée à l'origine des enveloppes lors de la traversée du réseau.....	100
Figure 60 Borne calculée et valeurs réelles.....	104
Figure 61 Effet du nombre de subdivisions .....	113
Figure 62 Effet du nombre de flux .....	114
Figure 63 Effet de la charge .....	115
Figure 64 Effet de la dispersion .....	116
Figure 65 Comparaison résultats théoriques / simulation .....	117
Figure 66 Application avec les groupes .....	120
Figure 67 Application "fresh traffic majorization" aux groupes sortant des ES.....	121
Figure 68 Application "fresh traffic majorization" avec l'hypothèse flux indépendants .....	122
Figure 69 Cas de la taille comptée en nombre de trames.....	123
Figure 70 Répartition des délais (ms) .....	124

## Introduction

### 1 Les systèmes avioniques : historique et évolution

L'augmentation du nombre et de la complexité des fonctions embarquées dans les aéronefs actuels implique un accroissement très important de la quantité de données échangées. Les architectures avioniques existantes reposent sur des bus avioniques de conception assez ancienne. Parmi ceux-ci, on peut citer le bus ARINC 429, bus mono émetteur très répandu, fiable et d'une grande simplicité d'utilisation mais aux performances limitées (débit maximal de 100 kbits/s). De plus l'accroissement des fonctions, et leur distribution sur plusieurs équipements, s'accompagnent non seulement d'une augmentation du volume total de données échangées, mais aussi d'une augmentation du nombre d'interconnexions nécessaires : les échanges ne suivent plus un simple schéma producteur – consommateur, mais ont lieu entre de nombreux intervenants. Il est évident dans un tel contexte que les bus mono émetteurs brident la croissance du système avionique. Pour répondre à cette augmentation des interconnexions, Boeing a choisi d'embarquer dans le cockpit du 777 des bus ARINC 629, bus multi émetteurs autorisant de bien meilleures performances (2 Mbps, jusqu'à 120 utilisateurs). Il apparaît cependant que cette amélioration de bus avionique implique des coûts de développement de composants spécifiques très importants. Ces différents facteurs font que la technologie des bus avioniques n'est plus comme il y a quelques années considérée comme une solution satisfaisante au problème de l'évolution du trafic dans les réseaux avioniques civils.

La solution retenue par Airbus pour l'A380 repose sur la technologie de l'Ethernet commuté. Le principal avantage de l'utilisation d'Ethernet est qu'il s'agit d'un standard bien connu, ce qui permet une réutilisation d'outils de développement ainsi que de composants matériels existants (COTS : commercial off-the-shelf). De plus, il existe une longue expérience industrielle de ce standard, ce qui permet d'avoir une bonne confiance en la fiabilité du matériel et sur la facilité de sa maintenance. L'inconvénient majeur de l'Ethernet, vis-à-vis d'une application dans des systèmes avioniques, est le non-déterminisme intrinsèque de sa méthode d'accès au support physique, CSMA/CD. La solution Ethernet commuté Full Duplex retenue permet de contourner cet inconvénient. Dans cette architecture, chaque équipement est seulement relié à un commutateur, par le biais d'un câble Full Duplex. De cette manière, il ne peut plus y avoir de collision sur le support physique, et le CSMA/CD n'est plus nécessaire. L'interconnexion des éléments repose donc sur des commutateurs Ethernet, reliés entre eux par la même technologie.

Une réflexion importante a été mise en place, autour du projet de norme ARINC 664, quant aux adaptations qu'il est nécessaire d'effectuer pour utiliser cette architecture dans le contexte aéronautique, et pouvoir ainsi profiter d'une technologie bien plus performante que les bus avioniques classiques (100 Mbps), tout en utilisant des composants existants. Le projet de norme prescrit en premier lieu de contrôler les flux avant leur entrée dans le réseau, grâce à la notion de Virtual Link. D'autre part, il spécifie que des commutateurs compatibles avec la norme 802.1D, dans leur version avionable, devront être utilisés.

## 2 Le problème de preuve de déterminisme

Si l'architecture proposée permet d'éliminer l'indéterminisme inhérent à l'Ethernet classique, il reste toutefois possible pour un commutateur compatible 802.1D de perdre des trames, par débordement de ses files d'attente. Or, comme tout équipement embarqué, il faut faire passer au système avionique l'épreuve de la certification, d'autant plus que le réseau de communication sera utilisé pour faire transiter des données d'application de niveau de criticité très élevé. En accord avec les autorités de certification, la démarche retenue a été de certifier séparément le réseau et le reste du système avionique (notamment toute la partie applicative). De cette façon, la certification du système avionique pourra utiliser comme hypothèse le fait que le réseau fournisse une certaine qualité de service, et s'en servir pour appliquer notamment des méthodes formelles.

La qualité de service que l'on cherche à assurer pour les applications s'appuie principalement sur les besoins des applications. On veut pouvoir garantir :

- un délai de traversée du réseau borné, pour préserver le temps de réponse des applications
- une gigue bornée, car de nombreuses applications utilisent les variations temporelles des données
- des pertes de trames très faibles, car les contraintes temps réel fortes interdisent les retransmissions.

Pour apporter ces garanties, qui ne découlent pas directement de l'architecture retenue comme nous l'avons vu, il est nécessaire de mettre en place une méthode de preuve de déterminisme.

Pour ce faire, une approche par simulation ne semble pas adaptée, car il est difficile de prouver avec certitude que la simulation recouvre tous les cas envisageables.

Nous avons alors choisi d'employer les résultats du Network Calculus, qui offrent des garanties de type déterministe (par calcul de bornes maximales), avec une méthode de complexité réduite.

## 3 Organisation du mémoire

Nous avons voulu voir dans une première étude si le Network Calculus, tel qu'il a été présenté par Cruz, donne des résultats intéressants dans notre contexte (**Chapitre 2**). Notre réseau de test est un prototype de réseau fourni par notre partenaire industriel, représentatif de la complexité du réseau qui sera réellement employé. Nous avons développé pour notre étude une modélisation de ce réseau avionique, et du trafic y circulant, en reprenant en partie les éléments décrits dans les travaux de Cruz. Nous explicitons ensuite les raisons pour lesquelles il est possible d'utiliser cette méthode dans notre contexte, alors que le Network Calculus a été développé a priori pour des réseaux à qualité de service de type DiffServ. La principale d'entre elles tient à la recommandation du projet de norme ARINC 664, selon laquelle les flux entrants dans le réseau doivent être mis en forme par chaque émetteur, pour limiter l'importance des rafales de trafic. En pratique, cette mise en forme du trafic est très proche de la caractérisation du trafic qui est l'idée forte des travaux de Cruz. Cette étude nous permet de présenter une méthode d'analyse de réseau avionique, qui bien que relativement sommaire, conduit à des résultats intéressants. D'une part, elle permet d'établir des bornes sur les délais (locaux ou de bout en bout) et les giges, ainsi que sur les tailles maximales des files d'attente. D'autre part, l'ordre de grandeur de ces bornes permet d'envisager de certifier le réseau de cette manière.

Cependant, les résultats obtenus ne sont pas totalement satisfaisants puisque les bornes obtenues semblent trop importantes. Nous avons alors décidé de raffiner notre méthode d'analyse du réseau, pour essayer d'améliorer les bornes obtenues (**Chapitre 3**). Une seconde version de la méthode a alors été développée, prenant en compte les améliorations apportées au Network Calculus au cours des dernières années, et notamment la notion de courbe de service et le formalisme proposés par Le Boudec. Nous améliorons aussi en partie la modélisation de certains éléments du réseau, en les décrivant plus finement. Enfin, nous ajoutons à notre méthode d'analyse de nouvelles fonctionnalités, comme le calcul des tailles de files d'attente en nombre de bits ou en nombre de trames, ou bien la prise en compte des flux de niveau de priorité différents. Conformément à nos attentes, les bornes calculées par cette nouvelle version de la méthode sont bien meilleures, et il ressort de notre analyse que notre réseau de test offre bien les garanties recherchées.

Nous avons toutefois pu remarquer que si le réseau de test est assez bien dimensionné au regard du trafic actuel, les marges sont faibles. En aéronautique plus qu'ailleurs, les marges doivent être conséquentes, d'une part pour des raisons de sécurité évidentes, et surtout compte tenu de la longue durée de vie des appareils, amenés à subir des évolutions (nouveaux équipements, nouvelles fonctions) qui engendreront une augmentation du trafic, que le réseau devra être capable d'absorber. Disposant dorénavant d'une méthode d'analyse de réseau avionique, nous avons voulu voir quelles modifications apporter au réseau, susceptibles de créer des marges supplémentaires (**Chapitre 4**). A court terme, la modification la moins coûteuse que nous avons envisagée consiste à utiliser la possibilité offerte par le réseau de jouer sur les priorités des flux. Nous montrons la complexité du problème du choix de ces priorités, ainsi que les gains qui en découlent. Parmi les méthodes proposées se trouvent notamment des heuristiques basées sur notre connaissance du réseau, ainsi qu'une méthode utilisant un algorithme génétique multicritère, SPEA2, couplé à notre outil d'analyse. Nous avons également étudié les améliorations que pourraient apporter une modification de la façon de raccorder les commutateurs entre eux, ou bien la modification de la politique de service des ports de sortie de ces commutateurs. Nous montrons que l'ensemble de ces méthodes forme un panel de techniques assez large pour permettre de dégager les marges recherchées.

Enfin, nous avons voulu mieux comprendre le fonctionnement du réseau étudié. En effet, l'analyse par notre outil fournit des bornes, c'est-à-dire des informations sur les valeurs maximales (délai ou taille de file d'attente) atteintes lors de cas pires. Or, l'intuition ainsi que des simulations effectuées par notre partenaire industriel nous font penser que les probabilités d'occurrence de ces cas pires sont très faibles. Nous avons alors pensé à développer en complément de notre analyse une approche plus stochastique, permettant de déterminer des valeurs qui n'auraient que de très faibles probabilités d'être dépassées (**Chapitre 5**). Pour ce faire, nous avons choisi d'utiliser des résultats présentés par Vojnović et Le Boudec. Nous justifions pourquoi ces résultats, développés dans le contexte de DiffServ, peuvent s'appliquer à notre contexte, puis nous présentons une méthode permettant d'appliquer ces résultats, en complément de notre outil d'analyse déterministe.

En résumé, nous présentons une méthode permettant d'analyser un réseau avionique en vue de sa certification. Cette méthode utilise les résultats du Network Calculus, et une modélisation exacte du réseau étudié ; elle offre l'avantage de donner des informations relatives aux valeurs maximales obtenues, lors de cas pires. Nous étudions également l'impact de certaines modifications du réseau vis-à-vis des résultats donnés par notre analyse, en montrant que ces modifications permettent d'améliorer sensiblement les bornes calculées. Enfin, nous complétons notre analyse par une approche plus statistique, qui peut être utilisée

dans un premier temps pour effectuer des tests matériels, avant peut être de servir de point de départ pour une nouvelle méthode permettant de certifier un réseau, avec des probabilités de panne compatibles avec les critères avioniques.

# CHAPITRE 1

## Le réseau avionique étudié

Le cadre général de cette thèse est celui des systèmes avioniques. Nous présenterons brièvement ces systèmes et leurs spécificités, avant de retracer l'évolution de leur architecture au cours de l'histoire de l'aviation moderne. Nous verrons que cette évolution a entraîné également une évolution de la technologie des moyens de communication. Cette dernière a été marquée par deux grandes tendances, qui ont convergé ces dernières années. La première tendance consiste à remplacer les bus avioniques mono-émetteurs classiquement utilisés (ARINC 429), par des bus multi-émetteurs (ARINC 629 ou MIL-STD 1553B dans le domaine militaire), afin de répondre à un besoin d'interconnexion de plus en plus important. Pour les aéronefs civils, qui sont ceux auxquels nous nous sommes principalement intéressés, la seconde tendance a consisté à réutiliser du matériel développé pour des réseaux locaux classiques, principalement pour des raisons économiques. Les standards ARINC 636 : Onboard Local Area Network et ARINC 646 : ELAN Ethernet Local Area Network constituent de bons exemples de standards visant à modifier des technologies existantes afin de les rendre avionables. Pour l'A380, Airbus a choisi une technologie Ethernet commuté, qui représente la conjonction de ces deux tendances. Nous présenterons cette technologie, ainsi que les principales raisons pour lesquelles elle a été choisie. Nous détaillerons également la norme en cours de préparation (ARINC 664), qui prévoit les modifications liées au contexte embarqué.

Par la suite, nous présenterons globalement le système avionique que nous avons particulièrement étudié, c'est-à-dire le système ADCN (Aircraft Data Communication Network), reposant sur la technologie réseau AFDX (Avionics Full Duplex). Ce réseau est celui qu'utilise l'Airbus A380, et s'inscrit dans le cadre des réseaux déterministes décrits par la norme ARINC 664.

Dans un dernier paragraphe, nous montrerons en quoi l'utilisation de ce nouveau type de réseau entraîne de nouveaux problèmes vis-à-vis du contexte avionique. Nous en déduirons qu'il est impératif de mettre en place une méthode permettant de certifier que le réseau apporte bien les garanties souhaitées en termes de qualité de service.

## 1 Architecture des systèmes avioniques

On appelle avionique d'un aéronef l'ensemble des équipements et systèmes informatiques ou électroniques embarqués à bord de cet engin. Cela comprend entre autres des calculateurs, le logiciel qu'ils exécutent, des capteurs et actionneurs, et les moyens de communication entre ces différents éléments. De nos jours, de très nombreuses fonctions sont confiées à ce système avionique. A titre d'exemple, on peut citer les fonctions de commande de vol, de pilotage automatique, de navigation, d'affichage des informations, ou bien des fonctions dites de "confort" comme le retour d'effort dans les commandes de pilotage. Une étude approfondie de ces systèmes peut être consultée dans l'ouvrage [1].

### 1.1 Evolution des systèmes avioniques

Les premiers pas de l'avionique correspondent au remplacement progressif d'équipements analogiques par leurs homologues numériques, dans les années 60. Par la suite, l'avionique a bénéficié des progrès de l'électronique puis de l'informatique, les premiers micro-ordinateurs

ayant été installés à bord en 1970 pour exécuter des fonctions liées à la navigation. Depuis, deux phénomènes poussent à la complexification du système avionique: le remplacement progressif des équipements analogiques d'une part, et l'adjonction de nouvelles fonctions d'autre part. Ces fonctions résultent de progrès dans les domaines de l'électronique, de l'informatique et des télécommunications, et répondent à des besoins nouveaux ou apportent de meilleures solutions à des besoins déjà présents. A titre d'exemple, on peut citer le système de commandes de vol, qui est passé progressivement d'un contrôle par câbles mécaniques jusqu'au système actuel, où un ordinateur gère les différentes servocommandes tout en respectant un domaine de vol précis.

On peut constater de manière simple l'augmentation de la complexité du système avionique: de 1983 (A310) à 1993 (A340), le nombre d'équipements embarqués a augmenté de 50% (de 77 à 115), alors que la puissance embarquée a été multipliée par 4, passant de 60 Mips à 250 Mips. De manière générale, à l'instar de la loi de Moore pour la puissance des processeurs, on a établi [4] que la complexité des systèmes avioniques double tous les 5 ans.

Aujourd'hui, le système avionique correspond à 30% du coût total d'un appareil civil, et jusqu'à 50% d'un appareil militaire [5].

## **1.2 Classification des systèmes avioniques**

Bien qu'un système avionique se rapproche d'un système informatique distribué, il est soumis à un jeu de contraintes plus important. Par exemple, les contraintes de volume et surtout de poids sont importantes pour des raisons évidentes de coût. Les systèmes embarqués doivent aussi fonctionner de manière nominale dans des conditions difficiles (chaleur, vibrations, environnement électromagnétique, etc.).

Une autre différence avec un système informatique classique, qui fait véritablement la spécificité d'un système avionique, est la contrainte de sûreté de fonctionnement. En effet, en cours de mission, la moindre défaillance met en danger la vie des occupants de l'appareil. Les normes DO-178B [6] et ARP 4754 [9] classifient les différents systèmes ou équipements en trois catégories: les systèmes critiques, essentiels, et non essentiels.

- Les systèmes critiques: Un système est critique si une défaillance éventuelle implique des conséquences catastrophiques, à savoir la perte de l'appareil et/ou de vies humaines. Le système de commandes de vol électriques en est un exemple frappant, car en cas de défaillance l'appareil n'est plus gouvernable. De tels systèmes ont de manière évidente des contraintes de sécurité très élevées, et obéissent généralement à des contraintes temps réel très strictes. Ceci implique souvent que ces systèmes sont de taille et de complexité limitée, avec des redondances multiples et une ségrégation spatiale. Pour des raisons de certification, ces systèmes doivent être déterministes, à savoir qu'on doit pouvoir prévoir leur comportement. Pratiquement, on implémente souvent les fonctions de ces systèmes avec des traitements périodiques de faible période, de l'ordre de 10 ms.
- Les systèmes essentiels: Un système est essentiel si une défaillance de sa part compromet la réussite de la mission en cours, sans pour autant mettre en danger de vie humaine. Sur un bimoteur, un des moteurs peut être considéré comme essentiel puisque l'avion peut toujours être piloté en cas de panne, mais cette panne oblige à un atterrissage prématuré sur l'aéroport le plus proche. A l'instar des systèmes critiques, ces systèmes obéissent aussi à des contraintes temps réel, et possèdent aussi la nécessité d'être déterministes. Leur logiciel est généralement de grande taille mais de complexité moyenne, et ils sont implémentés par des traitements cycliques dont la période est de l'ordre de 100ms.



- Les systèmes non essentiels: Ce sont les systèmes qui ne rentrent pas dans les catégories précédentes. Ces systèmes sont appelés à former une part de plus en plus importante du système avionique.

### **1.3 Architectures classiques**

Dans ce chapitre, nous montrons l'évolution des architectures employées pour les systèmes avioniques, depuis l'origine de l'aviation moderne jusqu'aux systèmes actuels, tels que ceux embarqués dans l'A380.

#### **1.3.1 Architecture centralisée**

Au commencement de l'avionique moderne, les rares fonctions embarquées étaient gérées par un calculateur unique, central. De nos jours, la taille des avions, le nombre de fonctions et leur complexité, ainsi que d'élémentaires raisons de sécurité (trop grande vulnérabilité) rendent cette architecture impropre à l'utilisation dans les aéronefs civils.

#### **1.3.2 Architecture fédérale**

Ce type d'architecture a été pour la première fois présenté dans le concept DAIS (Digital Avionics Information System) et a été principalement utilisé dans des appareils militaires, comme le F16 de l'US Air Force. Ces architectures sont composées de nombreux équipements informatiques qui implémentent les différentes fonctions avioniques, et sont répartis dans la totalité de l'appareil, à proximité des différents capteurs et actionneurs. Ceci présente l'avantage de diminuer le temps de communication entre les capteurs et les calculateurs, et aussi de diminuer la connectique nécessaire. Cependant, si le traitement des données est réparti, le contrôle des équipements est centralisé dans des calculateurs de moyenne puissance. Chacun de ces calculateurs contrôle l'ensemble des équipements d'un domaine donné, par exemple le système d'armes, et la communication entre les éléments est aussi contrôlée par le calculateur maître. Ceci explique l'utilisation massive de bus de type MIL-STD-1553B [18], pour lesquels le temps de parole est successivement (et explicitement) distribué aux équipements par le contrôleur du bus, ici le calculateur maître.

Cette architecture présente de nombreux avantages: la répartition des systèmes diminue la vulnérabilité, et la relative simplicité de l'ensemble permet une analyse précise du fonctionnement du système. Cependant, cette architecture suppose une grande intégration de l'ensemble du système pour pouvoir être entièrement contrôlé par quelques calculateurs. Ceci n'est pas le cas pour des appareils civils, où il est important pour les fabricants de faire jouer la concurrence entre différents équipementiers. Ce type d'architecture n'est donc pas commercialement adapté aux aéronefs civils.

#### **1.3.3 Architecture répartie**

Historiquement, le remplacement dans les aéronefs civils des équipements analogiques par leur équivalent numérique a conduit aux architectures réparties. Dans ce type d'architecture, à l'instar des architectures fédérales, les fonctions sont exécutées par différents équipements répartis dans l'appareil; la différence est qu'ici le contrôle des équipements n'est pas centralisé et les systèmes coopèrent pour l'accomplissement de la mission. Ils s'échangent les données au moyen de bus mono-émetteur [10] ou multiplexés [11]. En effet, contrairement à l'architecture fédérale, la conception du système réparti est indépendante du moyen de communication choisi. Ainsi, l'avionique de l'A320 et de l'A340 repose exclusivement sur de l'ARINC 429, alors que l'avionique du Boeing 777 utilise à la fois les deux types de bus, 429 et 629.

Dans ce type d'architecture, chacun des équipements accomplit une fonction du système. Du point de vue matériel, on a décidé de standardiser la forme de ces équipements, qui prennent

la forme de modules standards, appelés LRU (Line Replaceable Units). Ces modules regroupent des ressources de mémoire et de capacité de calcul, et offrent une interface standard aux applications qui vont exécuter les fonctions. Cette standardisation permet de faciliter la maintenance puisqu'il suffit, en cas de défaillance d'un équipement, de remplacer le LRU et de charger sur le nouveau module le logiciel correspondant à la fonction recherchée. Le principal inconvénient de cette architecture répartie est qu'elle implique une mauvaise utilisation des ressources : les ressources d'un module ne peuvent être partagés par plusieurs fonctions.

#### **1.4 Avionique modulaire intégrée**

L'IMA (Integrated Modular Avionics) est une architecture qui reprend les principes précédents en ce qui concerne la répartition des équipements, mais permet de remédier à l'inconvénient des architectures réparties. Le principe de cette architecture est de répartir dans tout le volume de l'appareil un ensemble de ressources (calcul, mémoire, communication), qui seront partagés par plusieurs applications, qui accompliront les différentes fonctions du système avionique.

De manière pratique, l'architecture physique est décrite par la norme ARINC 651 [12]. Les ressources sont regroupées dans des modules génériques appelés LRM (Line Replaceable Module), qui sont à leur tour regroupés dans des étagères, la communication au sein de ces étagères étant réalisée avec des bus spéciaux, généralement du type ARINC 659 [19]. Les modules peuvent être de trois types:

- Les modules cœurs sont ceux qui se chargent de l'exécution des applications.
- Les modules d'entrée/sortie permettent la communication avec des éléments ne respectant pas l'architecture IMA.
- Les modules passerelles servent à la communication entre étagères.

L'architecture logicielle est décrite par la norme APEX [13], qui permet d'offrir aux applications une interface générique vers le système d'exploitation. Le développement du logiciel est ainsi rendu indépendant du matériel sur lequel il sera exécuté, ce qui permet d'envisager des réductions de coût lors de la phase de développement logiciel.

Cette norme ARINC 653 décrit deux modes de communication sur les ports : « sampling » ou « queuing ». En mode sampling, dans un abonné consommateur d'une donnée, les couches basses du réseau ne présentent aux applications que la dernière valeur de la donnée, la plus « fraîche ». Dès qu'une nouvelle valeur de la donnée est reçue, elle écrase l'ancienne. Ce mode est particulièrement approprié pour des applications qui ont besoin de recevoir des données périodiquement. En mode queuing, les valeurs ne sont pas effacées systématiquement ; au contraire, elles sont stockées et présentées dans l'ordre de leur réception, jusqu'à ce l'application concernée ait eu le temps de les lire. Ce mode est plus approprié pour des transferts de données aperiodiques, pour lesquels il est nécessaire que toutes les données soient lues.

Les premiers exemples d'implémentation de l'IMA (AIMS – Airplane Information Management System sur le Boeing 777) laissent envisager de nombreux gains pour tous les acteurs de l'aéronautique:

- Adaptabilité: grâce à la modularité de l'architecture, il est possible de reconfigurer un appareil pour l'adapter à la mission envisagée.
- Maintenabilité: La standardisation des cartes permet un niveau de maintenance très élevé.
- Réduction des coûts: aussi bien dans l'acquisition du matériel, que dans sa maintenance et son évolution.

- Gain en masse et en volume: diminution du nombre d'équipements, et donc simplification du réseau de communication.

Compte tenu de ces gains envisagés, Airbus a tout naturellement choisi cette architecture pour le système avionique de l'A380. Dans le cadre du projet européen de recherche VICTORIA [15], Airbus et plusieurs grands partenaires ont proposé une architecture avionique qui englobe tous les systèmes du réseau, depuis les systèmes de contrôle de l'appareil jusqu'aux systèmes destinés aux occupations des passagers. Le réseau avionique global de l'A380, est un exemple de ce type d'architecture, intitulée IME (Integrated Modular Electronics). Au sein de cette architecture IME, les systèmes essentiels seront regroupés dans un réseau protégé du « monde ouvert », et qui possèdera une architecture de type IMA. Comme cette architecture IMA n'impose pas de moyen de communication spécifique entre ses différents composants, Airbus a choisi pour l'A380 d'utiliser la technologie de l'Ethernet Commuté Full-Duplex, conformément aux résultats du projet européen PAMELA [16]. Dans les paragraphes suivants, nous donnerons quelques raisons permettant de justifier ce choix, et nous présenterons cette technologie.

## 2 Ethernet commuté Full Duplex

Pourquoi choisir l'Ethernet commuté plutôt que des bus avioniques comme l'ARINC 429, ou l'ARINC 629 ?

Le premier avantage de cette architecture est le débit offert aux stations. L'Ethernet offre classiquement comme débit 10 ou 100 Mbit/s, et le gigabit/s est d'ores et déjà disponible. Les bus avioniques classiques offrent eux un débit de l'ordre de quelques Mbit/s. Un autre avantage de l'Ethernet est qu'il s'agit d'un standard IEEE (donc non propriétaire) bien connu. Ceci implique notamment des coûts d'utilisation réduits, car les investissements peuvent être amortis sur de plus grands volumes. Une technologie mûre présente aussi de nombreux avantages, notamment celui de présenter moins de surprises lors de sa mise en pratique. La connaissance générale de la technologie Ethernet permet également de réutiliser des outils de maintenance déjà développés pour d'autres applications. Enfin, le fait que ce sont les commutateurs qui réalisent les duplications des trames permet de diminuer le nombre de câbles embarqués dans l'aéronef, par rapport à des bus mono émetteurs. Le gain pour les compagnies est très important, car on sait que le fait de rajouter un kilo d'équipement dans un avion augmente sa masse au décollage d'environ 10 kilos.

### 2.1 La méthode d'accès Ethernet

La norme IEEE 802.3 [22] spécifie ce qu'on appelle communément un réseau Ethernet, plus précisément des réseaux dont le contrôle d'accès au medium physique (MAC : Medium Access Control) utilise l'algorithme du CSMA/CD. CSMA/CD est le moyen par lequel deux (ou plus) stations partagent le même support physique. Pour transmettre, une station attend une période d'inactivité sur medium physique (aucune autre station ne transmet), puis transmet ses données. Si le message entre en collision avec celui d'une autre station, chacune de ces stations émet pendant un temps prédéfini pour s'assurer de la propagation de la collision dans tout le système. Ensuite, les stations attendent un temps aléatoire avant de réessayer de transmettre leur message.

Le principal inconvénient de l'Ethernet est l'indéterminisme du temps d'accès au medium physique, qui est lié directement à cet algorithme. Pour diminuer ce risque de collisions, on adopte fréquemment une approche Ethernet commuté.

## 2.2 Réseau Ethernet commuté

Dans un réseau Ethernet commuté, la configuration qui est adoptée est celle d'une topologie en étoile, avec un commutateur central dont les ports sont reliés à un et un seul équipement, ce qui constitue un LAN différent pour chaque port. L'interconnexion avec les autres éléments est ensuite réalisée en reliant les commutateurs entre eux. Cette topologie a l'avantage de réduire par segmentation les domaines de collision au seul lien point à point entre un équipement et son commutateur, ou bien entre deux commutateurs.

Dans un réseau Ethernet commuté, l'équipement le plus important est bien sûr le commutateur. La norme qui définit ces équipements est le 802.1D [20]. On y apprend qu'un commutateur doit remplir trois types de fonctions :

- le relais et le filtrage des trames,
- la mise à jour des informations permettant de remplir le rôle précédent
- une surveillance de son fonctionnement interne

La figure suivante, extraite de la norme, présente l'architecture générale des différentes entités du commutateur qui interviennent lors du relais d'une trame :

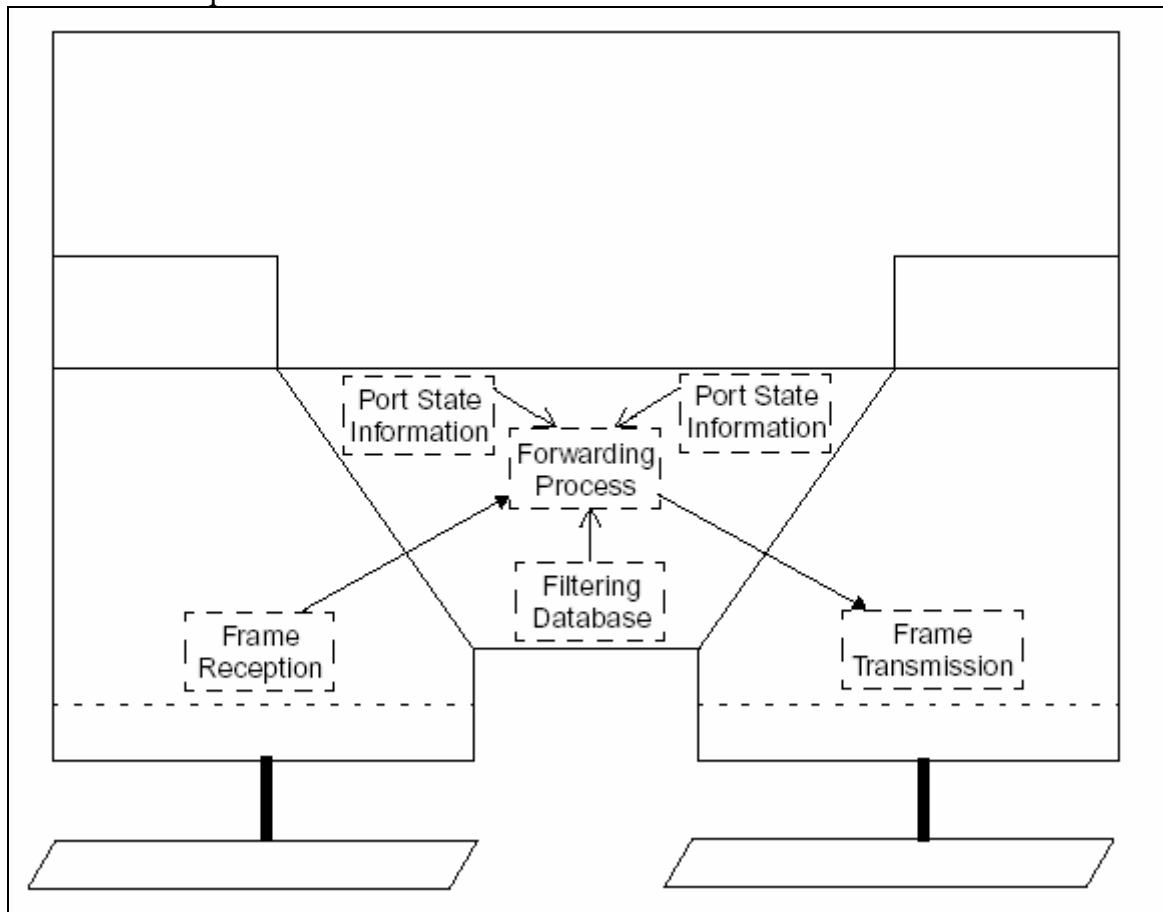


Figure 1 Relais d'une trame

Voici la liste des opérations que doit accomplir le commutateur lors de la réception d'une trame :

- Réception de la trame.
- Destruction des trames corrompues, ou n'étant pas du bon type (user\_data).
- Eventuellement, régénération de la priorité de la trame.
- Filtrage de la trame suivant les informations de filtrage, destruction si nécessaire.
- Aiguillage vers les autres ports du commutateur.
- Choix de la classe de la trame, puis mise dans la file d'attente correspondante.

- Destruction des trames ayant attendu trop longtemps.
- Sélection de la trame à transmettre.
- Emission de la trame.

Les informations de filtrage peuvent être acquises soit statiquement, soit de manière dynamique : à la réception de chaque trame, le commutateur « apprend » l'adresse MAC de l'émetteur de la trame. Ces informations peuvent aussi être obtenues par l'utilisation de l'algorithme GMRP (GARP Multicast Registration Protocol, où GARP signifie Generic Attribute Registration Protocol [21]). Bien entendu, une des entités du commutateur a pour fonction de vérifier l'obsolescence de ces informations, et de gérer le protocole GMRP.

Les ports du commutateur ne sont pas forcément tous en mesure de retransmettre la trame ; il se peut que l'utilisation de l'algorithme de spanning tree [20] conduise à les désactiver. C'est pour cela que l'opération aiguillage de la trame nécessite des informations sur l'état des ports.

### **2.3 Réseau Ethernet commuté Full-Duplex**

Nous avons vu que dans une architecture Ethernet commuté, les seules collisions possibles se situent au niveau des liens point à point. Pour pouvoir embarquer ce type de technologie, il faut éviter de telles collisions. La solution qui a été développée consiste à utiliser des liens bidirectionnels, qui opèrent selon le mode Full Duplex prévu dans la norme IEEE 802.3. Le mode Full Duplex permet une communication simultanée entre deux stations, reliées en point à point par un medium physique dédié. Ce mode d'opération ne nécessite pas de retarder l'émission d'un message, ni même d'écouter ou de réagir à l'activité sur le medium physique, puisqu'il n'y a pas de collision possible sur ce medium. Ceci implique donc que l'utilisation de l'algorithme CSMA/CD n'est plus nécessaire.

On appelle couramment ce type de réseau un réseau Ethernet commuté Full Duplex. Ces réseaux possèdent l'avantage de ne plus posséder d'indéterminisme quant au temps d'accès au support physique, et de ne pas entraîner de pertes de trames par collision.

Par rapport aux réseaux Ethernet classiques, l'architecture Ethernet commuté Full Duplex permet également d'utiliser des liens plus longs, comme indiqué dans le paragraphe 29.4 de la norme 802.3. En effet, il n'est plus nécessaire d'écouter l'activité sur le lien physique, donc la limitation sur la longueur de celui-ci n'a plus lieu d'être. D'autre part, la topologie en étoile permet d'obtenir de meilleures performances en terme de débit pour chacune des stations connectées, puisqu'il n'y a plus de perte de bande passante due aux collisions de trames.

Il ne faut pas croire cependant que la suppression de CSMA/CD rend toute l'architecture déterministe. Le problème est en quelque sorte décalé dans les commutateurs, où les différents flux vont entrer en compétition pour l'utilisation des ressources du commutateur. Si trop de trafic se dirige à un instant donné vers un seul port de sortie, une congestion de celui-ci est possible, ce qui implique de nombreuses pertes de trames. On pourrait également voir apparaître des rafales de trafic, dues à un encombrement passager d'un port du commutateur. Ces rafales risquent à leur tour d'encombrer le commutateur suivant, et on risque ainsi de trouver des commutateurs de cœur de réseau congestionnés, ce qui entraînerait également des pertes de trames. Comme nous le verrons, un des buts de notre étude est de comprendre si de telles congestions peuvent ou non intervenir dans le réseau étudié.

C'est au vu des avantages apportés par cette technologie Ethernet commuté Full-Duplex, qu'Airbus a choisi de l'utiliser en tant que moyen de communication pour son architecture IMA. Cependant, l'utilisation dans un contexte embarqué d'une technologie développée pour un autre contexte moins contraignant nécessite un certain « durcissement », c'est-à-dire une adaptation aux exigences aéronautiques. Nous présentons ci-dessous une norme qui est développée dans ce sens.

## 3 La norme ARINC 664

### 3.1 *Domaine d'application*

La norme ARINC 664 est actuellement en cours de rédaction, avec le soutien de nombreux industriels comme Airbus, Boeing, Rockwell-Collins ou Honeywell. Le but de cette norme est de proposer un standard pour les réseaux embarqués à bord d'aéronefs civils, en adaptant des standards industriels existants dans d'autres domaines. Elle décrit l'architecture globale du système d'information embarqué dans un aéronef, et montre notamment comment intégrer des réseaux supportant des applications critiques (liées au contrôle de l'appareil), d'autres réseaux supportant des applications essentielles (gestion de la cabine et des passagers) et des réseaux destinés au confort et à l'occupation des passagers. Le principal avantage de cette démarche est de permettre de fournir à toute l'industrie aéronautique un standard permettant l'interopérabilité à l'intérieur et entre plusieurs réseaux embarqués. En adaptant des standards bien connus, la norme ARINC permet alors à toute la communauté aéronautique des gains économiques liés à la réutilisation, sans pour autant compromettre les critères de performance très stricts de certaines applications avioniques. Parmi les standards qui seront adaptés, on retrouve principalement l'IEEE 802.3 [22], 802.1D[20], et IP [23].

La norme définit plusieurs sortes de réseaux: les réseaux conformes ("compliant networks") qui satisfont aux normes standard, et les réseaux profilés ("profiled networks"). Les réseaux profilés ne dérogent aux normes classiques que dans la mesure où une modification ou des restrictions supplémentaires sont nécessaires pour l'utilisation dans le contexte avionique. Les réseaux déterministes sont une sous-catégorie des réseaux profilés, qui permettent d'offrir des garanties temps réel aux applications.

La norme suggère que dans le système avionique, les applications devront être regroupées en domaines séparés, avec un domaine par niveau de criticité des applications. On trouvera donc les applications critiques et essentielles regroupées au sein de domaines possédant un réseau profilé ou déterministe, alors que les applications non essentielles seront dans un ou plusieurs domaines possédant un réseau conforme. De cette façon, ces applications pourront réutiliser des logiciels du commerce, qui n'auront pas à être certifiés au sens du DO-178B [6].

L'architecture générale de la norme est la suivante :

- partie 1 : Concepts et vue générale de la norme
- partie 2 : Spécifications pour l'implémentation de l'IEEE 802.3
- partie 3 : Caractéristiques modifiées de IP, TCP, UDP
- partie 4 : Gestion des adresses dans le réseau
- partie 5 : Définition des caractéristiques et des services offerts par les différents domaines
- partie 6 : Définition des protocoles de gestion de réseau
- partie 7 : Réseaux déterministes
- partie 8 : Exemples de services offerts par les couches hautes.

Les travaux réalisés dans le cadre de cette thèse portent principalement sur la partie déterministe du réseau, aussi nous détaillons dans le chapitre suivant les principales modifications exigées par la norme pour les réseaux déterministes.

### 3.2 *Restrictions imposées pour les réseaux profilés*

On peut classer les restrictions imposées en trois catégories, dont la dernière ne concerne que les réseaux déterministes.

### 3.2.1 Configuration statique

La configuration du réseau doit être statique, et entièrement connue avant le décollage. De cette façon, on évite tous les problèmes d'initialisation de réseau, ainsi que l'indéterminisme lié au temps de recherche des adresses, des routes, etc. Ceci implique entre autres que les tables de commutation (correspondance adresse MAC de destination / port(s) de sortie) doivent être configurées statiquement par l'intégrateur du système. La norme suppose en effet que ces réseaux utiliseront la technologie Ethernet commuté Full Duplex, et que les équipements transmettront des flux de données vers plusieurs destinations (caractère multicast). La commutation sera donc effectuée au niveau 2, sur la base d'adresses de groupe des destinations.

De plus, des algorithmes comme l'ARP, le GMRP ou le Spanning-Tree ne sont pas nécessaires dans ce contexte et doivent donc être désactivés.

### 3.2.2 Isolation des erreurs

De manière générale, le réseau doit assurer un confinement des erreurs, c'est-à-dire qu'une erreur locale ne doit pas se propager et détériorer le comportement d'autres éléments. Ceci implique par exemple un filtrage des trames dans les éléments chargés de les relayer (commutateurs notamment). Ce filtrage permet de supprimer les trames de longueur incorrecte, ou corrompues (vérification du champ FCS), ou dont la source ou la destination ne sont pas identifiées. Une des conséquences de ce filtrage est que les éléments formant le réseau ne doivent pas fonctionner en mode « cut through » mais bien en mode « store and forward », car les trames ne doivent être relayées que si elles sont valides. Une autre conséquence est que le mode de communication « broadcast » est interdit pour tous les équipements.

Le réseau doit également isoler les flux de données les uns des autres : cette exigence oblige à insérer dans chaque trame un champ qui identifie de manière unique le flux auquel elle appartient. Cette identification permet de réaliser des opérations de contrôle de la bande passante, de vérification de l'ordre des trames, ou de détecter des usurpations d'identité. Si des commutateurs sont utilisés dans le réseau, la norme précise que le phénomène de « head of line blocking » ne doit pas intervenir : si une trame doit être commutée vers plusieurs ports, et que l'un d'entre eux est bloqué, cela ne doit pas empêcher la commutation vers les autres, ce qui renforce l'isolation des flux.

La norme précise enfin que le réseau doit implémenter un protocole de surveillance du réseau, qui permet de garder la trace de tous les événements, comme les pertes de trames dans les commutateurs.

### 3.2.3 Trafic maîtrisé

La principale caractéristique des réseaux déterministes au sens de la norme ARINC 664 est que ce sont des réseaux qui garantissent une certaine qualité de service, qui doit être démontrée (cette qualité de service dépend des applications et est déterminée par l'intégrateur du système). Or une telle preuve n'est possible que si le trafic entrant dans le réseau est connu, et maîtrisé. Cette exigence est ce qui fait vraiment la spécificité des réseaux déterministes : chaque source doit passer un « contrat de trafic » avec le réseau, et s'engager à le respecter ; de son côté le réseau doit mettre en place des dispositifs visant à surveiller que les sources ne dépassent pas leur quota d'émission. Ainsi, les émetteurs de données doivent utiliser des régulateurs de trafic, alors que les commutateurs vont posséder des « policiers » c'est-à-dire des éléments qui rejeteront les trames en excès. Il existe de nombreuses façons de formaliser ce « contrat de trafic », mais la norme cite principalement le concept de Virtual Link, utilisé par Airbus, et que nous décrivons plus précisément dans la partie suivante, qui est consacrée à la présentation du réseau ADCN d'Airbus (Aircraft Data Communication

Network) que nous avons étudié, et qui englobe l'ensemble du système d'information de l'appareil.

## **4 Le réseau AFDX**

### **4.1 Les objectifs du réseau**

Les réseaux que nous avons étudiés sont pour la plupart des prototypes du réseau ADCN, qui a été développé pour l'A380. Ce réseau utilise dans ses couches basses la technologie réseau AFDX, qui est dérivée de l'Ethernet commuté. Le réseau dans son ensemble est conforme à la norme ARINC 664. Le domaine de sécurité le plus élevé est même cité comme exemple dans la partie 7 de cette norme, qui a trait aux réseaux déterministes. Dans la suite de cette thèse, nous ne présenterons que ce réseau déterministe, car c'est celui pour lesquels se posent les vrais problèmes liés à sa criticité.

Ce réseau est composé principalement de commutateurs, qui sont les éléments clés de l'architecture, et de producteurs/consommateurs de données, qui sont appelés les abonnés. Au sein de chaque abonné, l'équipement qui sert à se connecter au réseau, et qui gère la communication avec les autres équipements du réseau, se nomme le End System.

Le réseau a pour objectifs de:

- fournir des données fiables, ce qui est accompli notamment grâce à l'utilisation de FCS (Frame Check Sequence) Ethernet.
- offrir une disponibilité élevée. Sachant que de nombreux équipements seront reliés au réseau, la disponibilité du réseau aura une incidence directe sur celle de l'aéronef complet. Dans cette optique, on prévoit pour les commutateurs des MTBF (Mean Time Between Failure) très élevés, de l'ordre de 100000 heures.
- garantir l'ordonnancement des paquets au sein d'un même flux, ainsi qu'un chemin unique de bout en bout pour tous ces paquets.
- Fournir une isolation des erreurs, entre éléments consécutifs ou entre flux.
- fournir un service déterministe de transfert de donnée. La latence maximale de bout en bout, ainsi que la gigue, doivent être connues ; aucune perte de trame par congestion n'est tolérée.

Il faut souligner l'importance de ce dernier objectif, qui est directement relié à l'utilisation de moyens de communications partagés. En effet, dans le cas d'un réseau utilisant des bus avioniques dédiés, cet objectif est directement vérifié : les seules pertes sont dues à des erreurs physiques sur le bus, et le temps de transmission d'une donnée n'est influencé que par le débit du bus. Par contre, dans le cas d'un réseau Ethernet commuté Full-Duplex, le risque de pertes de trames est bien présent. L'utilisation de liens Full-Duplex supprime le risque de collisions sur le lien physique ; par contre il subsiste les pertes consécutives aux erreurs physiques, mais surtout les pertes liées aux congestions du commutateur. En effet, les files d'attente internes au commutateur sont d'une taille limitée, et des trafics confluent pourrait les faire déborder. De la même façon, le stockage de trames dans les files d'attente des commutateurs peut entraîner des délais (et des giges) importants. Ce dernier objectif n'est donc pas trivialement atteint, et touche de manière importante au processus de certification du système avionique, que nous présentons dans le paragraphe 6.

### **4.2 La notion de VL**

Pour répondre à ces objectifs, et conformément à l'ARINC 664, il est nécessaire de contrôler l'ensemble des flux qui pénètrent dans le réseau. Airbus a donc développé dans ce sens la notion de Virtual Link (VL). Cette notion est décrite dans la partie 7 de la norme ARINC 664 comme un exemple possible de représentation du trafic. L'idée principale derrière cette notion



est la volonté de fournir une ségrégation des flux, ce qui est imposé par la norme pour raison de sécurité : le mauvais comportement d'un flux ne doit pas nuire aux autres flux. En fait, la notion de VL va permettre de « virtualiser » un bus avionique classique pour chaque flux, où il serait le seul flux à émettre. On va donc essayer de garantir à chaque flux une certaine bande passante, ainsi que d'autres caractéristiques de qualité de service. L'avantage est double, puisqu'à part le gain recherché du point de vue de la sécurité, ce concept permet aussi de par sa similitude avec un bus ARINC 429, de conserver l'existant en de nombreux équipements. Le temps de transition entre les deux technologies s'en trouve alors fortement diminué.

Plus précisément, un VL est défini par :

- un identifiant unique,
- une ou plusieurs adresses de destination,
- la route empruntée pour rallier ces destinations,
- la taille maximale et minimale d'un paquet (en bits, notées  $s_{\max}$  et  $s_{\min}$ ),
- un temps minimum d'émission entre deux trames consécutives, appelé BAG (Bandwidth Allocation Gap).

On voit immédiatement que ces dernières données permettent de définir la bande passante maximale du lien : il ne peut émettre au maximum qu'un paquet de taille maximale toutes les BAG. Son débit maximal en bits par seconde, noté  $\rho$ , est donc :

$$\rho = \frac{s_{\max}}{BAG}$$

### 4.3 Services de communication

## 5 La problématique de certification du réseau

Nous avons vu dans les paragraphes précédents que l'évolution de la complexité des systèmes avioniques a conduit les avionneurs à utiliser de nouvelles architectures, et au sein de ces architectures de nouveaux moyens de communication. Cependant, la technologie utilisée pour ces moyens de communication ne comporte pas de mécanismes internes permettant d'assurer que le réseau offrira bien la qualité de service requise, qui comprend entre autres une latence et une gigue bornées, ainsi que l'absence de perte de trames par congestion. En effet, les commutateurs prescrits par la norme ARINC 664 sont conformes à la norme IEEE 802.1D, avec laquelle il est possible de perdre des trames. Dès lors, il est important de prouver que le réseau est capable d'offrir la qualité de service requise.

Ceci est d'autant plus vrai que le système avionique doit passer l'épreuve de la certification. Dans ce processus de certification, l'avionneur qui est responsable de tout l'appareil doit fournir aux autorités de certification, la JAA en Europe, la preuve que chaque fonction sera remplie conformément aux exigences exprimées. Le système avionique est bien entendu de niveau critique, et doit donc être certifié suivant les critères les plus stricts. La complexité de l'ensemble du système rend ce processus complexe. En accord avec les autorités de certification, la démarche retenue a été de certifier séparément le réseau, moyen de communication entre les différents équipements. De cette façon, on sépare en deux parties la complexité du problème : d'une part la question d'exécution de tâches temps réel, d'autre part les problématiques de communication. Pour le fonctionnement des équipements et de leurs applications, de multiples méthodes formelles ont été proposées et utilisées [5][25]. Ces méthodes nécessitent cependant une connaissance précise et sûre du fonctionnement du réseau.

On voit ici qu'il est fondamental de développer une méthode permettant de prouver que le réseau offre bien la qualité de service requise. Cette méthode devra prouver que le délai de traversée du réseau est borné pour tous les flux, et qu'il ne peut pas y avoir de trames perdues dans les commutateurs à cause de congestions.

Les travaux présentés dans cette thèse ont comme objectif de répondre à cette problématique, en proposant une méthode capable de prouver que le réseau répond à ces exigences.

Parmi les méthodes envisagées, nous avons fait le choix d'appliquer des résultats issus de la théorie du Network Calculus, que nous présenterons dans le deuxième chapitre de cette thèse. Ce choix se justifie par le fait que cette théorie est mieux adaptée au contexte que d'autres approches, par exemple par simulation ou par des méthodes formelles. Il est en effet difficile de prouver avec certitude des choses par simulation ; on peut en général tout au plus affirmer que les probabilités d'occurrence de problèmes sont faibles. La technologie employée étant récente en aéronautique, la confiance qu'on peut lui accorder est faible. Dès lors, il est logique que les autorités de certification préfèrent à ces garanties de type probabiliste des résultats déterministes, et refusent une approche de simulation pour la certification du réseau. Des méthodes formelles (par exemple basées sur des réseaux de Petri) pourraient apporter ce genre de garantie déterministe. Cependant, compte tenu de la complexité du réseau et de l'importance du trafic qui l'utilise, nous avons jugé qu'une approche par méthode formelle conduirait à un trop grand nombre de calculs d'états.

Le chapitre suivant présente la théorie du Network Calculus que nous avons choisi d'appliquer, et propose une première approche pour répondre à notre problématique. Notre choix s'est porté sur le Network Calculus car c'est une théorie qui a déjà été appliquée avec bonheur dans d'autres contextes comme par exemple Internet avec IntServ (Integrated Services) [7] ou les réseaux locaux industriels [8], qui sont des domaines dont le point commun est que les flux d'entrée dans le réseau y sont régulés.

Pour mieux situer le contexte de nos travaux, nous précisons que le domaine d'étude de cette thèse s'étend depuis les régulateurs de trafic dans les abonnés, jusqu'à réception des trames dans les abonnés destination. Du point de vue réseau, on étudie donc principalement les niveaux 1 et 2, et on prend en compte les trames au moment où elles sont assemblées.

## CHAPITRE 2

### Méthode de preuve de déterminisme par calcul réseau

Les travaux présentés dans ce chapitre sont fondateurs de cette thèse. Il s'agit d'une première réponse à la problématique proposée par Airbus : trouver une méthode permettant de certifier que le réseau de communication envisagé remplira bien les exigences fixées. Pour ce faire, une telle méthode doit permettre de calculer des bornes sur les temps de traversée du réseau, ainsi que sur les giges et les tailles des files d'attente. Nous avons retenu dans ce contexte les travaux fondamentaux de Cruz [26][27], le pionnier du Network Calculus. Dans ce chapitre, nous présenterons ces travaux, puis la modélisation du réseau que nous avons mise en place. Nous montrerons pourquoi les travaux de Cruz peuvent s'appliquer dans ce contexte, et présenterons la méthode de calcul que nous avons employée, reprenant celle décrite par Cruz.

#### 1 Les travaux fondateurs de Cruz

Note : Pour une meilleure homogénéité, nous présenterons cette section en utilisant les notations utilisées dans le reste du document, et non celles de Cruz dans son article.

Le contexte général des travaux de Cruz [26][27] est celui des réseaux de communication, avec commutation des paquets selon un routage statique. Cela signifie que toute donnée qui est émise sur le réseau par un utilisateur doit être sous forme de paquet, et non sous forme de bits individuels. Ces paquets circuleront alors dans le réseau suivant un itinéraire prédéfini, jusqu'à atteindre leur destination. On peut constater que ce contexte se rapproche beaucoup de ce qui peut exister pour des chaînes d'assemblage de produits manufacturés; Cruz fait d'ailleurs remarquer que ses résultats peuvent également s'étendre à ces situations-là (voir par exemple les travaux de Parekh et Gallager [31][32] et de Kumar et Perkins [28][29][30]).

La grande particularité des travaux de Cruz, par rapport aux autres analyses de ce type de réseau, réside dans la description des flux en entrée du réseau : il ne considère pas connu un modèle stochastique du comportement des flux, mais juste le fait que ces flux doivent satisfaire certaines contraintes de régularité. Par la suite, son analyse consiste à comprendre à quelles contraintes les flux satisfont, à l'intérieur du réseau cette fois.

Dans les paragraphes suivants, nous définirons précisément ces contraintes de régularité, puis nous présenterons la modélisation proposée par Cruz pour différents types d'éléments réseau, avant d'expliquer la méthode de calcul qu'il envisage.

##### 1.1 Modélisation des flux

On considère dans ce modèle des flux, c'est-à-dire une succession de paquets de données, éventuellement de taille variable. On va caractériser chacun de ces flux, en un point du réseau donné, grâce à une fonction croissante qui représente le trafic cumulé durant une certaine durée de temps. De façon plus formelle, on va noter  $R$  la fonction de trafic cumulé, qui sera telle que  $R(t_1, t_2)$  représentera la quantité totale d'information transmise par le flux en ce point, entre les instants  $t_1$  et  $t_2$ . On notera de manière abrégée  $R(t) \equiv R(0, t)$ . De manière immédiate, la fonction  $R(t)$  est positive et croissante, et une rafale de trafic est caractérisée par une forte augmentation de la fonction  $R$  durant un intervalle temporel de faible amplitude.

L'idée de Cruz est précisément de borner l'importance de ces rafales par une fonction qui ne dépend que de l'amplitude d'un intervalle temporel. Voici la définition de cette fonction, notée  $R^*$  et couramment appelée une enveloppe du flux :

$$\forall(t_1, t_2), R(t_1, t_2) \leq R^*(t_2 - t_1)$$

Par exemple si  $R^*$  est affine, la borne sur la quantité d'information que pourra émettre un flux durant un intervalle de temps sera égal à une constante plus une quantité proportionnelle à la durée de l'intervalle. Il est immédiat de voir que si  $R^*$  est une enveloppe d'un flux, alors toute fonction  $S$  telle que  $\forall t, S(t) \geq R^*(t)$  est aussi une enveloppe du flux. De même, si deux flux décrits par  $R_1$  et  $R_2$  ont respectivement pour enveloppe  $R_1^*$  et  $R_2^*$ , alors le flux formé des paquets de  $R_1$  et  $R_2$  aura pour enveloppe  $R_1^* + R_2^*$ .

On voit donc que la donnée de l'enveloppe permet d'exprimer des contraintes relatives à la régularité d'un flux, puisqu'elle permet de quantifier les rafales de trafic émises.

A partir de cette définition, il est assez immédiat de découvrir une relation entre l'enveloppe en entrée et celle en sortie de certains éléments réseau. Cruz considère l'exemple d'un élément à délai borné  $D$ , c'est-à-dire que toute information qui y pénètre n'y séjournera pas plus de  $D$ . Soit  $R_i^*$  l'enveloppe d'entrée d'un flux dans un élément de ce type. Cruz démontre que la fonction  $R_{i+1}^*$  telle que :

$$R_{i+1}^*(t) = R_i^*(t + D) \quad \text{Théorème 1}$$

est une enveloppe du flux en sortie. Ce résultat est très important, puisque couplé avec une méthode pour déterminer une borne sur le délai dans un élément, il forme la base d'une méthode d'analyse d'un réseau complet, qui sera développée dans 1.4. Il est toutefois évident qu'une meilleure modélisation du fonctionnement de chaque élément permettra d'améliorer ce résultat. Les prochains paragraphes présenteront les modélisations d'un certain nombre d'éléments simples.

## 1.2 Modélisation des éléments simples

### 1.2.1 Le lien à délai constant

Un lien à délai constant  $D$  est un élément qui possède un seul point d'entrée et un seul point de sortie. Toute donnée qui entre dans cet élément en ressort exactement  $D$  unités de temps plus tard. Soient respectivement  $R_i$  et  $R_{i+1}$  les fonctions d'entrée et de sortie d'un flux dans cet élément. On a par définition :

$$R_{i+1}(t) = R_i(t - D)$$

En ce qui concerne les enveloppes, il est facile d'un voir qu'un tel élément ne modifie pas la régularité d'un flux. On aura donc :

$$R_{i+1}^*(t) = R_i^*(t)$$

Un tel élément permet à l'évidence de modéliser le délai de propagation dû à un lien physique, ou bien pour modéliser des éléments qui ne traitent pas les données immédiatement.

### 1.2.2 Le tampon de réception

Un tampon de réception possède un seul point d'entrée et un seul point de sortie. Les données arrivent sur un lien physique de capacité  $C$ , et sont émises sur le point de sortie dès qu'un paquet a été reçu en entier. A l'évidence, si la taille des paquets est bornée par  $L$ , alors le délai subi par un bit ne saurait dépasser  $L/C$ . Grâce à cela, il est possible avec le théorème 1 de déterminer une enveloppe de sortie du tampon à partir de l'enveloppe d'entrée.

Le tampon de réception permet de modéliser des éléments du réseau qui ont besoin de recevoir le paquet en intégralité avant de le traiter, par exemple pour effectuer une vérification d'intégrité du paquet.

### 1.2.3 Le démultiplexeur

Un démultiplexeur est un élément qui a un seul point d'entrée et plusieurs points de sortie. Son rôle est de séparer différents sous-flux qui arrivent sur son point d'entrée. On prend comme hypothèse que les paquets sont marqués, de telle sorte que le démultiplexeur peut instantanément déterminer à quel sous-flux appartiennent les données. On rappelle que Cruz fait l'hypothèse d'un routage statique.

Dans le cas d'un démultiplexeur à deux sorties, si  $R_1$  et  $R_2$  décrivent les deux sous-flux en entrée, l'entrée du démultiplexeur sera décrite par  $R_1 + R_2$ , et les deux sorties seront respectivement décrites par  $R_1$  et  $R_2$ .

### 1.3 **Modélisation d'un multiplexeur**

Un multiplexeur est un élément possédant plusieurs entrées et un seul point de sortie. Sa fonction est de regrouper tous les flux entrants sur la seule sortie. Cruz fait l'hypothèse qu'un multiplexeur peut fonctionner en mode « cut-trough », c'est-à-dire que les bits d'un flux peuvent sortir immédiatement du multiplexeur. On va noter  $D$  la borne sur le délai subi par un bit quelconque à l'intérieur du multiplexeur, et  $Q(t)$  la quantité de bits qui y sont présents à l'instant  $t$ , aussi appelé l'arriéré de travail.

Cruz considère trois types de multiplexeurs, caractérisés par leur politique de service : FIFO, FIFO par flux, ou générique. Dans le premier type, les paquets sont retransmis dans l'ordre dans lequel ils sont arrivés. La convention adoptée est qu'un paquet est arrivé dès que son premier bit est reçu dans le multiplexeur. Les arrivées simultanées sur plusieurs liens sont départagées de manière arbitraire. On fait aussi l'hypothèse qu'un multiplexeur FIFO est conservatif, c'est-à-dire qu'il ne reste jamais inactif s'il a du travail à accomplir. Formellement, si  $Q(t) > 0$ , alors le débit instantané de sortie est égal à la capacité maximale du lien de sortie.

Pour plus de généralité, les deux autres types de multiplexeurs ne sont pas forcément conservatifs. Plus précisément, ils peuvent avoir des périodes de repos, qui peuvent modéliser par exemple un mécanisme de polling ou tout autre effet qui empêche l'émission durant une période. Par définition, une période de repos ne peut débuter au temps  $t$  que si  $Q(t) = 0$  et si le débit instantané sur le lien de sortie est nul. Durant une période de repos, le multiplexeur ne transmet aucun paquet, et toute donnée qui arrive est mise en attente. Evidemment, Cruz considère que ces périodes de repos sont bornées, on note  $V$  cette borne.

Un multiplexeur générique a pour caractéristique de retransmettre les paquets reçus dans un ordre aléatoire, sans autre contrainte que le fait de ne pas perdre de temps lorsqu'il n'est pas en période de repos.

Un multiplexeur FIFO par flux fait en sorte de retransmettre les paquets appartenant au même flux dans leur ordre d'arrivée, mais n'apporte aucune garantie en ce qui concerne les flux entre eux. On remarque qu'un multiplexeur FIFO est un cas particulier du multiplexeur FIFO par flux, si la durée maximale d'une période de repos est nulle.

Les paragraphes suivants étudient le comportement des multiplexeurs. Les différentes formules sont exprimées dans le cas où il n'y a que deux flux, mais elles peuvent être facilement étendues au cas plus complexe de  $n$  flux. On utilise les notations suivantes : les deux flux d'entrée sont décrits par  $R_1$  et  $R_2$ , et possèdent comme enveloppe  $R_1^*$  et  $R_2^*$ . La taille des paquets est fixe et notée  $L$ . Les capacités des liens d'entrée sont notées  $C_1$  et  $C_2$ , et celle du lien de sortie  $C_{out}$ . Pour les multiplexeurs non-conservatifs, on note la durée maximale d'une période de repos  $V$ . Enfin, on prend l'hypothèse de stabilité suivante :

$$\lim_{t \rightarrow \infty} (R_1^*(t) + R_2^*(t) - C_{out} \cdot t) = -\infty$$

Ceci est une condition suffisante : elle indique qu'au bout d'un long moment, le multiplexeur finit toujours pas évacuer tout arriéré de travail, et donc que le système ne peut devenir instable.

### 1.3.1 Bornes sur le délai

Cruz démontre les résultats suivants :

Dans un multiplexeur FIFO, le délai subi par un bit du flux 1 est majoré par :

$$D_{FIFO,1} = \frac{1}{C_{out}} \max_{u \geq 0} [R_1^*(u) + R_2^*(u + (L/C_2)) - C_{out} \cdot u]$$

Dans un multiplexeur générique, le délai subi par un bit quelconque est majoré par :

$$\bar{D} = \sup \left\{ \alpha : \alpha \geq 0, R_1^*(t) + R_2^*(t) + C_{out} \cdot V \geq C_{out} \cdot \alpha \right\}$$

Enfin, dans un multiplexeur FIFO par flux, le délai subi par un bit du flux 1 est majoré par :

$$\bar{D}_1 = \max \left\{ \alpha : \alpha \geq 0, \max_{\Delta \geq 0} [C_{out} \cdot V + R_1^*(\Delta) + R_2^*(\Delta + \alpha) - C_{out} \cdot \alpha] \geq C_{out} \cdot \alpha \right\}$$

### 1.3.2 Bornes sur les enveloppes en sortie

Pour un multiplexeur FIFO, le flux 1 possède en sortie l'enveloppe suivante :

$$R_{1,i+1}^*(t) = \min \left\{ C_{out} \cdot t, \max_{\substack{D \geq 0 \\ \Delta \geq 0}} \left[ \min \left\{ R_1^*(t + D), R_1^*(t + D + \Delta) + R_2^*(\Delta + \frac{L}{C_2}) - C_{out} \cdot (D + \Delta) \right\} \right] \right\}$$

Pour un multiplexeur générique, le flux 1 possède en sortie l'enveloppe suivante :

$$R_{1,i+1}^*(t) = \min \left\{ C_{out} \cdot t, \max_{\Delta \geq 0} [R_1^*(t + \Delta + V) + \min \{ R_2^*(\Delta + V), C_{out} \cdot \Delta \}] - C_{out} \cdot \Delta \right\}$$

De plus, il démontre que si les enveloppes en entrée sont concaves et continues, cette borne est serrée, ce qui veut dire qu'on ne peut pas trouver d'enveloppe plus petite que celle-ci avec les hypothèses considérées. On peut même construire des flux 1 et 2 et un multiplexeur FIFO par flux tel que l'enveloppe du flux 1 en sortie soit exactement la fonction définie ci-dessus. Cette fonction est donc aussi une borne à l'enveloppe de sortie, pour un multiplexeur FIFO par flux cette fois.

### 1.3.3 Bornes sur l'arriéré de travail

Pour un multiplexeur conservatif, et donc en particulier un multiplexeur FIFO, l'arriéré est majoré par :

$$\bar{Q} = \sup \left\{ x \geq 0 : R_1^*(x) + R_2^*(x) - C_{out} \cdot x \geq 0 \right\}$$

Pour un multiplexeur générique, et donc pour un multiplexeur FIFO par flux pour la même raison que précédemment, l'arriéré pour le flux 1 est majoré par :

$$\bar{Q}_1 = \max_{\Delta \geq 0} [R_1^*(\Delta + V) + \min \{ R_2^*(\Delta + V), C_{out} \cdot \Delta \}] - C_{out} \cdot \Delta]$$

### 1.3.4 Remarque sur l'utilisation de ces résultats

Dans certains cas où les fonctions de trafic cumulé possèdent des formes suffisamment simples (typiquement affine), il est possible de trouver des expressions explicites pour toutes les formules citées ci-dessus. Cruz en donne d'ailleurs certaines dans le cas de deux flux en entrée. Cependant, dans la majorité des cas, les formules précédentes sont difficilement

exploitables. En effet, il est souvent difficile de trouver les différents extremums qui y sont utilisés. On citera comme simple exemple le cas où les flux sont décrits par des fonctions en escalier : le flux en entrée dans le multiplexeur est donc décrit par une somme de fonctions en escalier. Or, pour un tel type de fonctions, il est difficile de trouver les coordonnées de l'intersection avec une simple droite affine, ce qui est pourtant nécessaire pour la formule 0 par exemple.

Ceci peut être considéré comme une limitation à l'utilité de ces résultats. Cependant, si on les utilise avec des fonctions affines, les formes simplifiées s'avèrent parfaitement exploitables.

#### **1.4 Méthode de calcul sur un réseau complet**

A partir des résultats précédents, Cruz propose une méthode de calcul pour analyser un réseau d'interconnexion des éléments définis plus haut, et pour lequel on connaît toutes les enveloppes des flux au point de leur entrée dans le réseau. Le principe de la méthode étant détaillé dans l'article, nous nous contenterons d'en présenter les principaux points.

Tout d'abord, il s'agit d'une méthode qui utilise explicitement des enveloppes affines pour les différents flux, de par l'utilisation de certaines propriétés de linéarité. D'autre part, elle ne peut s'appliquer directement qu'à un ensemble de flux unicast, car elle utilise l'ordre des différents éléments traversés par un flux. Enfin, cette méthode exige de représenter l'interconnexion des éléments par une matrice, dont les éléments dépendent des paramètres initiaux des flux à leur entrée dans le réseau. Le point clé de la méthode est l'inversion de cette matrice, ce qui n'est bien sûr pas toujours possible et pose des problèmes de conditionnement de matrice.

Il existe toutefois un cas particulier dans lequel cette méthode se réduit à une expression assez simple : il s'agit du cas où le routage des différents flux ne fait pas apparaître de « boucle de trafic ». Pour expliciter la notion de boucle, il faut introduire la relation suivante :

On dit que le multiplexeur M2 est en aval du multiplexeur M1 si et seulement si :

- Le prochain multiplexeur rencontré au long de son chemin par un flux sortant de M1 est M2, ou
- M2 est en aval d'un multiplexeur qui est lui-même en aval de M1 (transitivité)

On dit inversement que M1 est en amont de M2. Une boucle peut se définir comme étant la présence dans le réseau de deux multiplexeurs tels que l'un est à la fois en amont et en aval de l'autre.

Un réseau « feed-forward » est un réseau dépourvu de telles boucles de confluence des flux. Dans ce type de réseau, la méthode d'analyse est simplifiée, et utilise exclusivement les résultats des parties 1.2 et 1.3. En effet, on connaît les enveloppes des flux aux différents points d'entrée dans le réseau. Il suffit donc d'analyser les éléments qui reçoivent ces flux initiaux (on peut en analyser au moins un sinon il y aurait une boucle), puis de procéder par un calcul de proche en proche : l'analyse de chaque élément permet de déterminer les enveloppes en sortie de celui-ci, qui sont en entrée d'un autre élément. Ainsi, d'étape en étape on connaît l'enveloppe des flux dans de plus en plus d'endroits du réseau, jusqu'à analyse complète de celui-ci (complète car le réseau est dépourvu de boucles).

Cette méthode est certes moins générale que la méthode globale, puisqu'elle ne s'applique qu'à des topologies ne présentant pas de boucles, mais elle possède l'avantage de pouvoir s'appliquer à toutes les configurations de trafic sans boucle, du moment que chacun des éléments du réseau satisfait sa condition de stabilité.

## 2 Pourquoi le Network Calculus

Nous détaillons dans ce paragraphe les raisons pour lesquelles il nous a semblé intéressant d'utiliser la théorie du Network Calculus pour répondre à notre problématique. Nous indiquons également pourquoi cette théorie peut s'appliquer à notre contexte, alors qu'elle a été développée dans un contexte différent.

### 2.1 Intérêt de la méthode

La principale caractéristique de la méthode d'analyse de réseau proposée par Cruz est qu'elle fournit des résultats déterministes. A partir du moment où les flux qui entrent dans un réseau peuvent être modélisés par la notion d'enveloppe, et se comportent conformément à cette enveloppe, la méthode permet de fournir des bornes qui seront absolues, dans le sens où la probabilité pour qu'elles soient dépassées est nulle. Ceci provient du fait que la notion d'enveloppe ne considère pas un intervalle temporel particulier, mais plutôt tous les intervalles d'une amplitude donnée. De plus, les différentes bornes sont obtenues en considérant les extremums de quantités pour toutes ces amplitudes différentes. On s'affranchit ainsi de la notion de recherche de pire cas, qui pourrait être envisagée dans une démarche de recherche de bornes déterministes. En effet, une recherche de pire cas implique notamment l'observation d'évènements sur un intervalle particulier. On essaie généralement de particulariser l'intervalle considéré en le caractérisant par un évènement donné, par exemple l'arrivée du premier paquet d'un flux dans un élément. Ce faisant, se pose obligatoirement la question de savoir si on considère vraiment l'intervalle où se déroule le pire cas. Une démarche de pire cas se déroule donc toujours en deux phases : d'abord l'intuition du pire cas, et l'étude de celui-ci, puis la preuve que le cas considéré est bien le pire. Cette dernière partie est d'ailleurs fréquemment difficile, et l'expérience montre que l'intuition du cas pire est rarement justifiée, surtout dans des cas complexes.

La méthode de Cruz, parce qu'elle considère tous les intervalles possibles, présente donc l'avantage de fournir des bornes exactes, et avec des efforts de démonstration moindres. Or ceci est particulièrement important en vue de la certification du système avionique, qui nécessite que toute borne calculée soit soigneusement démontrée.

### 2.2 Applicabilité du Network Calculus

Nous avons vu qu'il n'est pas possible d'appliquer les résultats du Network Calculus à n'importe quel réseau. Les prochains paragraphes explicitent les hypothèses qui sont vérifiées pour un réseau avionique conforme à l'ARINC 664.

#### 2.2.1 Construction d'un modèle

Comme souvent, on applique la méthode d'analyse sur un modèle du réseau. Il est bien évident que la performance globale de l'analyse dépendra de la validité et de la qualité de la démarche de modélisation. Dans un premier temps, on peut se questionner sur l'existence même d'un tel modèle du réseau. En effet, les différents éléments présentés dans les paragraphes 1.2 et 1.3 ne permettent pas de modéliser n'importe quel type de comportement. En particulier, ce sont des éléments statiques, dont les caractéristiques sont bien définies. Il est donc impossible de les utiliser pour modéliser un caractère dynamique.

Dans le cadre d'un réseau conforme à l'ARINC 664, cette limitation n'en est pas une puisque la norme impose un caractère statique du réseau, comme indiqué dans le chapitre précédent.. Il sera donc possible d'analyser le réseau avec un modèle unique, en sachant que celui-ci sera valable durant toutes les étapes de la mission de l'aéronef, aussi bien au sol qu'en vol.

Une autre question que l'on peut se poser est de savoir si on possède suffisamment de connaissances sur les différents éléments du réseau, pour permettre l'élaboration d'un modèle



réaliste. Dans le monde de l'Internet, on considère qu'il est difficile d'appliquer le Network Calculus, en partie parce qu'on ne sait pas construire un modèle d'une partie entière du réseau. Bien souvent, les équipements possèdent une grande hétérogénéité, et il n'y a pas d'entité centrale qui connaisse les caractéristiques de tous les connectés au réseau. Le cas d'un réseau avionique est très différent. En effet, il existe toujours un responsable clairement défini pour chacun des différents systèmes. Ainsi, chaque équipementier est responsable de son matériel, et l'avionneur est responsable de l'intégration de tous ces équipements dans l'aéronef. Chaque équipementier peut donc fournir un modèle de son matériel, qui sera forcément précis de par la connaissance complète du matériel par son fabricant. De même, l'avionneur sera capable de fournir un modèle précis de l'interconnexion des différents équipements, puisque c'est lui qui la définit puis la réalise. On voit donc qu'il est possible de modéliser un réseau avionique conforme à l'ARINC 664, et même que ce modèle peut être d'une très bonne précision.

### 2.2.2 Flux d'entrée

Une autre raison pour laquelle il est difficile d'appliquer le Network Calculus au cas de l'Internet est que le trafic qui y circule est dans la plupart des cas mal connu. De nombreuses recherches actuelles visent à créer à partir de mesures réelles un modèle de ces trafics, ou au moins à catégoriser ces flux en diverses « races » [34]. Dans le cas particulier d'architectures à qualité de service, il est possible de connaître les caractéristiques des flux qui pénètrent le réseau grâce au contrat de trafic qui est lié avec le gestionnaire du réseau lors de l'établissement de connexion (contrôle d'admission). Dans un de ces réseaux, il est possible d'appliquer les résultats du Network Calculus, comme l'a prouvé Le Boudec dans [33].

Cependant, il apparaît clairement que ce genre d'analyses ne peut être faite que dans des cas où le comportement des flux est statique : pas d'admissions de nouveau flux dans le réseau, par exemple. Ceci limite donc les possibilités offertes par cette méthode d'analyse.

Par contre, dans le cas du réseau avionique qui nous intéresse, les conditions se prêtent tout à fait à l'application de la méthode. Les flux sont connus à l'avance, puisque chaque responsable d'équipement qui désire transmettre une donnée sur le réseau doit auparavant en demander l'autorisation auprès du gestionnaire du réseau. Faute de cela, tous ses paquets seraient détruits, puisque appartenant à un flux non identifié. En effet, la norme impose pour les réseaux déterministes de vérifier dans chaque commutateur que toute trame reçue appartient bien à un flux identifié, et de détruire toutes les autres. Au moment de réaliser l'analyse, on sait donc exactement quels sont les équipements susceptibles d'émettre des données à tout moment.

De plus, pour des raisons de sécurité évoquées dans le chapitre précédent, le comportement de ces flux est connu à l'avance. Chacun d'entre eux est tenu à respecter son contrat de trafic, et les éléments du réseau se comportent de sorte que toute infraction est sanctionnée par la perte de paquets. On peut donc appliquer les résultats du Network Calculus sur une caractérisation des flux dont on est sûr qu'elle ne variera pas en cours de mission, et qu'elle décrit bien le pire comportement que puissent avoir ces flux.

A titre d'exemple, on peut considérer le cas d'un contrat de trafic de la forme : le flux n'émettra que des données qui ne feront pas déborder un dispositif de type seau percé. Dans ce cas, on sait qu'alors le flux possède pour enveloppe la droite affine de pente le débit de sortie du seau, et pour ordonnée la taille maximale de ce seau. En fait, il y a même équivalence entre ces deux propriétés.

### 2.2.3 Topologie du réseau et routage des flux

On a vu dans le paragraphe 1.4 que la méthode de calcul de Cruz pour un réseau entier ne s'applique que dans certains types de réseau : soit tel que la matrice d'interconnexion est

inversible, soit de type « feed-forward ». En regardant ces conditions de plus près, on peut constater qu'elles ne dépendent que de trois paramètres : la topologie des liens physiques qui relie les éléments du réseau, les enveloppes initiales de chacun des flux, et finalement la route qu'empruntent ces flux pour aboutir à leur destination.

Encore une fois, le fait que notre étude se déroule dans le cadre des réseaux avioniques va permettre d'appliquer la méthode de Cruz. En effet, dans un réseau de type Internet personne ne possède le contrôle absolu de la topologie du réseau, et pour des raisons de performance évidentes, le routage y est dynamique. Or une des caractéristiques des réseaux ARINC 664 est qu'il existe un responsable unique, qui possède tout le contrôle du réseau et des flux qui y circulent. Il peut en particulier modifier le plan de raccordement des différents composants du réseau. Il peut également décider quelle sera la route qu'emprunteront les différents flux (on rappelle que les tables de commutation sont fixes). Enfin, il peut fixer les différents contrats de trafic à un niveau qu'il considère adéquat.

On voit donc que le responsable du réseau possède toute latitude pour concevoir un réseau tel qu'il sera aisé d'y appliquer la méthode de calcul de Cruz.

On a présenté ici les raisons pour lesquelles il est possible d'appliquer le Network Calculus à un réseau avionique : il est possible d'en construire un modèle fidèle, le trafic en entrée du réseau est contraint, et la topologie et le routage y sont très adaptés à la méthode envisagée. On souligne l'importance du rôle central joué par l'avionneur, qui possède toute la connaissance du réseau ; cela constitue la principale différence avec des réseaux plus libres ou hétérogènes, tels qu'Internet où n'existe pas de contrôle central.

Nous présentons dans la partie suivante la façon d'appliquer le Network Calculus dans notre contexte.

### **3 Application du Network Calculus au réseau étudié**

#### **3.1 Description du cas d'étude**

Le cas d'étude que nous avons considéré nous a été communiqué par Airbus. Il ne s'agit en aucun cas de la description exacte du futur réseau embarqué de l'A380, mais d'une maquette qui a été jugée suffisamment représentative et complexe, de sorte qu'elle a pu servir à tester certains outils de conception.

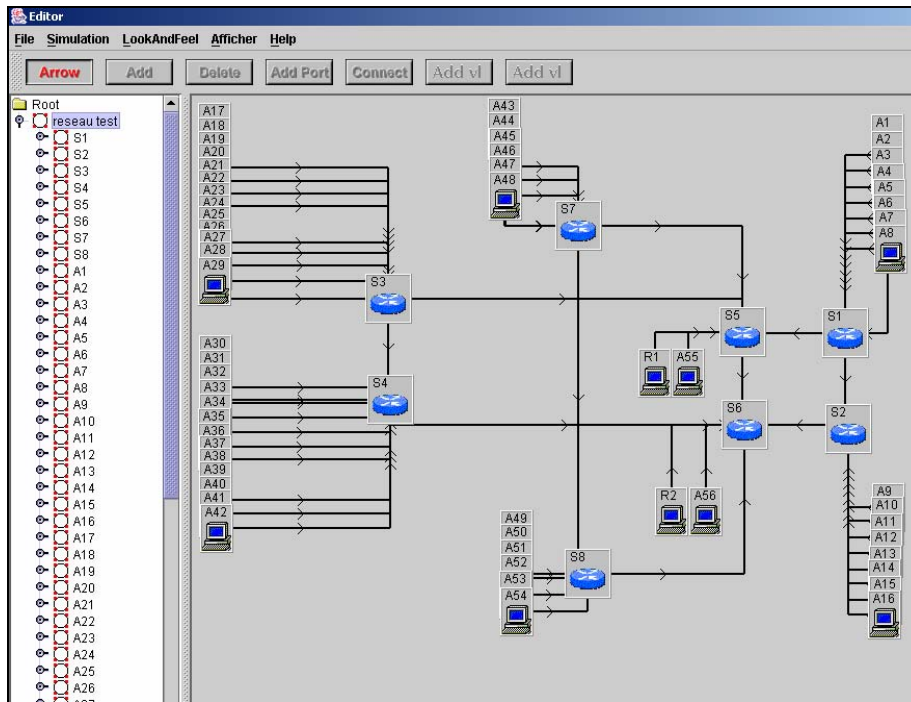


Figure 2 Le réseau de test

Il s'agit d'un réseau composé de commutateurs reliés entre eux, auxquels sont raccordés des stations émettrices également appelées abonnés. La topologie du réseau est relativement simple, et quasiment symétrique. On retrouve ici une architecture commune en avionique, où on considère souvent l'aéronef comme constitué de deux bords. Ceci permet d'apporter à l'architecture une redondance fondamentale pour la sécurité. De façon générale, les données seront traitées sur chaque bord par des équipements correspondants, qui communiqueront ensuite pour consolider leurs résultats.

Le réseau est constitué d'une cinquantaine d'abonnés, et de huit commutateurs. Ces commutateurs utilisent la politique de service FIFO. S'agissant d'un réseau commuté Full-Duplex, on retrouve bien sûr le fait que chaque abonné n'est relié qu'à un seul commutateur. Le trafic est constitué d'environ 500 flux multicast, ayant entre un et 5 destinataires. La plupart d'entre eux sont constitués de paquets de petite taille puisqu'ils embarquent une charge utile d'une vingtaine d'octets. La cadence d'émission est de l'ordre de grandeur d'un paquet toutes les trentaines de millisecondes. Seuls quelques flux de contrôle sortent de ce gabarit, et nécessitent une attention particulière puisqu'ils sont constitués de plus gros paquets (plusieurs centaines d'octets de charge utile) émis plus fréquemment, environ un paquet toutes les quelques millisecondes.

Le routage des flux est lui aussi relativement simple, et suit l'algorithme suivant :

Soit un flux émis par l'abonné A vers l'abonné B (on rappelle que les VL peuvent être des flux multicast). On note  $S_A$  le commutateur relié à A, et  $S_B$  celui de l'abonné destination.

Si  $S_A$  et  $S_B$  sont confondus, le flux n'emprunte que ce commutateur.

Si  $S_A$  et  $S_B$  sont directement reliés, le flux les traverse successivement.

Dans les autres cas, récursivement, le flux se propage de  $S_A$  vers le commutateur ayant la plus grande connectivité (qui est directement relié au plus de commutateurs).

Cet algorithme implique évidemment un fort trafic sur les liens reliant les commutateurs centraux (qui sont ceux de plus grande connectivité), mais il présente un avantage important : allié à la topologie simple du réseau, il permet d'éviter toute boucle de trafic, ce qui nous l'avons vu est d'une grande importance pour en permettre l'analyse.

### 3.2 Modélisation du trafic

Dans le cas d'étude considéré, le trafic nous a été décrit grâce à l'utilisation de la notion de Virtual Link (VL), présentée dans le chapitre précédent.

Revenons sur la notion de BAG pour préciser les hypothèses retenues par la suite. Si on considère un modèle du trafic par paquets, la définition du BAG est immédiate : le temps minimum entre deux paquets, dont l'apparition est supposée instantanée. Par contre, si le modèle retenu est un modèle fluide, c'est-à-dire que chaque bit d'un paquet est transmis tour à tour, il convient de mieux définir le BAG. S'agit-il du temps minimum entre les premiers bits de deux paquets consécutifs? Entre les deux derniers bits de deux paquets consécutifs? Entre le dernier bit du premier paquet et le premier du second paquet? D'autre part, la connaissance du temps qui sépare le premier du dernier bit du paquet suppose qu'on connaît la vitesse de transmission en l'élément considéré...

Nous avons choisi, pour cette première étude, de considérer un modèle de trafic par paquets, ce qui nous permet de nous affranchir de ces difficultés très liées à l'implémentation retenue.

#### 3.2.1 Enveloppe initiale

Pour mener à bien l'analyse du réseau avec la méthode de Cruz, nous avons vu qu'il est fondamental de bien définir les enveloppes initiales des flux, c'est-à-dire les enveloppes au point d'entrée dans le réseau. Pour ce faire, nous utilisons les informations fournies par la définition de chaque VL. Celles-ci peuvent être facilement traduites en une enveloppe de forme affine. En effet, en reprenant la définition d'une enveloppe, on constate que l'ordonnée à l'origine d'une droite représente la quantité maximale de bits que peut émettre instantanément le flux. La pente de la droite représente le débit moyen maximal admissible pour le flux. Dans notre cas, le flux ne saurait émettre plus de  $S_{\max}$  bits instantanément, puisque c'est la taille maximale d'une trame : l'ordonnée à l'origine de son enveloppe est donc  $S_{\max}$ . D'autre part, le flux ne peut émettre deux trames consécutives que séparées d'un intervalle minimum de BAG. Ceci implique donc que le débit maximal émis par le flux sera de  $\frac{S_{\max}}{BAG}$  bits par unité de temps.

On a donc simplement comme enveloppe pour le flux:

$$R^*(t) = S_{\max} + \frac{S_{\max}}{BAG} \cdot t$$

#### 3.2.2 Modélisation du caractère multicast des VL

Les VL peuvent être multicast, c'est-à-dire destinés à plusieurs abonnés différents. Au lieu de modéliser ces VL par un ensemble de flux unicast (représentant chacun une communication de bout en bout), circulant dans le réseau conformément au routage présenté précédemment, on a choisi une modélisation sous forme d'arbre de segments de flux unicast. Dans les parties où tous les flux unicast suivent le même chemin (typiquement au début), le VL n'est représenté que par un seul flux. Dès que les routes vers des abonnés différents divergent, on duplique le flux précédent, créant ainsi une nouvelle branche de l'arbre.

Chaque succession de segments de flux entre la racine et une feuille de l'arbre représente donc un chemin de bout en bout, d'un abonné à l'autre.

### 3.3 Modélisation des éléments du réseau

#### 3.3.1 L'abonné

Un abonné est un équipement qui va être raccordé au réseau. La partie spécifique de l'abonné qui s'occupe de le relier au réseau AFDX se nomme le End System. La fonction principale du

End System est de fournir aux applications de l'abonné un service de transfert de données de niveau transport. C'est en particulier à lui de mettre en forme les flux conformément à la notion de VL. Il lui revient aussi de réaliser l'assemblage et le désassemblage de toutes les trames, ainsi que leur redirection vers les bons ports applicatifs.

Nous rappelons que notre étude prend comme point de départ les trames qui viennent d'être mises en conformité avec la notion de VL.

Pour ce faire, le End System contient nécessairement un dispositif de régulation des flux. La définition des VL étant relativement simple, il est facile de construire un tel dispositif, de type seau percé. Pour chaque VL, le End System doit maintenir un compte. Ce compte se remplit proportionnellement au temps qui passe, à la vitesse  $1/BAG$  unité par unité de temps. Il peut prendre la valeur maximale de 1, et est initialisé à cette valeur. Une trame arrivante est émise si le compte possède la valeur 1, sinon elle est stockée jusqu'à ce que le compteur se remplisse. A la suite de son émission (qui nous le rappelons est supposée instantanée) la valeur du compteur est diminuée de 1.

Cet algorithme très simple permet de manière évidente d'espacer les trames d'au moins BAG unités de temps.

Compte tenu du point de départ de notre étude, nous avons retenu un modèle de l'abonné simple, constitué de multiples sources régulées et d'un multiplexeur FIFO qui représente le port d'émission de l'abonné. A noter que le délai de transmission sera pris en compte dans le port d'entrée correspondant du commutateur, ce qui explique qu'il n'apparaisse pas ici.

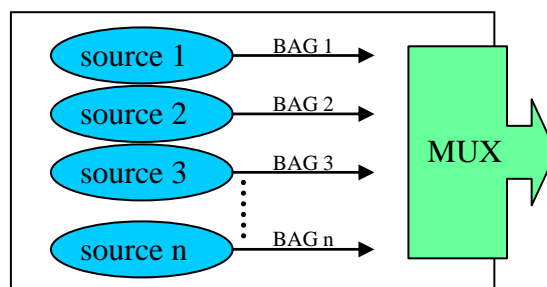


Figure 3 Modélisation d'un abonné

### 3.3.2 Les liens

L'émission des données sur les liens physiques engendre deux types de délais : le délai d'émission et le délai de propagation. Pour simplifier l'analyse du modèle, nous avons choisi de prendre en compte le délai d'émission des trames dans le commutateur (il est en effet égal au délai de réception de la trame dans le commutateur).

Le délai de propagation est pour sa part de nature constante. On peut noter que ce type de délai, dans le Network Calculus, ne modifie pas les enveloppes des flux : il ne modifie pas la « régularité » du flux que représente l'enveloppe. De ce fait, il n'est pas nécessaire de les prendre en compte au moment de l'analyse. Pour mesurer le délai de bout en bout global maximal par exemple, il suffira d'ajouter a posteriori la somme des délais constants, aux bornes sur le délai calculées par le Network Calculus. Dans la suite de cette étude, nous étudierons donc seulement les délais variables, considérant que l'étude des délais constants peut être effectuée parallèlement de manière indépendante.

Dans ce cadre, le modèle de lien retenu est très simple : puisque le délai d'émission est pris en compte dans le port d'entrée du commutateur, notre lien n'est qu'un répéteur, qui a pour fonction de propager les enveloppes des flux, sans les modifier.

### 3.3.3 Le commutateur

Dans le chapitre précédent, nous avons présenté les fonctions globales d'un commutateur au sens du standard IEEE 802.1D. Cependant, comme l'impose la norme ARINC 664, les opérations de filtrage et de relais des trames ne peuvent être perturbées par les autres fonctions du commutateur, comme par exemple la surveillance de son bon fonctionnement. Nous avons donc décidé de ne prendre en compte dans notre modèle que les opérations liées au traitement des trames. Plus précisément, voici les tâches successives que va accomplir le commutateur pour une trame donnée:

- réception intégrale de la trame,
- vérification de l'intégrité de la trame (contrôle du champ FCS),
- vérification de la validité de la trame (longueur correcte, VL connu, adresse émetteur connue),
- filtrage suivant le trafic du VL (« policing »),
- consultation de la table de commutation,
- redirection sur le(s) bon(s) port(s) de sortie,
- émission de la trame au plus tôt.

Notre modèle de commutateur va naturellement refléter cette succession d'opérations, et donc son architecture générale sera relativement simple. La définition des caractéristiques précises de chacun des éléments de ce modèle provient d'exigences formulées par Airbus, pour qu'un commutateur soit déclaré propre à être embarqué. Il serait bien entendu intéressant d'affiner le modèle en le recalant par rapport aux performances d'un commutateur réel, plutôt que de se fier uniquement à ses spécifications. Malheureusement à l'heure où ce modèle a été choisi, il n'existait pas encore d'exemplaire opérationnel d'un commutateur conforme.

Comme indiqué plus tôt, nous avons décidé de prendre en compte la durée d'émission sur le medium physique lors de la réception de la trame dans le commutateur. Nous avons modélisé cette tâche (qui est en fait accomplie par le port d'entrée du commutateur) en utilisant un élément tampon de réception, en sachant que le débit entrant est fixé à 100 Mbits/s. On a vu que le délai imposé par ce tampon de réception dépend de la taille maximale des trames qu'il recevra ; il faudrait donc construire le modèle en fonction du trafic qui y circulera. Comme nous souhaitons pouvoir utiliser notre modèle pour différentes configurations, nous allons considérer que la taille maximale des trames que recevront tous les tampons de réception du réseau sera la taille maximale d'une trame Ethernet, soit 1538 octets, en tenant compte du préambule et du gap entre trames, comme l'indique la figure ci-dessous.

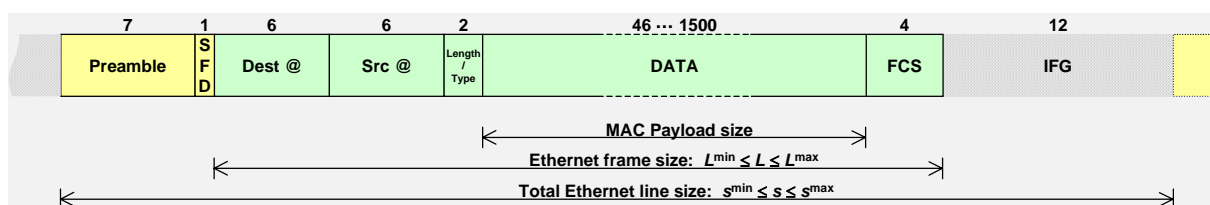


Figure 4 Format d'une trame Ethernet

Les opérations suivantes, à savoir le filtrage de la trame suivant les différents critères (taille maximale, adresse source connue, « policing », etc.), peuvent être effectuées plus ou moins rapidement, selon la manière dont elles ont été implémentées par le fabricant du matériel. Cependant, une des exigences de performance du cahier des charges spécifie que chaque port du commutateur doit être capable de filtrer 125 trames en une milliseconde. Nous avons donc décidé de modéliser ces opérations par un élément à délai borné, avec la valeur du délai fixée à 8  $\mu$ s.

L'opération suivante, l'aiguillage des trames vers le ou les bons ports de sortie, s'inscrit dans la même logique. Elle peut se dérouler plus ou moins vite selon le commutateur considéré, mais Airbus possède une exigence sur la vitesse de l'aiguillage. Le commutateur, même lorsque tous ses ports reçoivent bout à bout des trames de taille minimale, doit être capable d'aiguiller  $N \times 125$  trames en une milliseconde, où  $N$  est le nombre de ports. Pour chaque port, nous avons donc là aussi décidé de modéliser l'opération d'aiguillage par un élément à délai borné, avec la valeur du délai fixée à  $8 \mu\text{s}$ , suivi d'un passage dans un démultiplexeur. Enfin, l'émission de la trame sur le bon port de sortie peut être modélisée par un élément multiplexeur FIFO, dont la capacité de sortie sera fixée à celle du lien physique, à savoir  $100 \text{ Mbits/s}$ .

Voici donc la chaîne de traitement que va rencontrer chaque trame :

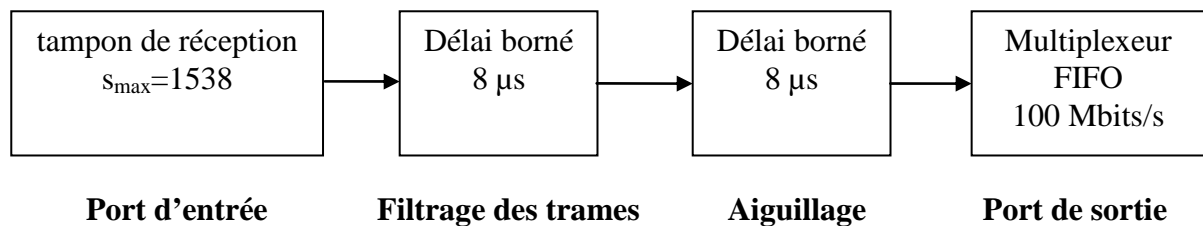


Figure 5 Chaîne de traitement d'une trame

On va maintenant utiliser cette chaîne pour construire le modèle du commutateur complet. Il suffit pour cela de se remémorer que l'élément à délai borné se comporte exactement de la même façon, qu'y passent un seul flux ou plusieurs. On va se servir de cette propriété pour « rassembler » tous ces éléments qui étaient en parallèle, puisqu'ils ont tous la même valeur de délai. Voici donc un premier modèle du commutateur complet :

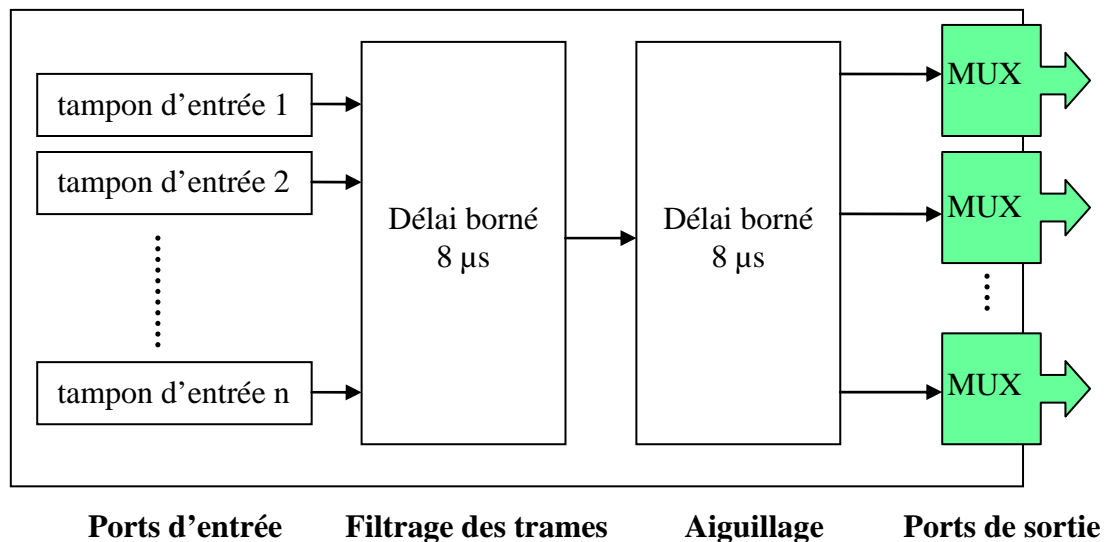


Figure Premier modèle du commutateur

Il faut bien garder en mémoire que pour le commutateur réel, il n'existe que des ports qui font à la fois l'entrée et la sortie, car on est dans le contexte d'une utilisation des liens Full Duplex, or le modèle ci-dessus introduit une dissymétrie entre port d'entrée et port de sortie. On peut

toutefois conserver ce modèle, en imposant la restriction suivante : si un abonné est relié au port  $i$  en entrée (il émet ses trames vers le port  $i$ ), alors il devra être relié aussi au port de sortie  $i$  (l'abonné recevra des trames du port de sortie du commutateur qui a le même numéro que celui vers lequel il émet). On capture ainsi à moindre frais le caractère Full Duplex, tout en conservant un modèle d'une grande lisibilité.

Enfin, on peut constater que le comportement du tampon de réception ne diffère pas de celui d'un élément à délai borné, pour lequel la valeur du délai serait égale à la taille maximale d'une trame divisée par la capacité du lien d'entrée, soit  $125 \mu\text{s}$ . Ceci nous permet alors d'établir notre modèle final du commutateur : en utilisant le fait que deux éléments à délai borné placés en série se comportent comme un seul élément, dont le délai vaut la somme des deux premiers.

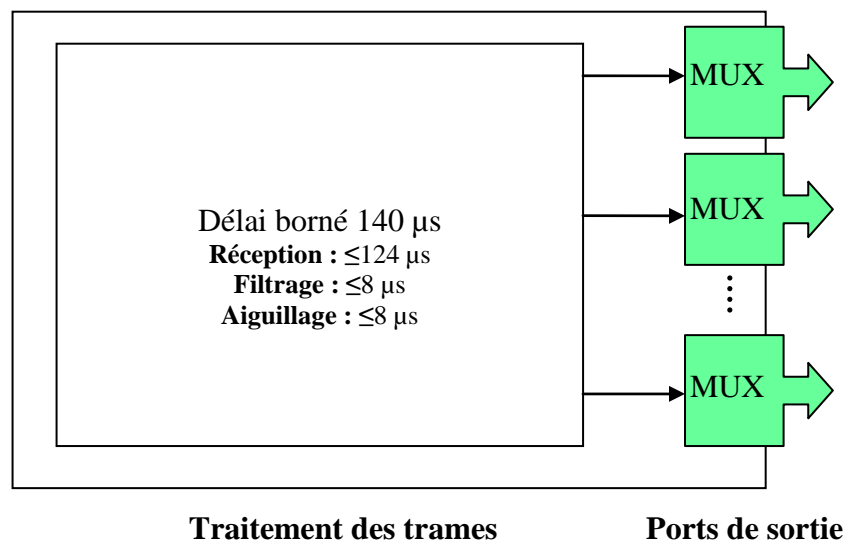


Figure 6 Modèle du commutateur

Bien entendu, ce modèle reste relativement simple, et serait certainement très enrichi par une étude comparée avec un commutateur réel. Cependant, nous pensons qu'il permet d'une part de bien capturer le principal effet des commutateurs sur le trafic, qui est dû au goulot d'étranglement que représentent les ports de sortie. D'autre part, les différents délais sont fixés à des valeurs qui doivent être pessimistes, puisqu'un commutateur qui ne respecterait pas ces valeurs ne pourrait pas être utilisé dans l'aéronef. Nos résultats seront donc peut être pessimistes, mais ils seront au moins valides, ce qui est l'objectif de l'étude.

### 3.3.3.1 Discussion sur le policing

Le réseau que nous avons étudié est conforme aux propositions de la norme ARINC 664 pour un réseau déterministe. En ce sens, il incorpore un dispositif permettant de réaliser une ségrégation spatiale des erreurs, c'est-à-dire de limiter la propagation d'erreurs dans le réseau. Ceci est réalisé par des fonctions dites de « policing » implantées dans chaque port d'entrée de chaque commutateur. Ces fonctions permettent de fixer pour chaque VL une limite sur le trafic maximal reçu ; si le VL ne se comporte pas nominalement et enfreint cette limite, les trames supplémentaires sont détruites. Ceci permet, dans le cas où un élément en panne émet subitement beaucoup trop de trafic, de limiter l'impact de ce dysfonctionnement sur tous les autres VL.

La façon la plus simple d'implémenter ce policing est d'utiliser le paradigme du seau percé : le seau se remplit à un taux fixé, avec une taille maximale donnée. Quand une trame arrive



dans le port, elle n'est acceptée que si le seau contient une quantité de liquide égale à la taille de la trame, et cette quantité est enlevée du seau ; dans le cas contraire, la trame est détruite et le liquide reste dans le seau.

Pour fixer les paramètres de ce seau percé (vitesse de remplissage, taille maximale), il est pratique d'utiliser l'enveloppe du VL, calculée en ce point par notre méthode. On sait en effet que si on choisit comme vitesse de remplissage la pente de l'enveloppe, et comme taille maximale l'ordonnée à l'origine de cette même enveloppe, alors le seau percé acceptera toutes les trames du VL considéré.

Lors de l'analyse du réseau, une question se pose cependant au sujet du policing : faut-il le prendre en compte ? En effet, si tous les VL se comportent de manière nominale, il n'y aura jamais de trame rejetée, et l'effet du policing sur le trafic généré par les VL sera nul. Dès lors, on peut choisir de ne pas modéliser ces fonctions. Cependant, il peut être intéressant d'analyser le comportement du réseau, en présence de VL qui émettent plus de trafic que prévu. Dans ce cas de nombreuses trames vont être rejetées. Pour analyser le réseau, on doit alors utiliser comme enveloppe des flux à la sortie des éléments de policing, l'enveloppe maximale que permettront de faire passer ces éléments, et non plus leur enveloppe d'entrée dans ces éléments.

Nous pensons qu'il s'agit principalement là d'un choix qui est offert à l'utilisateur de la méthode d'analyse et qui dépend du but de celle-ci : si on veut étudier le fonctionnement nominal du réseau, on peut ne pas modéliser le policing dans les commutateurs. Si on souhaite étudier un comportement dégradé du réseau en présence de pannes, alors il faudra modéliser les éléments de policing. La modélisation que nous proposons, dans le cas où ce policing utilise le paradigme du seau percé (b,r), est un élément qui va imposer à chaque flux une enveloppe  $b+r.t$ , quelle que soit l'enveloppe en entrée.

Dans le reste de ce chapitre, nous avons choisi de n'étudier que le fonctionnement nominal du réseau ; nous indiquerons cependant, dans le chapitre suivant, comment prendre en compte l'impact du policing sur notre méthode d'analyse, et les résultats obtenus.

### **3.4 Méthode de calcul mise en oeuvre**

L'objectif du calcul dans cette première étude était de calculer la latence de bout en bout pour chacun des flux, obtenue en additionnant tous les délais locaux subis par les flux. Plus précisément, s'agissant de flux multicast, on voulait déterminer une borne sur cette latence de bout en bout entre un abonné émetteur et chacun des abonnés récepteurs. Comme indiqué dans le premier chapitre de cette thèse, la latence est définie entre le moment où une trame sort du dispositif de contrôle de flux du End System de l'abonné, jusqu'au moment où elle pénètre dans le End System de l'abonné destination.

Cette latence peut être considérée comme la somme de deux types de délai : les délais fixes d'une part, correspondant par exemple à des latences de propagation du signal sur un lien physique, et d'autre part des délais pouvant être variables, qui vont dépendre surtout de l'ensemble du trafic des différents flux. Pour calculer la latence totale, il faut donc calculer la contribution de chacun des éléments rencontrés tout au long du parcours du flux dans le réseau. Le calcul des délais de propagation est bien connu, on sait par exemple que la propagation sur un lien de 20 mètres prend 0,114  $\mu$ s. C'est donc dans le calcul des délais dépendant de la configuration du trafic que va porter principalement notre étude.

Pour calculer les valeurs des différents délais, les résultats de Cruz nécessitent de connaître les enveloppes de chacun des flux dans tous les points du réseau. La notion de VL nous permet facilement de décrire les flux initiaux, c'est-à-dire de connaître les enveloppes des flux au moment où ils pénètrent dans notre domaine d'étude du réseau. Ensuite, le Network Calculus nous fournit un ensemble d'équations qui relie dans chaque élément l'enveloppe de sortie d'un flux à toutes les enveloppes des flux entrant dans cet élément. Notre cas d'étude

étant dépourvu de boucles de trafic, il est possible de résoudre ces équations séquentiellement. On peut alors réorganiser les équations « manuellement », ou bien procéder par une approche de type « dataflow ». On va simplement parcourir l'ensemble du réseau, et ne résoudre que les équations des éléments dont toutes les enveloppes d'entrée sont connues. On va ainsi réaliser une première « passe » de calcul, et les passes suivantes permettront de résoudre les équations pour de nouveaux éléments, jusqu'à ce que toutes les enveloppes soient connues.

Compte tenu de la description des flux par la notion de VL, nous avons décidé de modéliser les flux du réseau au moyen d'enveloppes affines notées  $(\sigma, \rho)$ . Ceci permet entre autres de simplifier l'expression de la formule 0 comme suit, en tenant compte de l'hypothèse du modèle de trafic par paquet présentée dans le paragraphe 3.2:

$$D = \frac{\sigma_1 + \sigma_2}{C}$$

Où C représente la capacité du lien de sortie, à savoir 100 Mbits/s. Cette formule se généralise très facilement au cas de n flux :

$$D = \frac{\sum_{i=1}^n \sigma_i}{C}$$

Par contre, les expressions concernant l'enveloppe des flux en sortie d'un multiplexeur FIFO sont difficilement généralisables. Nous avons alors choisi de déterminer cette enveloppe en considérant que le multiplexeur FIFO est en particulier un élément à délai borné D, et en utilisant le Théorème 1 : si l'enveloppe d'entrée d'un flux est  $R_i^*(t) = \sigma + \rho t$ , l'enveloppe en sortie sera :

$$R_{i+1}^*(t) = (\sigma + \rho.D) + \rho t$$

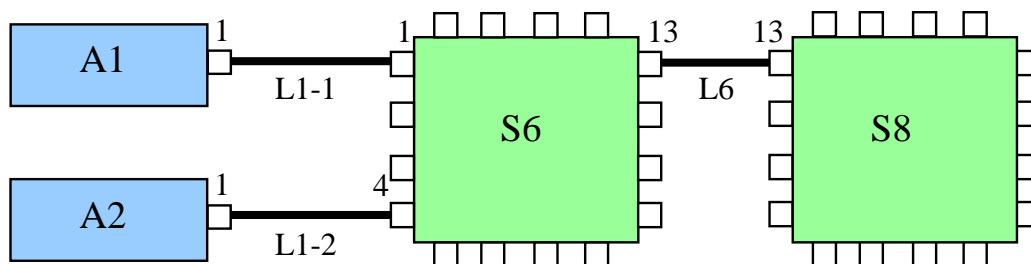
### 3.5 Description de l'outil réalisé

Nous avons réalisé pour cette étude un logiciel de calcul en Java qui permet d'automatiser l'analyse du réseau. Ce logiciel prend en entrée une description de l'architecture physique du réseau au format texte dont voici un extrait :

```

S6 FCFSSwitch S6-1 S6-2 S6-3 S6-4 S6-5 S6-6 S6-7
S6-8 S6-9 S6-10 S6-11 S6-12 S6-13 S6-14 S6-15 S6-16
S8 FCFSSwitch S8-1 S8-2 S8-3 S8-4 S8-5 S8-6 S8-7
S8-8 S8-9 S8-10 S8-11 S8-12 S8-13 S8-14 S8-15 S8-16
A1 EndSystem A1
A2 EndSystem A2
L6 TrafficLink S6-13 S8-13
L1-1 TrafficLink A1 S6-1
L1-2 TrafficLink A2 S6-4

```



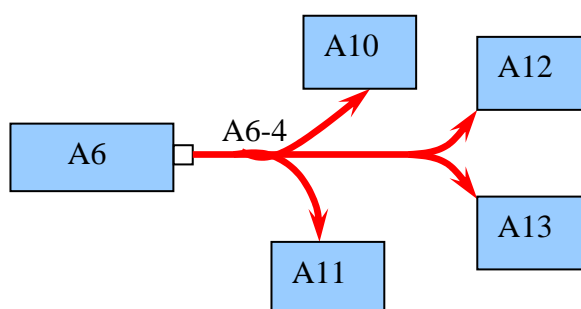
Il s'agit bien sûr d'une description toute simple, avec pour chaque élément un nom, un type, et le nom des différents ports. Le logiciel crée d'après ce fichier l'ensemble des objets qui représentent le réseau, en utilisant des éléments conformes à la modélisation présentée : multiplexeurs, démultiplexeurs, éléments à délai borné, etc...

Le logiciel prend également comme entrée une description du trafic du réseau, conforme à la notion de VL. On trouve pour chaque VL : un nom, le nom de l'élément émetteur, la valeur du BAG (en microsecondes), la taille maximale d'un paquet, et la liste des destinataires ; ci-dessous se trouve un extrait de ce fichier :

```

A6-Service-R1      A6 2000 226 R1
A6-1               A6 32000 16 A1 A7 A8
A6-2               A6 32000 16 A2 A3 A4 A5
A6-3               A6 32000 16 A9 A14 A15 A16
A6-4               A6 32000 16 A10 A11 A12 A13
A6-5               A6 32000 16 A30 A31 A32 A48
A6-6               A6 32000 16 A33 A34 A35 A47
A6-7               A6 32000 16 A36 A37 A38 A46
A6-8               A6 32000 16 A30 A31

```



Le logiciel va alors créer des objets VL, qui posséderont comme attributs les différentes caractéristiques présentées précédemment.

Par la suite, la difficulté réside dans la programmation des VL à l'intérieur du réseau. Nous avons choisi de le faire en représentant chaque VL par un arbre de flux individuels (implémentés par des objets de classe Flow). Chacun de ces objets représente le VL dans une portion du réseau, plus précisément entre deux éléments du modèle. Il possède en particulier en attribut deux valeurs (de type float) qui représentent l'enveloppe affine du VL entre ces deux points précis. La notion d'arbre de flux individuels permet de prendre en compte le caractère multicast des VL ; on l'instancie en faisant en sorte que chaque objet flow possède

une référence vers l'objet flow qui le précède immédiatement dans l'arbre. Ainsi, un parcours de l'arbre depuis la racine vers une des feuilles représente le chemin du VL entre l'abonné émetteur et un des abonnés récepteurs. Un tel parcours est par exemple effectué lors du calcul de la borne sur la latence de bout en bout, ce qui permet d' additionner toutes les contributions à la latence totale le long du chemin du VL.

Une fois les différents éléments et flux instanciés, on peut procéder à l'analyse du réseau. Celle-ci se déroule exactement de la façon indiquée au paragraphe 3.4, par passes successives, et jusqu'à ce qu'on ne puisse plus rien calculer de nouveau. A l'intérieur de chaque nœud, le calcul de délai se déroule selon les résultats de Cruz. Pour un multiplexeur, par exemple, on commence par faire la somme de tous les  $\sigma_i$  des flux entrants. Le résultat est alors obtenu en divisant cette valeur par la capacité du lien physique de sortie, soit 100 Mbits/s. Ensuite, le multiplexeur parcourt tous les objets flows qui sont issus de lui ; pour chacun, il calcule la nouvelle enveloppe, obtenue en prenant l'enveloppe en entrée et en la décalant de la valeur du délai vers la gauche, conformément au Théorème 1. Par exemple, pour une enveloppe d'entrée  $(\sigma_i, \rho_i)$ , l'enveloppe en sortie est fixée à :

$$\sigma_{i+1} = \sigma_i + \rho_i * D$$

$$\rho_{i+1} = \rho_i$$

L'enveloppe ainsi propagée permettra de calculer la valeur de la borne sur le délai dans l'élément suivant lors d'une prochaine passe de calcul.

Finalement, lorsque tous les calculs de borne ont été effectués, il ne reste plus qu'à trouver la valeur de la latence de bout en bout. On pourrait le faire en sommant chacune des bornes locales, tout au long d'un chemin ; vu la simplicité de notre modèle, nous avons préféré utiliser la remarque suivante : chaque élément va décaler l'enveloppe du flux vers la gauche, d'une quantité égale à la borne sur le délai en ce point. Dès lors, on peut facilement accéder à la valeur de la latence de bout en bout (à laquelle on devra rajouter les délais constants), qui sera égale à la valeur de la translation entre l'enveloppe initiale et l'enveloppe finale. On utilisera alors l'expression :

$$D_{total} = \frac{\sigma_{final} - s_{max}}{\rho_{final}}$$

La simplicité de notre modélisation des éléments et des flux, combinée à la méthode de calcul par propagation, accorde à notre outil de bonnes performances en terme de rapidité de calcul. Sur ce cas particulier de réseau, l'analyse est menée à bien en un temps très court, de l'ordre de la seconde.

## 4 Etude du réseau au moyen de l'outil développé

### 4.1 Calcul avec une politique FIFO

La topologie du réseau que nous avons étudié est symétrique, et les flux qui y circulent le sont également. Il est donc assez logique que les résultats fournis par notre analyse soient eux aussi symétriques, ce que nous constatons bien. De plus, nous avons décidé de ne représenter que les valeurs des bornes calculées dans les ports des commutateurs. En effet, les valeurs calculées dans d'autres éléments sont facilement obtenues, et d'importance relative faible. De même, nous n'avons pas représenté la latence de bout en bout pour chacun des flux, car nous n'avons pas de donnée objective à laquelle comparer nos résultats. Par contre, Airbus nous avait fourni une valeur maximale admissible pour le temps de traversée du commutateur (hors temps de réception) : 1 milliseconde. Nous savons que ce temps de traversée est

constitué du temps de filtrage de la trame (moins de 8 microsecondes), du temps d'aiguillage (moins de 8 microsecondes), et du temps d'attente dans le port de sortie. Compte tenu des ordres de grandeur de ces quantités, il est évident que la discussion se déroule sur la valeur de cette dernière contribution, ce qui explique notre choix de ne représenter que celle-ci. Voici donc un graphique qui représente, pour chacun des ports de sortie des commutateurs, la valeur de la borne sur le délai maximal, calculée en millisecondes. Compte tenu de la symétrie des résultats obtenus, ne figurent sur le graphique que la moitié des résultats obtenus, mais l'analyse a été conduite sur le réseau complet.

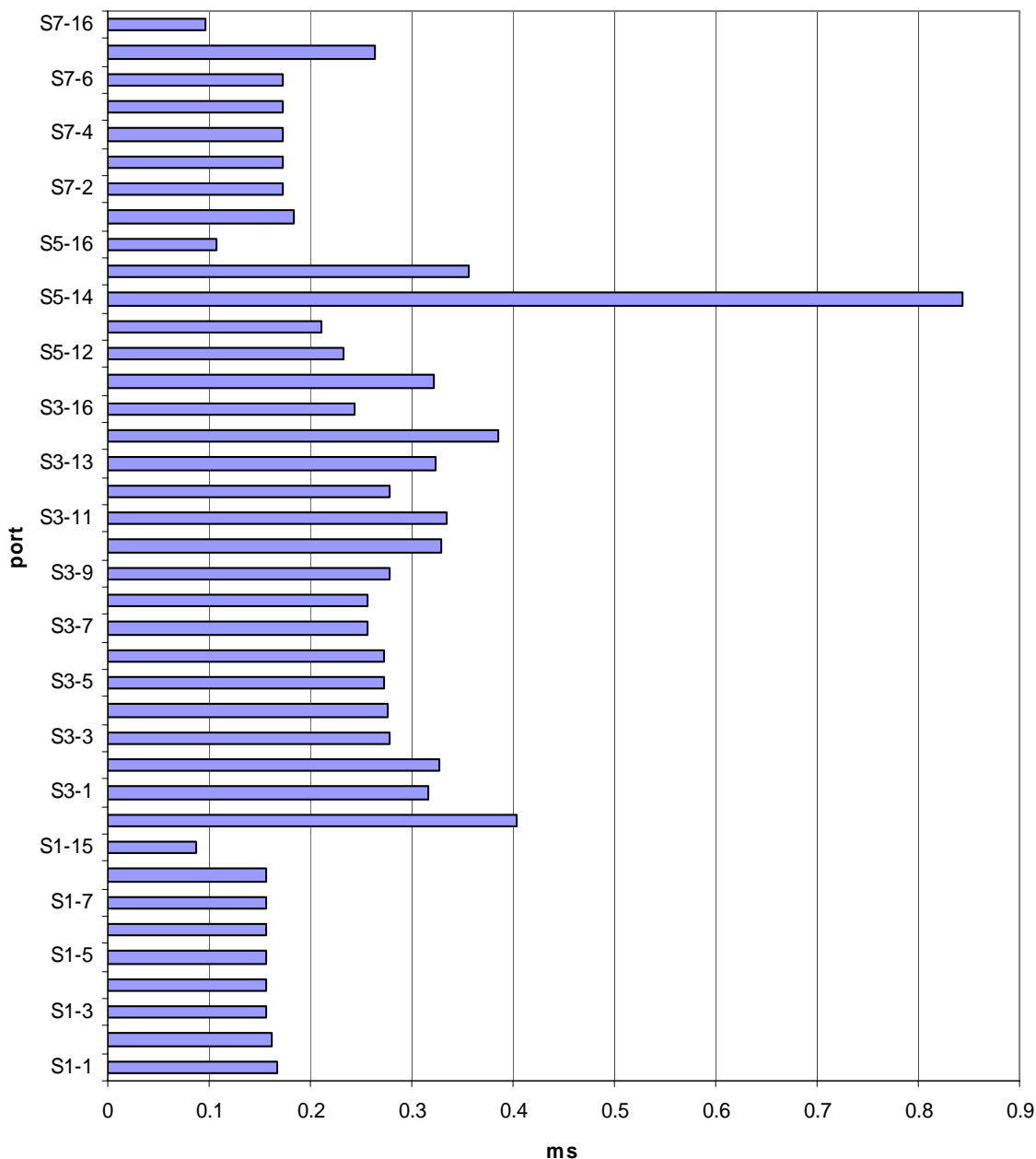


Figure 9 Commutateurs 1,3,5,7 avec politique FIFO

Ces résultats nous permettent d'effectuer plusieurs remarques. Tout d'abord, on peut constater que l'ordre de grandeur des résultats est correct, puisqu'on est bien proche de la milliseconde. On reproche souvent aux résultats issus du Network Calculus d'être exagérément pessimistes, et de ne pas permettre d'analyser réellement des réseaux. Si cette remarque est certainement justifiée dans les cas d'application du domaine d'Internet, on voit que dans notre cas

avionique, qui est très contraint, les résultats sont satisfaisants et certainement utilisables pour dimensionner le réseau.

La seconde remarque est que le réseau qui nous a été fourni satisfait à la contrainte exprimée de temps de traversée inférieur à 1 milliseconde. Cependant, il apparaît que le commutateur central, qui est selon la topologie et le routage choisis le plus chargé, possède un port qui se rapproche fortement de la contrainte.

## **4.2 Comparaison avec d'autres politiques de service**

Le réseau qui nous a été fourni possède des commutateurs dont les ports de sortie fonctionnent selon le mode FIFO. Il nous a cependant semblé intéressant de voir quel type de résultats nous aurions pu obtenir avec d'autres politiques de service. L'analyse que nous avons faite précédemment est très peu modifiée, puisqu'on ne modifie dans le modèle que le type de multiplexeur. Ceci nous impose juste de modifier la formule utilisée pour le calcul de la borne sur le délai dans les multiplexeurs.

### **4.2.1 Politique conservatrice générique**

Nous avons considéré la politique de service d'un multiplexeur générique (comme l'a défini Cruz), mais en rajoutant l'hypothèse qu'il est conservatif ( $V=0$ ). Ceci est intéressant dans la mesure où cela permet de voir quelle serait la valeur maximale du délai dans le cas d'un commutateur qui ne fonctionnerait plus normalement : il ne respecterait aucun ordre dans la retransmission des trames, tout en restant conservatif.

Pour un tel multiplexeur, et avec des enveloppes d'entrées affines, la formule 0 se réécrit :

$$D = \frac{\sum_i \sigma_i}{C - \sum_i \rho_i}$$

La figure ci-dessous présente les résultats obtenus :

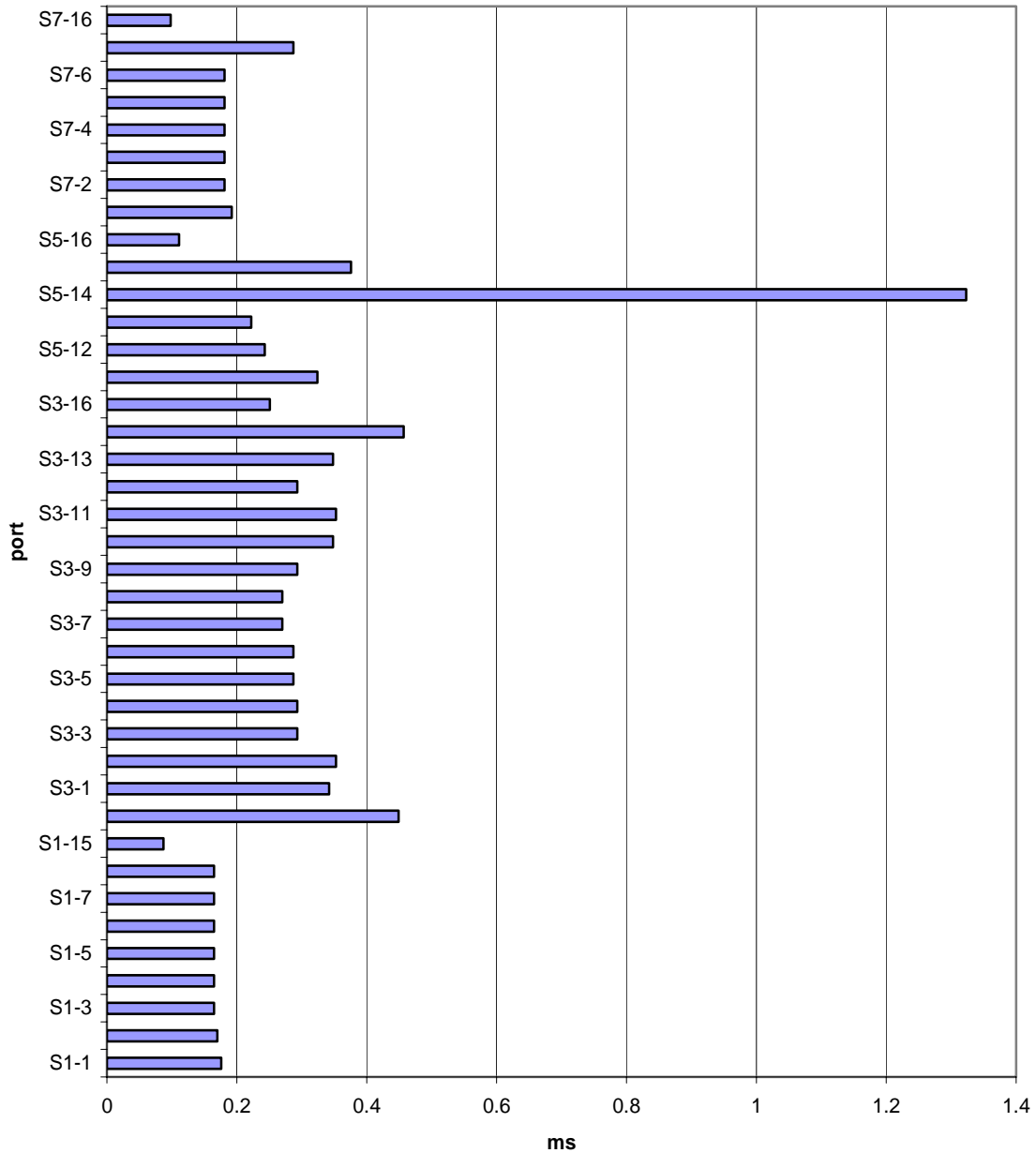


Figure 10 Commutateurs 1,3,5,7 avec GMUX

On peut naturellement constater que les résultats sont moins bons que dans le cas de la politique de service FIFO. En particulier, le port le plus chargé ne respecte plus la contrainte de la milliseconde. On comprend donc qu'il est nécessaire de posséder une connaissance suffisante de la politique de service mise en oeuvre, si on veut pouvoir obtenir de bons résultats.

#### 4.2.2 Politique à deux niveaux de priorité

Nous avons constaté que le trafic qui nous a été fourni se divisait en deux catégories : la plupart des VL ont des tailles de données relativement faibles, et des BAG égaux à 32 millisecondes. Les autres VL, moins nombreux, ont des tailles plus importantes et des BAG de quelques millisecondes. Intuitivement, ce sont ces quelques VL qui vont engorger les ports

de sortie des commutateurs. Nous avons donc décidé de voir quels résultats nous pourrions obtenir, si nous donnions la priorité aux autres flux.

Cruz ne fournit pas explicitement la valeur du délai pour ce type de multiplexeur, mais la généralisation des résultats est immédiate. On trouve que pour des enveloppes d'entrée affines, les délais pour des flux de priorité 1 (la plus haute) et 2 sont respectivement donnés par :

$$D_1 = \frac{\sum_{i \text{ de priorité } 1} \sigma_i + s_{\max,2}}{C}$$
$$D_2 = \frac{\sum_i \sigma_i}{C - \sum_{i \text{ de priorité } 1} \rho_i}$$

Avec  $s_{\max,2}$  la taille maximale pour une trame appartenant à un flux de priorité basse.

L'autre différence dans la méthode de calcul réside dans la propagation de l'enveloppe. Précédemment, on décalait l'enveloppe d'entrée de D vers la gauche. Dans le cas de deux niveaux de priorités, il convient de décaler l'enveloppe de  $D_1$  si le flux est de priorité 1, de  $D_2$  sinon. Voici une figure qui présente les résultats pour les deux niveaux de priorités, dans chaque port :



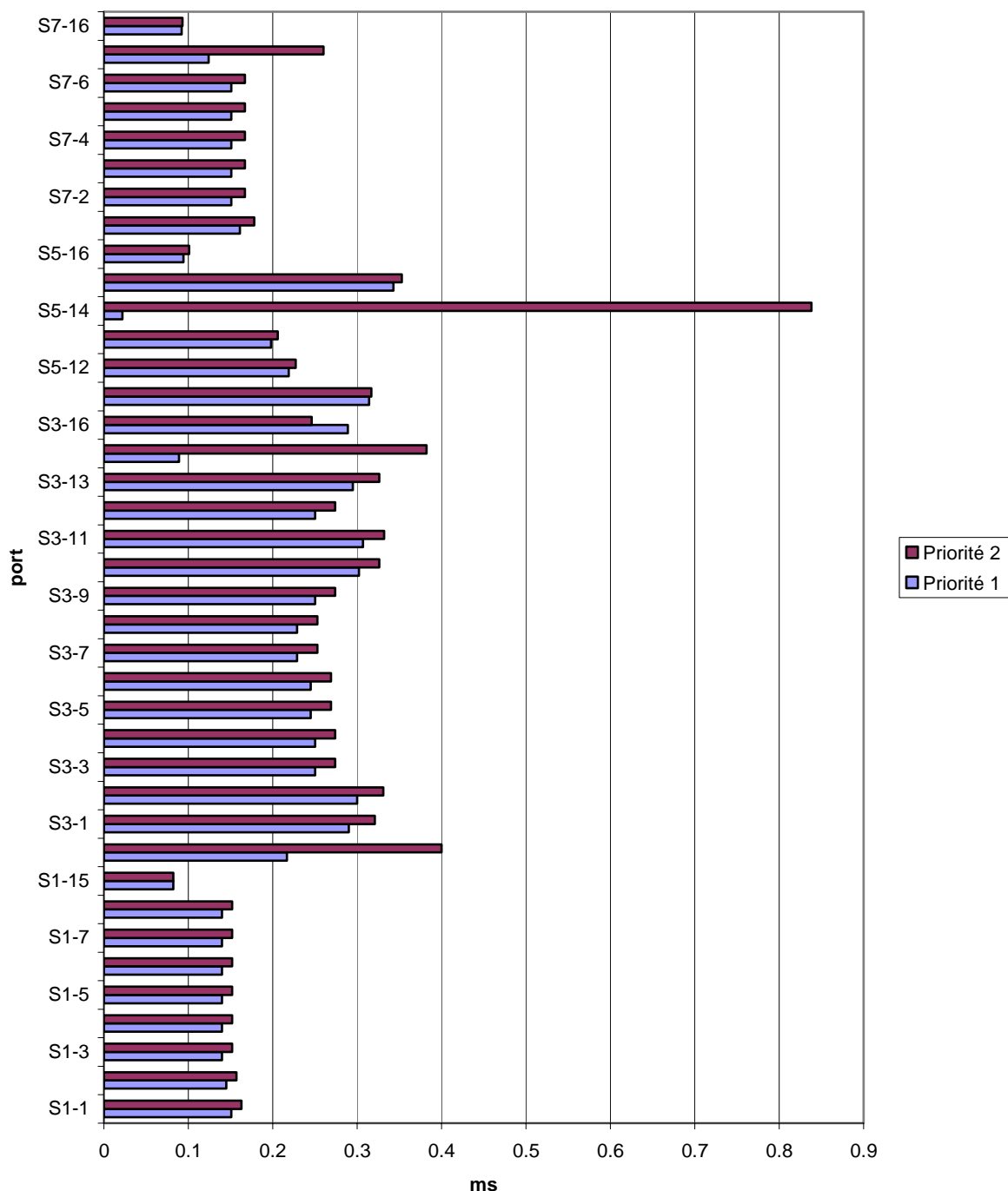


Figure 11 Commutateurs 1,3,5,7 avec double FIFO

On peut ici constater que les valeurs sont largement diminuées pour des flux de priorité haute, par rapport au traitement FIFO, ce qui est intéressant puisque ces flux sont de loin les plus nombreux. Il est particulièrement intéressant de constater que pour les flux de priorité 2, les valeurs ne sont que faiblement dégradées par rapport au cas FIFO, alors qu'on aurait pu s'attendre au contraire. On peut expliquer ce phénomène en pensant que si les flux de priorité haute subissent moins de délai, leur enveloppe en sortie du multiplexeur est moins décalée, donc elle est plus basse. Dans les éléments suivants, la contribution de ce flux au délai total sera donc moins importante, ce qui explique cette faible dégradation des résultats.

### **4.3 Interprétation physique de ces résultats**

Nous voyons d'après la formule 0 que les bornes sur les tailles de files d'attente sont directement proportionnelles à la somme des  $\sigma_i$ . Essayons d'interpréter ce résultat : le  $\sigma_i$  de chaque flux représente sa capacité à émettre instantanément des bits. Il s'agit donc directement d'une mesure de l'irrégularité des flux. On peut donc dire que les tailles des files d'attente dépendent directement de la régularité des flux qui les traversent.

D'autre part, les commutateurs centraux sont ceux qui regroupent le plus de flux, il est donc normal d'obtenir pour eux les bornes les plus élevées.

Enfin, la régularité des flux est de plus en plus dégradée au fil de la traversée du réseau, à cause des « bouchons » créés par les files d'attente. On aura donc des files d'attente de taille plus importante pour les ports situés près d'un abonné destination, par rapport aux ports situés près d'un abonné dans le sens de l'émission.

## **5 Conclusion**

L'objectif de ce chapitre était d'étudier si l'utilisation des résultats de Cruz était pertinente pour notre problématique. Nous avons d'abord montré que ces résultats s'appliquent bien dans notre contexte, notamment grâce au fait que les réseaux avioniques que nous avons étudiés sont des réseaux très contraints. Nous avons également souligné le fait que les bornes établies par Cruz nous permettent d'étudier les grandeurs qui nous intéressent, à savoir les tailles des files d'attente et les délais de bout en bout. Nous avons décrit une méthode d'analyse d'un réseau avionique basée sur les travaux de Cruz. Les résultats de cette analyse, sur un réseau de test qui nous a été fourni par Airbus, sont encourageants dans le sens où les bornes calculées sont d'un ordre de grandeur satisfaisant. Nous avons également montré que le facteur déterminant dans cette méthode d'analyse est la régularité des flux étudiés.

Cette première étude s'est donc révélée satisfaisante ; cependant les bornes obtenues nous semblent assez hautes au regard des tailles envisagées des mémoires des commutateurs, et semblent indiquer que le réseau ne pourrait absorber beaucoup plus de trafic, tout en continuant à satisfaire les spécifications des délais de traversée des commutateurs. Nous pensons toutefois que la relative simplicité de notre méthode d'analyse est la source de ce problème. Il est donc intéressant de raffiner notre méthode d'analyse, pour essayer de calculer des bornes moins pessimistes : ce sera l'objet du troisième chapitre de cette thèse.

## CHAPITRE 3

### Extensions de la méthode de preuve du déterminisme

Dans le chapitre précédent, nous avons proposé une méthode d'analyse de réseau et prouvé qu'elle était bien applicable au cas d'un réseau avionique. Les résultats d'analyse d'un prototype de réseau fourni par notre partenaire industriel ne sont cependant pas satisfaisants, dans le sens où ils indiquent que dans certains commutateurs les bornes obtenues dépassent les spécifications des contraintes sur le délai. De plus, la méthode ne permet pas d'évaluer le respect de certaines autres contraintes portant sur le nombre maximum de trames présentes dans les files d'attente. Nous nous sommes penchés sur notre méthode d'analyse, et avons essayé d'en améliorer les résultats et d'y ajouter les fonctionnalités requises. Pour ce faire, nous avons tout d'abord pensé à utiliser les dernières avancées du Network Calculus développées à la suite des travaux fondateurs de Cruz présentés dans le deuxième chapitre.

#### 1 Network Calculus généralisé

##### 1.1 Introduction de la notion de courbe de service

Les travaux de Cruz constituent la première contribution à l'ensemble de résultats connus aujourd'hui sous le nom de Network Calculus. L'intérêt principal de ces travaux est la notion d'enveloppe, qui permet de caractériser un trafic. On a cependant pu constater que la caractérisation des différents éléments reste assez rudimentaire, et difficilement généralisable. A la même époque, Parekh et Gallager [27][32] ainsi que Chang [37][40][44] se concentrent sur certains types d'éléments, et grâce à une compréhension plus précise de leur fonctionnement, établissent des bornes déterministes sur les tailles de files d'attente, ou même des bornes sur le délai de bout en bout de traversée d'un réseau.

A partir de ces travaux, l'essentiel de l'effort de recherche a porté sur la généralisation des résultats précédents, en cherchant un concept qui permette de décrire précisément le comportement des éléments réseaux, à l'instar de la notion d'enveloppe pour les flux. Cruz propose une première version de cette notion [38], qui sera appelée courbe de service. L'idée consiste en effet, comme dans le cas des enveloppes, à borner le « service » rendu durant n'importe quel intervalle temporel, c'est-à-dire la quantité de trafic servi par l'élément en fonction seulement de la longueur de l'intervalle. Cette première définition souffre de quelques inexactitudes, et ne se prête pas à une application immédiate car elle suppose de trop nombreuses hypothèses. En effet, elle repose principalement sur la notion de « période d'activité », c'est-à-dire de période telle que ses instants de début et de fin sont caractérisés par un arriéré de travail nul. Or l'existence même de telles périodes n'est pas assurée dans le cas général.

Pour combler ces lacunes, quatre équipes différentes ont proposé simultanément des définitions plus précises et équivalentes de la notion de courbe de service. Cruz tout d'abord, en collaboration avec Sariowan et Polyzos, dans une étude [39] visant à mettre au point une politique de Call Admission Control (CAC ou contrôle d'admission des connexions). Chang de son côté a également fourni une définition, dans un article utilisant le formalisme classique des systèmes linéaires [44] ; cet article présente la particularité d'utiliser un modèle en temps discret, ce qui simplifie l'expression de la définition. Agrawal Cruz Okino et Rajan ont pour leur part utilisé le concept de courbe de service pour étudier certains protocoles de contrôle de

congestion [41]. Enfin, Le Boudec[42][33] a présenté un formalisme, basé sur l'algèbre min-plus, qui permet de définir les notions de courbes de service et d'enveloppes au moyen d'un nombre limité d'opérateurs, généralisant ainsi la notion de convolution qui avait été utilisée entre autres par Agrawal. Ce formalisme étant devenu le plus répandu, nous l'avons retenu pour notre étude, et le présentons brièvement dans le paragraphe suivant.

## 1.2 Le formalisme de Le Boudec

Les travaux de Le Boudec ont pour cadre les réseaux à garantie de service, comme IntServ (Integrated Services). Dans ce cadre, les connexions ne sont acceptées que si le réseau peut leur garantir une certaine qualité de service. La manière de décrire ce service est délicate puisque chacun des nœuds du réseau peut utiliser des politiques de service très différentes ; la notion de courbe de service permet de caractériser ce service. Ce type de réseau nécessite également que les différents flux de trafic respectent un contrat de trafic, qui peut également être exprimé facilement par le formalisme mis en place.

L'idée qui a conduit à introduire ce formalisme est la volonté de se rapprocher de la théorie des systèmes, qu'on utilise par exemple pour les réseaux électriques ou le traitement du signal. Si on considère cette dernière analogie, on pourrait voir la courbe de service comme la réponse impulsionnelle du système, qui permet par un opérateur de convolution de calculer la sortie d'un système dont les entrées sont connues.

On peut également souligner le fait que le formalisme introduit par Le Boudec fait largement appel à l'algèbre min-plus, qui appartient à la catégorie des algèbres dites exotiques. Par analogie avec l'algèbre  $(\mathbb{R}, +, *)$ , dans cette algèbre l'addition est remplacée par le minimum, alors que la multiplication est remplacée par l'addition. Cette algèbre permet d'obtenir des résultats très puissants, sur l'existence de solutions optimales d'un système d'équations par exemple. Cependant, pour les utilisations les plus courantes, les opérateurs introduits peuvent être simplement considérés comme un moyen de notation simple et élégant.

Parmi les applications de ses travaux au domaine de l'Internet, Le Boudec présente le cas de la transmission d'un flux multimedia en streaming. Il explique notamment comment la théorie du Network Calculus permet de choisir la manière d'émettre le flux, la taille nécessaire du buffer de réception, le délai de retransmission, le tout lorsque la courbe de service de bout en bout du réseau est connue.

### 1.2.1 Caractérisation des éléments par la notion de courbe de service.

Définition: Soit un système  $S$  et un flux passant par  $S$  de fonction d'entrée  $R(t)$  et de sortie  $R^*(t)$ . On dit que  $S$  offre au flux une courbe de service  $\beta$  si et seulement si  $\beta$  est une fonction positive croissante telle que pour tout  $t \geq 0$ :

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

En pratique, si  $\beta$  est continue et  $R$  continue à gauche, on peut remplacer cette définition par :  
Pour tout  $t$ , il existe un  $t_0$  tel que :

$$R^*(t) \geq R(t_0) + \beta(t-t_0)$$

Cette dernière définition est plus facilement intelligible. Elle signifie qu'on peut toujours trouver un instant à partir duquel l'élément a servi le flux au moins à la vitesse représentée par  $\beta$ .

On peut citer trois exemples qui permettent d'explicitier cette notion de courbe de service. Prenons tout d'abord l'exemple d'un élément qui fait sortir les bits (quand il y en a en attente !) à un débit constant  $C$ . Ceci peut par exemple modéliser un élément dont le débit de sortie est égal à celui du lien physique qui lui est raccordé. On démontre facilement que la

courbe de service offerte par cet élément au flux est de la forme  $\lambda(t) = C.t$ . Pour cet élément, le  $t_0$  correspond à ce qu'on appelle communément période d'activité, c'est-à-dire le début d'une période où des bits sont en attente de traitement.

Le deuxième exemple est proche du précédent, puisqu'il s'agit de celui d'un élément implémentant GPS (General Processor Sharing). Dans cette politique de service, chaque flux se voit affecté un poids  $p_i$ , et on garantit à chacun des flux qu'il recevra une fraction de la capacité totale de l'élément, proportionnelle à son poids. La courbe de service offerte au flux  $i$  se déduit immédiatement de cette définition :

$$\beta_i(t) = \frac{p_i}{\sum p_i} \times C.t$$

Enfin, le dernier exemple que nous avons choisi sera très utile pour notre analyse : il s'agit de l'exemple d'un élément à délai borné  $T$ , c'est-à-dire que la seule garantie de service offerte au flux est qu'aucun de ses bits ne subira un retard de plus de  $T$  unités de temps dans cet élément. La fonction  $\delta_T$  suivante permet de caractériser un tel élément:

$$\begin{cases} \delta_T(t) = 0 & \text{pour } t < T \\ \delta_T(t) = \infty & \text{pour } t \geq T \end{cases}$$

On peut en effet facilement réécrire la définition d'une courbe de service avec cette fonction  $\delta_T$  en :  $R^*(t) \geq R(t - T)$ , ce qui est précisément le comportement recherché. On remarque que ceci permet de généraliser le résultat du Théorème 1 du chapitre précédent.

### 1.2.2 Opérateurs pour le calcul réseau

Nous présentons ici les opérateurs mathématiques présentés par Le Boudec, qui s'appliquent à des fonctions croissantes, et nulles pour des instants négatifs (causalité).

La convolution de deux fonctions  $f$  et  $g$  se définit par :

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

On peut noter que cette opération est bien l'analogue dans l'algèbre min-plus de la convolution de deux fonctions dans l'algèbre  $(R, +, *)$ . Cette opération possède entre autres les propriétés d'associativité et de commutativité. Une des autres propriétés qui sera utilisée dans la suite est que la convolution de deux fonctions convexes et affines par morceaux est convexe, et sa courbe se trouve en mettant bout à bout les différents segments de droite, rangés par ordre croissant des pentes.

Parmi les intérêts de cet opérateur, on peut citer le suivant : il est facile de définir la courbe de service cumulée de deux éléments placés successivement sur le chemin du flux.

Prenons l'exemple suivant :

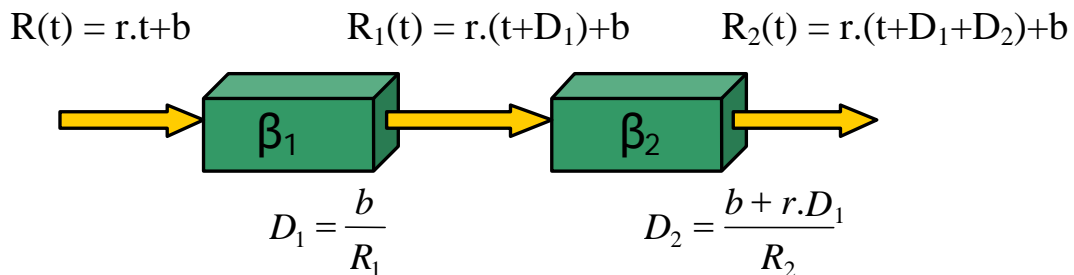


Figure 12 Deux éléments consécutifs

Dans ce schéma, nous avons représenté un flux qui passe à travers deux éléments FIFO de débit de sortie  $R_1$  et  $R_2$ , ainsi que les valeurs calculées pour les bornes sur les délais. Par définition de la courbe de service et en utilisant les propriétés de l'opérateur convolution, ce système est équivalent au suivant :

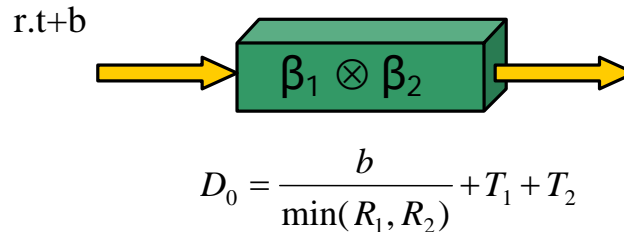


Figure 13 Courbe de service cumulée

On remarque avec ce système que la valeur de la borne sur le délai, calculée en utilisant la courbe de service concaténée, est meilleure que la somme des deux précédentes, ce qui permet d'illustrer le phénomène connu sous le nom de « pay bursts only once ».

La déconvolution est une sorte d'opération duale de la convolution ; on la définit comme suit :

$$\alpha \oslash \beta (t) = \sup_{u \geq 0} \{ \alpha(t+u) - \beta(u) \}$$

Dans la suite de cette thèse, nous utiliserons ces opérateurs comme des notations plutôt que comme des outils permettant d'obtenir des résultats avancés. Par exemple, on peut réécrire le fait que la fonction  $R^*$  est enveloppe d'un flux dont le trafic cumulé est noté par  $R(t)$  en :

$$R \leq R \otimes R^*$$

De même, on peut dire que  $\beta$  est une courbe de service offerte à ce flux si et seulement si :

$$R_{i+1} \geq R_i \otimes \beta$$

où  $R_{i+1}$  désigne le trafic cumulé en sortie de cet élément.

### 1.2.3 Reformulation des théorèmes fondamentaux du calcul réseau

Le Boudec redémontre trois théorèmes très importants du Network Calculus, en utilisant les notations précédentes.

- Borne sur le délai

Soit un flux de courbe d'arrivée  $\alpha$ , qui traverse un système  $S$  lui offrant une courbe de service  $\beta$ . Pour tout  $t$ , le délai virtuel  $d(t)$  est tel que :

$$d(t) \leq h(\alpha, \beta)$$

Où  $h(\alpha, \beta)$  représente la déviation horizontale entre les deux courbes  $\alpha$  et  $\beta$  :

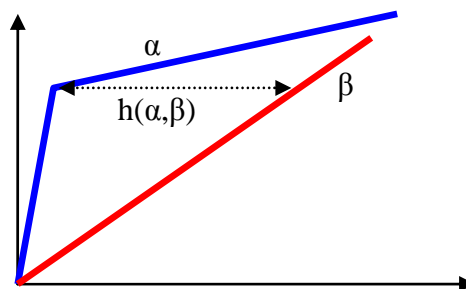


Figure 14 Déviation horizontale entre deux courbes

- Borne sur l'arriéré de travail

Soit un flux de courbe d'arrivée  $\alpha$ , qui traverse un système S lui offrant une courbe de service  $\beta$ .

L'arriéré de travail  $R(t) - R_{i+1}(t)$  vérifie pour tout t :

$$R_i(t) - R_{i+1}(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\}$$

En d'autres termes, cet arriéré est majoré par la déviation verticale maximale entre les deux courbes.

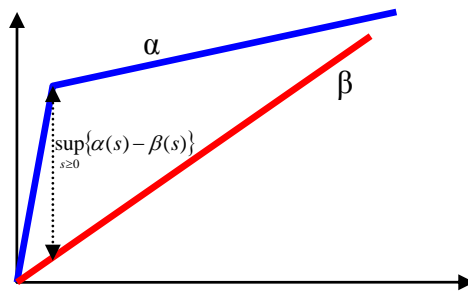


Figure 15 Déviation verticale entre deux courbes

- Caractérisation de la sortie d'un élément

Soit un flux de courbe d'arrivée  $\alpha$ , qui traverse un système S offrant une courbe de service  $\beta$ .

Le flux de sortie de S a comme courbe d'arrivée la courbe  $\alpha^*$ , définie par :

$$\alpha^* = \alpha \oslash \beta$$

Cas particulier :

Dans le cas où l'élément offre une courbe de service de type délai borné,  $\beta(t) = \delta_D$ , on peut facilement calculer une courbe d'arrivée du flux de sortie grâce au théorème précédent, et aux propriétés des opérateurs  $\oslash$  et  $\delta$ . En effet, dans ce cas particulier, la courbe  $\alpha^*$  n'est autre que la courbe  $\alpha$  décalée de D vers la gauche :

$$\alpha^*(t) = \alpha(t + D)$$

On retrouve bien dans cette expression le fonctionnement d'un élément à délai borné décrit par Cruz.

#### 1.2.4 Courbe de service maximale

On connaît parfois des informations supplémentaires sur le fonctionnement des éléments, qu'on aimerait modéliser par le Network Calculus. Le cas le plus fréquent est celui d'un élément au sein duquel existe un délai de traversée minimum, et un débit de sortie maximum. Pour modéliser ce type d'éléments, on définit une notion de courbe de service maximale, qui sera analogue à la courbe de service (minimale) précédente.

Définition: Soit un système S et un flux passant par S de fonction d'entrée  $R(t)$  et de sortie  $R^*(t)$ . On dit que S offre au flux une courbe de service maximale  $\gamma$ , fonction positive croissante nulle pour les instants négatifs, si et seulement si  $\gamma$  est telle que pour tout  $t \geq 0$  et tout  $s \leq t$ :

$$R^*(t) \leq R(s) + \gamma(t - s)$$

$$\text{soit : } R^* \leq R \otimes \gamma$$

Cette notion de courbe de service maximale permet dans certains cas d'améliorer la borne sur le délai. Elle permet également d'améliorer la connaissance du flux en sortie de l'élément, par le théorème suivant :

Soit un flux de courbe d'arrivée  $\alpha$  qui traverse un système qui lui offre une courbe de service  $\beta$  et une courbe de service maximale  $\gamma$ , alors le flux en sortie de cet élément a comme enveloppe :

$$\alpha^* = (\alpha \otimes \gamma) \oslash \beta$$

Comme la convolution de  $\alpha$  et  $\gamma$  possède la propriété d'être inférieure en tout point aux deux fonctions, cette enveloppe ne peut qu'être meilleure que celle donnée seulement par la notion de courbe de service,  $\alpha \oslash \beta$ .

### 1.3 Application au réseau étudié

Nous avons utilisé le formalisme décrit ci-dessus pour réécrire la méthode d'analyse présentée dans le deuxième chapitre.

#### 1.3.1 Description des flux

Nous avons vu précédemment que la notion d'enveloppe ne change pas dans ce nouveau formalisme. Comme dans la méthode précédente, on utilise donc pour modéliser un VL un arbre de diffusion multicast, formé de segments de flux qui sont disposés dans chaque élément avant tout calcul, et conformément à la description du réseau fournie. L'enveloppe initiale choisie pour le flux qui est à la racine de l'arbre est fixée à :

$$\sigma = S_{\max}$$

$$\rho = \frac{S_{\max}}{BAG}$$

#### 1.3.2 Description des éléments

La description choisie utilise la notion de courbe de service décrite précédemment.

Ainsi, on modélise un élément à délai borné par la courbe de service  $\delta_D$ , notamment le traitement des trames dans le commutateur.

Pour les multiplexeurs, on choisit de les modéliser par des droites  $\lambda(t) = C.t$ , où  $C$  vaut 10 ou 100 Mbps, selon les capacités des ports de sortie des commutateurs ou des End System. Il faut toutefois noter que cette courbe de service est offerte au flux total entrant, et non individuellement à chaque flux. Pour des raisons explicitées dans le paragraphe suivant, on va dans un deuxième temps modéliser les multiplexeurs par un élément à délai borné.

#### 1.3.3 Méthode de calcul

On procède comme précédemment à un calcul par propagation des enveloppes de proche en proche.

Pour calculer les valeurs des bornes sur les délais et les tailles de files d'attente, on commence par calculer l'enveloppe totale entrant dans l'élément considéré (ce qui est équivalent à trouver l'enveloppe du flux agrégé entrant), en additionnant toutes les enveloppes des flux. Ensuite, les bornes sont obtenues en calculant la déviation horizontale et verticale entre cette enveloppe totale et la courbe de service de l'élément.

Pour les éléments à délai borné, on applique le troisième théorème cité ci-dessus pour connaître une enveloppe de chaque flux en sortie. Avec la courbe de service choisie, on a déjà remarqué que cela revient à décaler l'enveloppe d'entrée de la valeur du délai vers la gauche. Dans le cas des multiplexeurs, il est difficile de calculer la déconvolution de l'enveloppe des



flux avec une courbe de service linéaire. De plus, cette courbe de service n'est pas offerte individuellement à chacun des flux. Utiliser le troisième théorème permettrait donc seulement de calculer (et c'est difficile) l'enveloppe en sortie du multiplexeur pour le flux total. Pour un flux individuel, on sait seulement que la borne sur le délai, calculée en prenant en compte l'enveloppe du flux total et la courbe de service linéaire, reste valable : si aucun bit du flux total ne reste plus de  $D$  dans le multiplexeur, alors ce sera en particulier le cas pour chacun des flux. On décide donc de considérer que les multiplexeurs se comportent comme des éléments à délai borné  $D$ , vis-à-vis de chacun des flux individuels. L'enveloppe en sortie de chaque flux est alors obtenue avec l'application du troisième théorème pour une courbe de service  $\delta_D$ , c'est-à-dire en décalant l'enveloppe de  $D$  vers la gauche.

Cette méthode de calcul n'utilise pas la notion de courbe de service cumulée que nous avons présentée précédemment. En effet, dans ce contexte il est très difficile de calculer la courbe de service individuelle offerte à un flux par un élément donné. On connaît principalement la courbe offerte à l'ensemble des flux. Sans la connaissance de chaque courbe de service individuelle, il n'est pas possible de calculer les courbes de services cumulées de bout en bout.

## 2 Proposition de la notion de groupe

### 2.1 Motivation

L'analyse effectuée dans le deuxième chapitre de cette thèse établit des bornes valides dans toutes les configurations des temps d'émission possibles. Pour calculer l'enveloppe globale des flux entrant dans un élément, on additionne les courbes d'arrivées de tous les flux. Intéressons-nous à la signification physique de cette addition, et en particulier à la valeur de cette somme calculée en 0. Pour un flux individuel, la valeur de la courbe d'arrivée en 0 représente la quantité maximale de bits que ce flux peut émettre instantanément. Faire la somme pour tous les flux revient à borner la quantité maximale de bits que peut émettre le flux agrégé, par la somme de ces quantités pour chaque flux. De manière moins précise mais plus imagée, on exprime ainsi le fait que tous les flux peuvent émettre une trame « en même temps ». Or il nous est apparu qu'en pratique, certains flux ne pouvaient pas émettre en même temps, notamment ceux qui ont parcouru un même lien physique ensemble : pour eux, les trames doivent être sérialisées, et les arrivées de trames ne peuvent être simultanées.

On peut illustrer ceci sur un exemple simple.

Prenons le cas de deux files d'attente FIFO, reliées par un simple lien physique de capacité  $C$ . En entrée de la première file, on place deux flux caractérisés par une enveloppe affine notée  $(\sigma_i, \rho_i)$ .

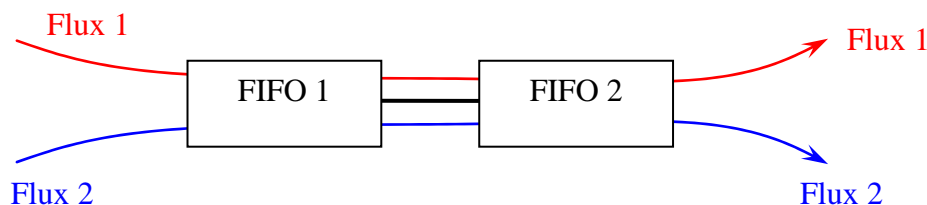


Figure 16 Exemple de l'utilité des groupes

Le calcul réseau nous donne les valeurs des délais dans les deux files d'attente,  $D_1$  et  $D_2$ .

$$D_1 = \frac{\sigma_1 + \sigma_2}{C}; D_2 = \frac{\sigma_1 + \sigma_2 + D_1(\rho_1 + \rho_2)}{C}$$

Or pour le cas considéré, sachant que le débit en sortie des deux éléments est C, les bits ne devraient pas être retardés dans la deuxième file d'attente, puisqu'ils ont déjà été ordonnés par la première ; on devrait avoir  $D_2 = 0$ .

On réalise ici que le calcul effectué jusqu'à présent n'utilisait pas cet effet de sérialisation des trames sur les câbles, que nous avons modélisés par un délai constant. Nous avons donc décidé d'introduire un concept permettant de prendre en compte le fait que différents flux ont déjà traversé des éléments communs, et donc qu'il existe des relations entre eux : la notion de groupe de flux.

## 2.2 Caractérisation des groupes

### 2.2.1 Formation d'un groupe

On définit un groupe de flux comme étant l'ensemble des groupes sortant d'un même multiplexeur plus les flux sortant du même multiplexeur et qui ne se trouvent pas représentés dans un de ces groupes (on ne considère pas les groupes qui sont entièrement inclus dans un autre).

On constate que la définition des groupes est purement locale, puisque liée à un multiplexeur : deux flux peuvent être dans un même groupe pour un commutateur donné, et non dans le commutateur suivant. On constate également que ce ne sont pas les VL (flux multicast) qui font partie des groupes, mais les segments de flux, qui vont d'un élément à un autre.

Cette définition, dans le cadre du modèle considéré, est équivalente à dire qu'on considère que les groupes sont formés des flux circulant sur un même lien physique entre deux commutateurs. En effet, nous avons modélisé les ports de sortie des commutateurs comme étant des multiplexeurs FIFO de capacité de sortie le débit du lien, donc les deux définitions sont équivalentes. Ainsi, ce concept de groupe concerne bien les flux qui subissent l'effet de sérialisation décrit précédemment.

Il est possible lors de la création d'un groupe dans un multiplexeur de constater que plusieurs flux faisaient déjà partie d'un même groupe dans un multiplexeur amont, donc ils ont déjà été sérialisés. Pour prendre en compte cette information, on utilise lors de la création d'un nouveau groupe les groupes déjà existants. La condition portant sur le cardinal de l'ensemble signifie alors simplement qu'on utilise pour cela les groupes les plus importants (en nombre de flux contenus). Ceci est notamment utile pour le cas où l'un des groupes entrant est lui-même constitué de plusieurs groupes.

### 2.2.2 Enveloppe de groupe

Nous avons vu dans l'exemple précédent que la prise en compte de l'effet de sérialisation doit donner des valeurs de bornes bien plus faibles. Pour ce faire, il convient de décrire précisément le trafic généré par un groupe, en définissant sa courbe d'arrivée, dont nous allons essayer de donner une interprétation physique.

Tout d'abord, en notant  $R_g(t)$  l'enveloppe du groupe, et  $R_i(t)$  celles des flux, on doit avoir de manière immédiate :

$$R_g(t) \leq \sum_i R_i(t)$$

En effet, il est évident que le groupe de flux ne peut émettre plus de trafic pendant un intervalle de durée t que la somme de ce que chacun des flux peut émettre.

D'autre part, on sait qu'aucun des flux individuels ne peut émettre plus vite que la capacité de sortie du multiplexeur (notée C) puisque celle-ci représente le débit maximal du lien physique

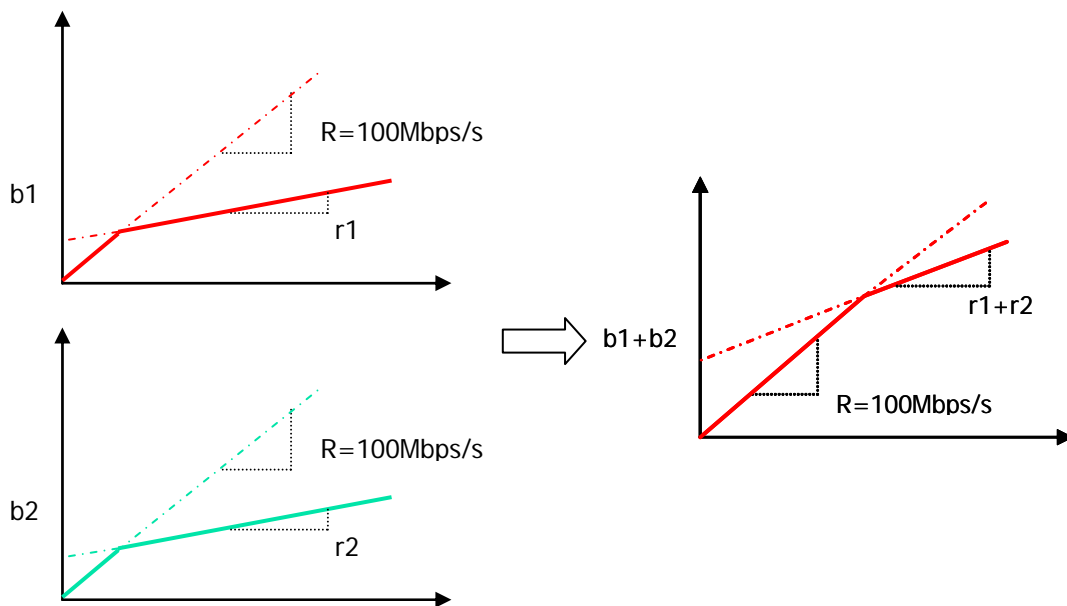
auquel est rattaché le multiplexeur. Les flux se partageant le lien physique, le groupe des flux ne peut à son tour excéder cette limite. En terme d'enveloppe, on doit donc avoir :

$$R_g(t) \leq C.t$$

L'enveloppe du groupe satisfait ces deux conditions, elle doit donc en tout point satisfaire le minimum des deux, ce que l'on peut réécrire en :

$$\forall t, R_g(t) \leq \min \left\{ \sum_i R_i(t), C.t \right\}$$

On représente cette enveloppe dans la figure suivante, dans le cas d'un groupe formé de deux flux, avec une capacité C de 100 Mbps.



**Figure 17 Modifications des courbes d'arrivée par les groupes**

Si on reprend l'exemple du paragraphe précédent, avec la courbe d'arrivée du groupe définie comme ci-dessus, on peut immédiatement vérifier qu'on obtient bien un délai nul dans la file d'attente FIFO 2.

### 2.2.3 Lien avec la notion de courbe de service maximale

La courbe d'arrivée précédente possède l'avantage de satisfaire notre compréhension physique du comportement du réseau. Il est également intéressant de noter que le concept de groupe peut facilement être pris en compte dans le Network Calculus.

En effet, au lieu de modéliser le lien physique par un élément de type délai constant, on peut utiliser un élément plus complexe, un élément à courbe de service maximale. Cette notion de courbe de service maximale permet de spécifier le service maximum rendu par un élément ; dans le cas d'un lien physique de capacité C, le lien ne peut émettre plus de C bits par seconde. On peut donc le modéliser par un élément de courbe de service maximale  $\lambda(t) = C.t$ . On sait d'autre part que le lien physique offre également la courbe de service classique (c'est à dire le service garanti par le lien)  $\lambda(t)$ . Ce service est offert au flux agrégé sortant du multiplexeur ; en appliquant le Network Calculus, on sait que l'enveloppe de ce flux agrégé en sortie du lien est inférieure à :

$$R'(t) = \left( \lambda(t) \otimes \sum_i R_i(t) \right) \oslash \lambda(t)$$

On peut maintenant utiliser le fait que la convolution de deux fonctions linéaires par morceaux et concaves est obtenue en mettant bout à bout les différents morceaux, par ordre décroissant des pentes. En appliquant cette démarche, on constate que l'enveloppe de groupe définie précédemment est exactement égale à l'enveloppe en sortie d'un lien physique modélisé par une courbe de service  $\lambda(t)$  et une courbe de service maximale  $\lambda(t)$ .

Par la suite, nous avons décidé d'utiliser cette notion de groupe, plutôt que de rendre plus complexe le modèle des liens. Néanmoins, la discussion précédente justifie que notre choix ne nous fait pas sortir du contexte du Network Calculus.

### **2.3 Intégration de la notion de groupe dans le calcul**

Nous avons défini dans le paragraphe précédent la notion de groupe, en précisant bien qu'il s'agissait d'une notion propre à un multiplexeur donné (donc locale). Nous étudions maintenant la façon d'utiliser ces groupes dans l'analyse d'un réseau complet.

#### **2.3.1 Propagation de l'enveloppe d'un groupe**

Le comportement des différents éléments réseaux (à part le multiplexeur), comme les éléments à délai constant ou à délai borné, ne dépend pas du nombre de flux qui les traversent. Dès lors, il est immédiat de constater qu'ils affectent tous les flux de la même manière, donc le flux agrégé qui les traverse également. Ainsi, il est possible de propager les enveloppes des groupes au travers de ces éléments, en considérant qu'elles sont modifiées de la même façon que celles des flux individuels. Par exemple, au passage dans un élément de délai borné  $D$ , l'enveloppe du groupe sera décalée de  $D$  vers la gauche.

#### **2.3.2 Fin de la propagation**

Il est évident que la notion de groupe permet d'obtenir des enveloppes plus faibles, donc plus intéressantes pour le calcul. On cherche donc à conserver cette information le plus longtemps possible. Cependant, si les flux composant le groupe se séparent, on ne peut plus conserver le groupe. L'idéal serait de déterminer, au moment de la séparation, quelle est la contribution de chacun des flux à l'enveloppe du groupe. Malheureusement, il n'existe pas actuellement de résultat théorique sur cette séparation. Dans notre étude, nous avons donc décidé de faire l'analyse en utilisant à la fois les groupes et les flux individuels, de manière à pouvoir retrouver l'enveloppe de chacun des flux lorsqu'un groupe se sépare. L'analyse est donc effectuée simultanément sur les groupes et les flux. Le paragraphe suivant précise quels groupes et flux sont utilisés lors des calculs de délai dans un multiplexeur, le suivant présentera en pratique le mode de calcul utilisé.

#### **2.3.3 Multiplexeurs et groupes**

Lors du calcul des délais et des tailles de files d'attente dans les multiplexeurs, il faut déterminer l'enveloppe globale entrante. Nous avons choisi d'utiliser pour ce calcul le plus de groupes possible, afin de diminuer au maximum l'enveloppe entrante. Celle-ci est donc obtenue en additionnant les enveloppes des groupes (non inclus dans des groupes plus grands), puis en additionnant les enveloppes individuelles des flux restants, c'est-à-dire ceux qui ne faisaient partie d'aucun groupe.

De la sorte, on prend en compte les contributions de chacun des flux à l'enveloppe entrante, mais en utilisant le plus possible la notion de groupe.

### 2.3.4 Pratique du calcul

La première étape du calcul est de former l'ensemble des groupes. Comme indiqué ci-dessus, on forme les groupes dans un multiplexeur donné soit à partir de groupes formés précédemment, soit à partir de flux individuels.

Pour tous les éléments sauf les multiplexeurs, on effectue les modifications d'enveloppe de tous les flux et les groupes, en particulier le décalage vers la gauche pour les éléments de type délai borné.

Pour les multiplexeurs, on effectue les calculs des bornes de délai  $D$  et des tailles de files d'attente  $B$ , en utilisant comme enveloppe du flux total entrant l'enveloppe présentée dans le paragraphe précédent 2.3.3. On précise ensuite comment sont obtenues les enveloppes des flux et groupes sortant d'un multiplexeur :

- Pour les groupes formés en amont du multiplexeur considéré, l'enveloppe de sortie est celle d'entrée décalée de  $D$  vers la gauche, puisqu'on considère les multiplexeurs comme des éléments à délai borné  $D$ , de courbe de service  $\delta_D$  1.3.2).
- Pour tous les flux entrant dans ce multiplexeur, l'enveloppe de sortie est celle d'entrée décalée de  $D$  vers la gauche pour la même raison.
- Pour les groupes créés dans le multiplexeur considéré, l'enveloppe est construite comme indiqué dans le paragraphe 2.2.2, à partir des enveloppes décrites dans les deux points précédents.

## 2.4 Gains apportés

Nous avons appliqué cette nouvelle méthode d'analyse sur le prototype de réseau décrit dans le chapitre précédent. La figure ci-dessous représente le nombre de groupes pris en compte pour le calcul des bornes dans chacun des multiplexeurs, ainsi que le nombre de flux entrants.

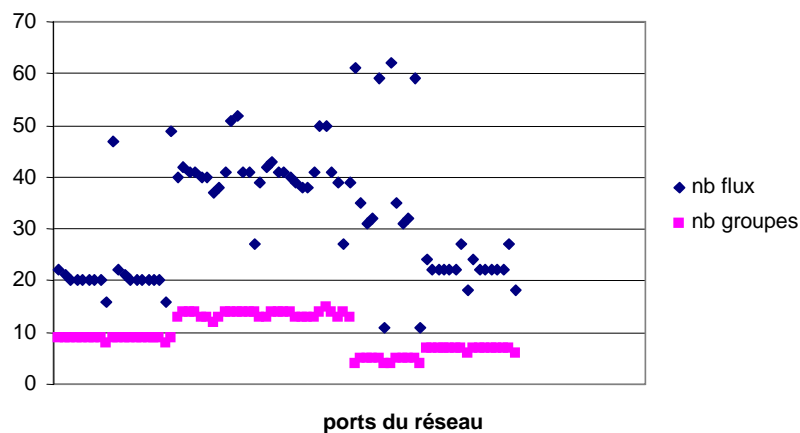


Figure 18 Nombre de flux et de groupes utilisés par le calcul dans chaque port

On peut constater tout d'abord qu'on n'utilise jamais plus de 16 groupes pour les calculs. Ceci est tout à fait normal, puisque dans ce réseau chacun des commutateurs ne dispose que de 16 ports d'entrée-sortie, et puisqu'on crée un groupe par lien physique. Au maximum, on ne peut donc jamais avoir plus de 16 groupes en sortie d'un port donné, car les flux sortant d'un port sont forcément entrés dans le commutateur sur un des 15 autres ports (15 groupes au plus), ou ont été générés dans le commutateur lui-même (1 groupe).

L'autre constatation est que le nombre de groupes utilisés est bien plus faible que le nombre de flux présents, ce qui implique que l'enveloppe utilisée dans les calculs des multiplexeurs va être bien meilleure que dans l'analyse précédente. Dans le second chapitre de cette thèse, nous avons montré que ce qui influence le plus le résultat du calcul est la taille maximale de rafale autorisée par l'enveloppe totale d'entrée. L'intérêt des groupes est de largement

diminuer cette taille de rafale admissible (pour des flux du même groupe, il n'est plus possible d'émettre des trames simultanément). On s'attend donc à ce que les résultats de la nouvelle analyse soient bien meilleurs que précédemment. Nous présentons dans le graphique suivant les gains obtenus grâce à l'utilisation conjointe de la notion de groupe et du formalisme plus évolué présenté en 1.2.

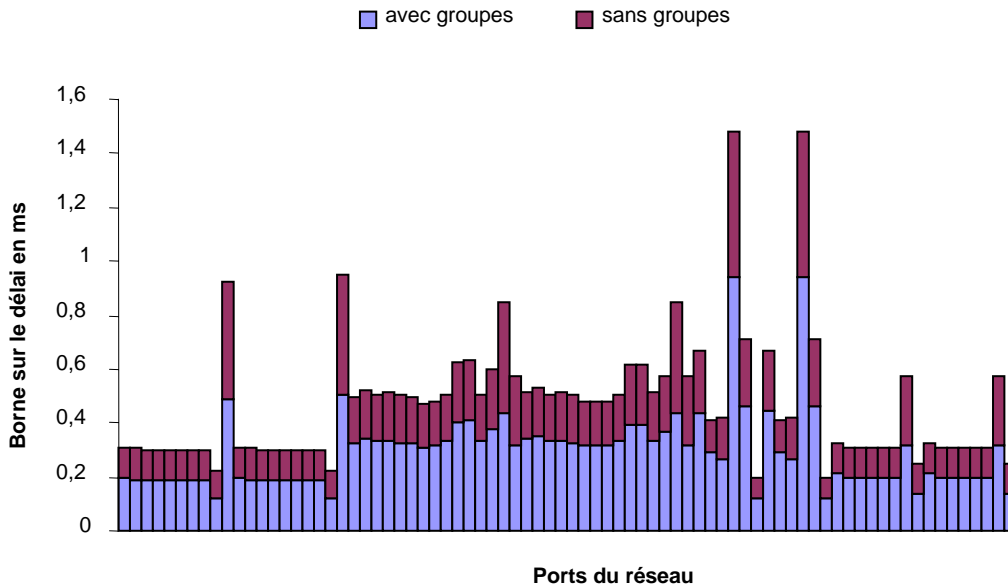


Figure 19 Comparaison analyse avec ou sans groupes

Chacune des barres du graphique représente la valeur calculée de la borne sur le délai dans un port de sortie d'un commutateur du réseau.

On constate immédiatement un gain de l'ordre de 40%, ce qui confirme bien l'intuition précédente que ce mode de calcul donne de bien meilleurs résultats en prenant en compte l'effet de sérialisation des flux.

### 3 Analyse en trames

Jusqu'à présent, nos analyses ont utilisé comme unité les bits et les secondes. En particulier, les valeurs des courbes d'arrivée et de service représentaient un nombre de bits en fonction du temps. Cependant, il peut être intéressant d'obtenir des résultats exprimés en nombre de trames, et non en nombre de bits. Prenons l'exemple suivant : dans un commutateur, les trames sont stockées au fur et à mesure de leur arrivée dans une zone de mémoire. Par la suite, les différentes opérations de relais et d'aiguillage internes au commutateur s'effectuent non pas sur des trames complètes, mais sur des pointeurs vers l'adresse mémoire à laquelle la trame a été stockée. Supposons qu'une de ces opérations nécessite de placer ces pointeurs dans une file d'attente de taille limitée. Si on cherche à vérifier que cette file d'attente ne perd pas de trames par congestion, ce n'est pas la taille totale en bits des différentes trames représentées par les pointeurs qui compte, mais bien le nombre de trames en attente dans cette file. On comprend donc aisément que dans certains cas on cherche des résultats exprimés en nombre de trames.

Pour ce faire, nous avons essayé de mener en parallèle à l'analyse classique une analyse « dans le domaine des trames ».

Note : Dans la suite de cette thèse, les résultats présentés ont été obtenus avec un nouveau réseau de test, qui est d'architecture globalement similaire, mais possède un trafic plus important.

### 3.1 Mode de calcul en trames

#### 3.1.1 Enveloppes en trames

A l'instar de ce qui existe dans le domaine des bits, on va définir une enveloppe en trames  $R_f^*(t)$  comme étant le nombre maximal de trames que peut émettre un flux durant un intervalle temporel quelconque d'amplitude  $t$ .

Comme dans le domaine des bits, les deux principales questions sont les suivantes : quelle est l'enveloppe initiale en trames ? Comment sont modifiées ces enveloppes en trames au passage d'un élément réseau ? Pour répondre à la première question, nous décrivons ci-dessous les enveloppes en trames, sous deux jeux d'hypothèses différents. Le premier reprend les hypothèses utilisées dans cette thèse, alors que le second jeu d'hypothèses correspond exactement à la définition des VL telle que donnée dans l'annexe 1 de la partie 7 de la norme ARINC 664.

##### 3.1.1.1 Premier jeu d'hypothèses

Dans ce premier cas, on se place dans le cadre précédemment étudié d'un modèle par paquets. On reprend la définition précédente des VL, dont on rappelle le point qui concerne la cadence maximale d'émission: un VL émet les trames instantanément (les trames sont générées à l'intérieur du End System, avant le multiplexeur de sortie), mais deux trames ne peuvent être séparées de moins de BAG unités de temps.

La figure ci-dessous représente une telle émission à partir d'un instant  $t_0$ .

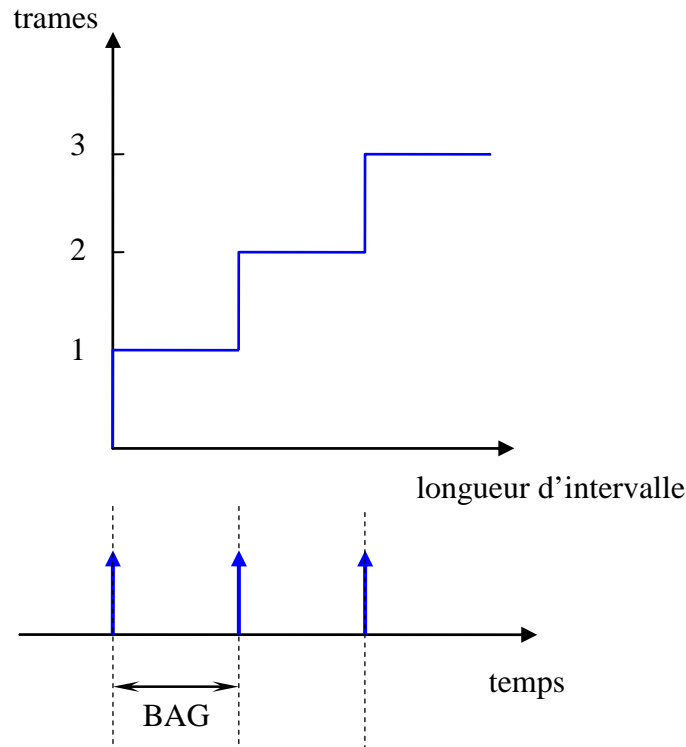


Figure 20 Schéma d'arrivée des trames

Comme nous l'avons représenté sur la figure, il est très facile de déterminer une enveloppe affine, en nombre de trames, pour un VL ayant ces caractéristiques. On a :

$$\sigma = 1 ; \rho = \frac{1}{BAG}$$

Ce qui signifie exactement que le VL peut émettre instantanément une trame, puis ne peut émettre qu'au débit maximal d'une trame toutes les BAG unités de temps.

Dans la suite de notre étude, nous nous placerons dans ce cadre précis. Cependant, pour mieux illustrer la notion d'enveloppe en trames, ainsi que l'importance de la définition du BAG, nous avons déterminé l'enveloppe initiale sous un jeu d'hypothèses différent. Le paragraphe suivant est donc à lire comme une simple illustration des difficultés de l'analyse dans le domaine des trames.

### 3.1.1.2 Second jeu d'hypothèses

Les hypothèses considérées ici sont les suivantes :

- Le VL émet ses trames sur un lien de capacité finie (plus de transmission instantanée)
- Le temps minimum entre le début de deux trames consécutives est BAG
- La taille des trames est comprise entre  $S_{\min}$  et  $S_{\max}$ .

De plus, on suppose que le temps d'émission d'une trame de taille  $S_{\max}$  est inférieur à BAG, pour des raisons de stabilité du modèle.

On rappelle que la convention adoptée dans cette thèse, comme dans la littérature, est qu'une trame est comptée transmise lorsque tous ses bits ont été transmis ou (c'est équivalent dans le cadre d'un élément FIFO), lorsque le dernier bit de la trame a été transmis. Le schéma suivant récapitule ces hypothèses (il ne représente qu'un seul VL, avant qu'il ne soit multiplexé avec d'autres).

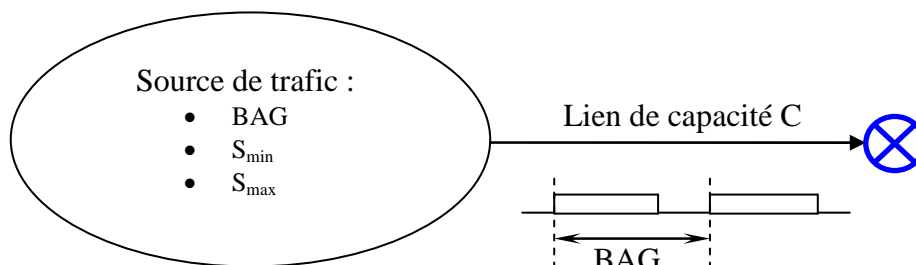


Figure 21 Second jeu d'hypothèses

On essaie de déterminer une courbe d'arrivée affine pour le VL au point marqué d'une croix, compte tenu de la convention précédente.

Dans un premier temps, il est immédiat de trouver le pire schéma d'émission pour ce VL, c'est-à-dire le schéma tel que le trafic cumulé (toujours en considérant la convention, et donc en se référant au dernier bit des trames) sera le plus important. Il s'agit d'une trame de taille  $S_{\max}$ , suivie uniquement de trames de taille  $S_{\min}$ . C'est en effet avec ce schéma que les derniers bits de deux trames consécutives seront les plus proches possible.

Le schéma ci-dessous représente cette émission à partir d'un instant de référence  $t_0$ .



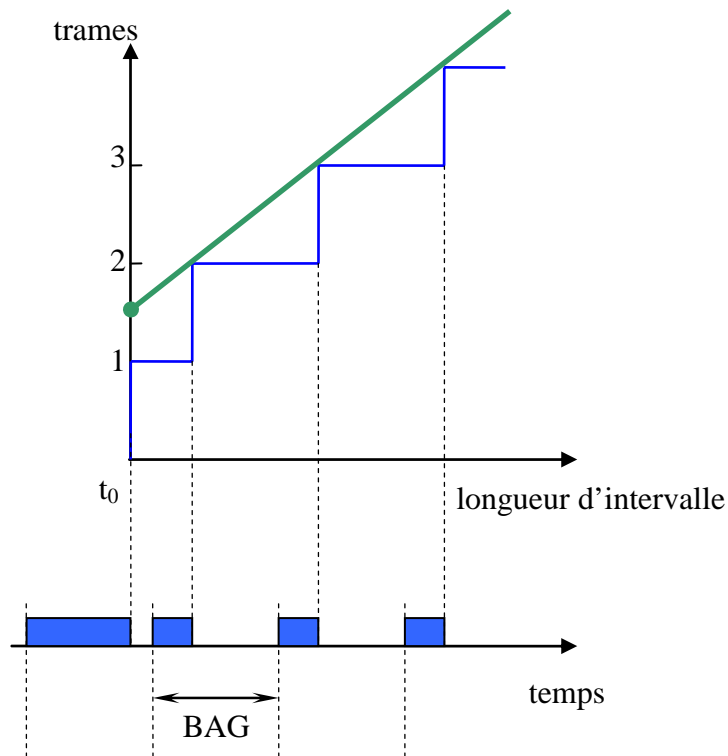


Figure 22 Schéma d'arrivée pire cas en termes de trames

Nous donnons ci-dessous la valeur de l'enveloppe (en trame) affine de plus faible pente :

$$\sigma = 1 + \frac{1}{BAG} \frac{S_{\max} - S_{\min}}{C} ; \rho = \frac{1}{BAG}$$

Bien sûr, si on cherchait l'enveloppe en bits, ce schéma d'émission ne serait pas le pire. Ce serait alors l'envoi par le VL de trames de taille maximale tous les BAG. Pour information, voici l'enveloppe affine de plus faible pente que l'on pourrait avoir en bits sous ce jeu d'hypothèses.

$$\sigma = S_{\max} \left( 1 - \frac{S_{\max}}{C \cdot BAG} \right) ; \rho = \frac{S_{\max}}{BAG}$$

Cette équation est celle de la droite représentée ci-dessous.

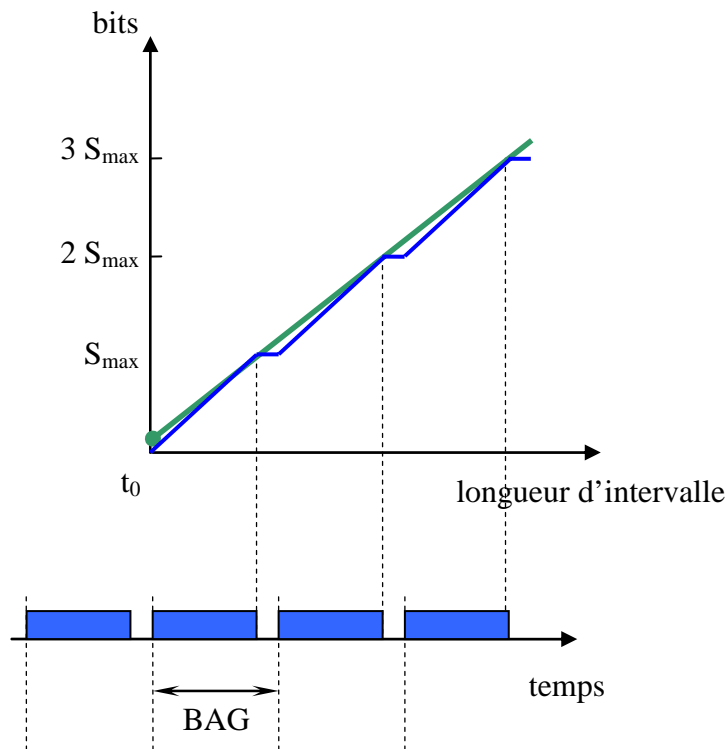


Figure 23 Schéma d'arrivée pire cas en termes de bits

### 3.1.2 Courbe de service en trames

De la même façon que précédemment, il est nécessaire de définir une courbe de service en termes de trames pour caractériser les éléments. La courbe de service  $\beta_f(t)$  d'un élément représente donc le nombre minimum de trames que peut traiter un élément durant un intervalle temporel d'amplitude  $t$ .

Pour les éléments de type délai borné, la courbe de service ne change pas. En effet, puisqu'on compte qu'une trame est reçue (resp. transmise) lorsque son dernier bit est reçu (resp. transmis), le délai maximal subi par une trame n'est autre que celui subi par son dernier bit, qui est majoré par le délai subi par un bit quelconque.

Pour les multiplexeurs par contre, le changement est important. En effet, nous avons jusqu'ici modélisé ces derniers en fonction du débit du lien physique  $C$  qui leur est raccordé, 10 ou 100 Mbps. Par contre, nous ne disposons pas d'hypothèses sur le débit des liens en termes de trames par seconde. Pour déterminer la courbe de service en trames, nous nous sommes placés dans le cas pire d'émission en termes de trames : le multiplexeur fait sortir les bits à une vitesse de  $C$  bits par seconde, donc il émet un nombre de trames plus faible lorsqu'il n'émet que des trames de taille maximale. En notant ici  $S_{\max}$  le maximum des  $S_{\max}$  de tous les flux dans ce multiplexeur, le débit de ces trames est donc de  $C/S_{\max}$  trames par seconde. De plus, même si le multiplexeur est conservatif dans le domaine des bits (il fait sortir les bits de manière continue), il ne l'est plus dans le domaine des trames avec la convention retenue : il faut en effet attendre que le dernier bit d'une trame soit transmis avant de compter la trame transmise. Le multiplexeur peut donc attendre jusqu'à  $S_{\max}/C$  unités de temps entre deux trames consécutives. On obtient ainsi la courbe de service en trames suivante, qui minore la courbe en escalier que nous venons de décrire :

$$\beta_f(t) = \max\left(0, \frac{C}{S_{\max}} \left(t - \frac{S_{\max}}{C}\right)\right)$$

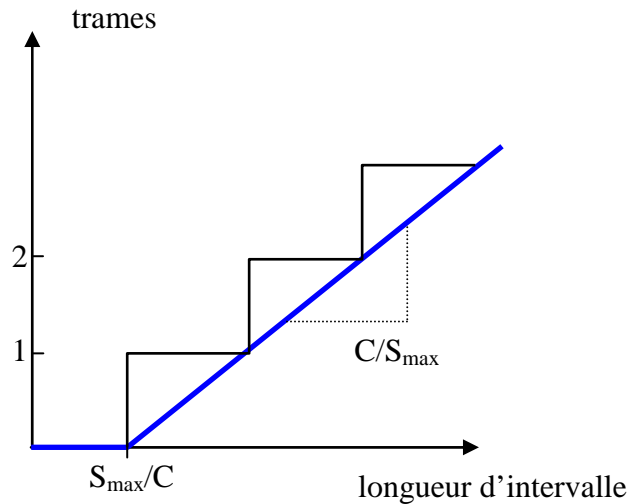


Figure 24 Courbe de service en trames d'un multiplexeur

### 3.1.3 Enveloppes de groupe en trames

Dans le domaine des trames, l'enveloppe d'un groupe ne peut plus se calculer de la même manière. On sait tout de même que l'enveloppe de groupes est toujours bornée par :

$$R_g^f(t) \leq \sum_{i \in g} R_i^f(t)$$

On cherche à exprimer d'autre part que le groupe ne peut émettre plus de trames qu'il est physiquement possible d'en émettre sur un lien de capacité  $C$ . Au maximum, en faisant passer  $C$  bits/s, il est possible de faire circuler  $C/S_{\min}$  trames par seconde sur un lien physique pour ce groupe ( $S_{\min}$  désignant ici le minimum des  $S_{\min}$  pour tous les flux appartenant au groupe). On a alors la relation :

$$R_g^f(t) \leq \frac{C}{S_{\min}} \cdot t$$

D'où l'expression en termes de trames de l'enveloppe d'un groupe :

$$\forall t, R_g^f(t) \leq \min \left\{ \sum_{i \in g} R_i^f(t), \frac{C}{S_{\min}} \cdot t \right\}$$

Une fois les groupes créés, la propagation de leurs enveloppes répond aux mêmes règles que celle des enveloppes en termes de bits.

### 3.1.4 Applicabilité des théorèmes

Nous connaissons maintenant les enveloppes et les courbes de service en termes de trames. On peut facilement remarquer que les démonstrations des théorèmes du Network Calculus sont également valides dans le domaine des trames, comme l'a d'ailleurs signalé Le Boudec dans [33].

Tous les ingrédients sont réunis pour effectuer une analyse complète du réseau en termes de trames. En particulier, on peut remarquer que les résultats obtenus pour les tailles des files d'attente seront un nombre de trames, ce qui était la motivation de cette analyse.

### 3.1.5 Stabilité des éléments dans le domaine des trames

Les paragraphes précédents ont présenté des résultats généraux permettant d'accomplir l'analyse d'un réseau dans le domaine des trames. Nous avons naturellement essayé

d'effectuer cette analyse sur notre exemple de réseau avionique. Malheureusement, ceci est impossible pour une raison relativement simple. Par rapport au domaine des bits, les enveloppes initiales des flux sont divisées par  $S_{\max}$ . Les courbes de service sont elles décalées, mais surtout divisées par  $\max(S_{\max})$ . Par rapport au domaine des bits, le réseau analysé dans le domaine des trames paraît donc plus chargé. En fait, les courbes de service calculées en termes de trames pour les multiplexeurs sont tellement faibles dans notre cas, que pour certains d'entre eux elles ne coupent pas la courbe d'arrivée. Ceci caractérise une situation d'instabilité, dans laquelle il n'est pas possible d'appliquer les théorèmes : on ne peut pas calculer les différents extremums qui y figurent.

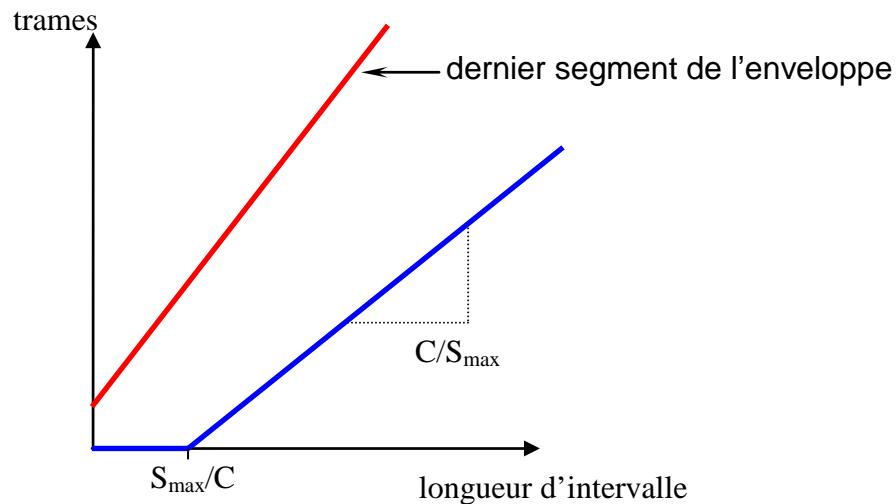


Figure 25 Illustration de l'instabilité

Notre exemple de réseau ne se prête donc pas à l'analyse en termes de trame décrite précédemment. La méthode reste cependant valide, et pourrait donner de bons résultats sur un autre réseau (par exemple avec des flux de trames de taille constante). Il nous reste cependant à étudier le cas où l'analyse dans les deux domaines simultanément est possible, notamment la manière de consolider les résultats obtenus par les deux analyses. Ceci souligne l'importance du choix du modèle lorsqu'on souhaite appliquer une méthode basée sur le Network Calculus.

### 3.2 Proposition d'une méthode de calcul en trames

Nous avons vu précédemment qu'il n'est pas possible dans le cas du réseau considéré d'appliquer une analyse entièrement en trames. Il convient donc de déterminer une méthode permettant d'obtenir une borne sur les tailles de files d'attente en nombre de trames.

#### 3.2.1 Calcul direct à partir des bornes en bits

Nous supposons connue une borne sur la taille de files d'attente en termes de bits ( $B_b$ ) pour un multiplexeur donné. Pour être sûrs d'avoir une borne valide en nombre de trames, on doit se placer dans le cas pire, qui est celui pour lequel toutes les trames qui constituent cette file d'attente de  $B_b$  bits sont de taille minimale. Dans ce cas, une borne sur la taille de la file d'attente est donnée par :

$$B_f = \left\lceil \frac{B_b}{\min(S_{\min})} \right\rceil$$

On peut prouver la validité de cette formule par l'absurde : s'il y avait un nombre  $B$  de trames,  $B > B_f$ , ces trames auraient par définition des tailles supérieures ou égales à  $\min(S_{\min})$ ,

et donc la taille de la file d'attente en bits serait supérieure ou égale à  $B_f * \min(S_{\min})$ , ce qui est strictement supérieur à  $B_b$  et contredit l'hypothèse.

Nous présentons ci-dessous un extrait des résultats obtenus grâce à cette méthode de conversion.

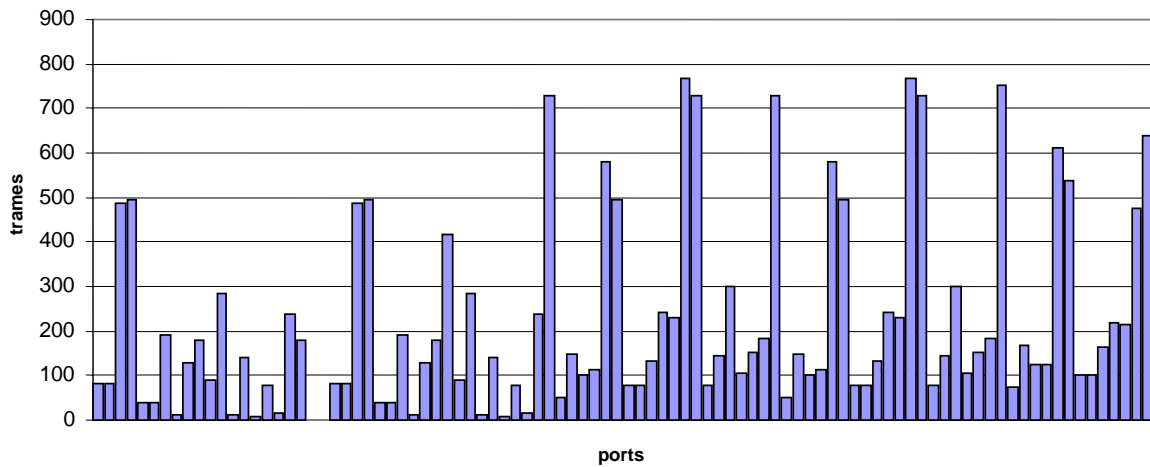


Figure 26 Résultats obtenus à partir des bornes en bits et de la taille minimale

### 3.2.2 Deuxième méthode

Nous avons développé conjointement avec Rockwell Collins une autre méthode permettant de calculer des bornes en trames, à partir des résultats sur les bornes en bits, et des enveloppes des flux dans les deux domaines.

Les résultats obtenus avec cette deuxième méthode (dont un extrait est représenté ci-dessous) sont légèrement meilleurs qu'avec la méthode plus simple précédente ; cependant, la complexité accrue des calculs ne nous a pas semblé être compensée par l'amélioration des résultats, aussi avons-nous utilisé dans cette thèse la méthode la plus simple.

■ deuxième méthode    ■ calcul classique

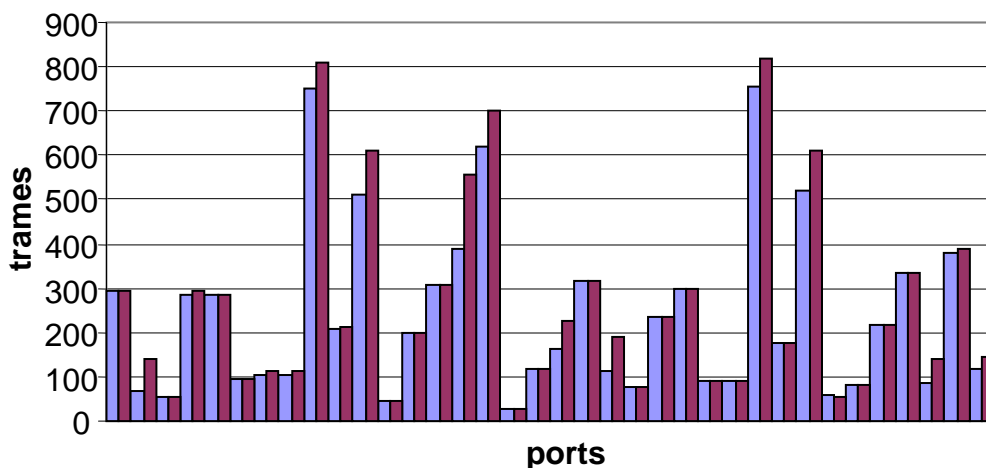


Figure 27 Comparaison des résultats des deux méthodes

Nous avons présenté dans ce troisième paragraphe une méthode d'analyse entièrement en trames, ainsi qu'une méthode permettant d'obtenir des bornes en nombre de trames à partir de celles en bits, dans le cas où cette analyse n'est pas applicable.

## 4 Prise en compte des priorités

Jusqu'à présent dans ce chapitre, nous avons montré des modifications apportées à la méthode d'analyse du réseau présentée dans le chapitre 2. A présent, nous allons présenter une modélisation différente des commutateurs, de manière à être plus fidèle au matériel qui pourra être utilisé dans un réseau avionique. Cette modélisation prend en compte un fonctionnement plus complexe des commutateurs, dans lequel la politique de service des ports de sortie n'est plus FIFO.

La politique de service retenue est à base de priorité statique. On peut la définir comme suit : chaque VL se voit attribuer une priorité différente, 1 (priorité haute) ou 2 (priorité basse). Pour permettre une utilisation plus générale, nous utilisons dans cette partie 4 une définition légèrement différente, dans laquelle la priorité des VL n'est pas définie globalement, mais localement : ainsi, un VL peut être de priorité haute pour un commutateur donné, mais de priorité basse dans le suivant. Un port de sortie transmet en priorité les trames de priorité 1, dans leur ordre d'arrivée, puis les trames de priorité 2, également dans leur ordre d'arrivée. Le traitement est non préemptif, c'est-à-dire que le traitement d'une trame de priorité 2 ne peut être interrompu par l'arrivée d'une trame de priorité 1.

Cette politique de service possède l'avantage d'être plus souple que le FIFO strict, puisqu'elle permet une différenciation de service. Elle est également relativement simple d'implémentation, puisque deux files d'attente suffisent, et que l'algorithme de choix de la file à servir est très simple.

Nous modéliserons ces ports de sortie par des multiplexeurs, offrant une courbe de service différente aux flux agrégés de priorité haute et basse.

Note : Le cas de multiplexeurs FIFO n'est autre qu'un cas particulier de cette politique, où tous les flux ont la même priorité.

### 4.1 Définition de la courbe de service

Cette politique de service est bien connue et a souvent été étudiée. On trouve le résultat suivant dans la littérature (voir Le Boudec [33] par exemple) :

Soit un élément à taux d'émission constant  $C$ , qui sert deux flux  $H$  et  $L$  avec une priorité non préemptive pour  $H$ . On note  $R_H^*(t)$  et  $R_L^*(t)$  les courbes d'arrivée de ces deux flux, et  $S_{\max}^L$  la taille maximale d'une trame du flux  $L$ .

Alors, le flux  $H$  reçoit une courbe de service de type  $\beta_{R,T}(t)$  avec  $R = C$  et  $T = \frac{S_{\max}^L}{C}$ .

Le flux de priorité basse reçoit une courbe de service  $S_L(t) = (C.t - R_L^*(t))^+$ .

### 4.2 Impact sur le calcul

#### 4.2.1 Propagation de l'enveloppe

Les calculs permettant de déterminer des bornes sur les délais et les tailles de files d'attente s'effectuent en utilisant les courbes de service définies ci-dessus. On obtient donc par exemple une borne sur le délai  $D_1$  subi par une trame appartenant à un flux de priorité 1.

Au moment de calculer l'enveloppe de sortie, on considère que ce multiplexeur se comporte pour tous les flux de priorité 1 comme un élément à délai borné  $D_1$ , de courbe de service  $\delta_{D_1}(t)$ . Pour un flux de priorité 1, l'enveloppe de sortie sera donc égale à celle d'entrée décalée de  $D_1$  vers la gauche, et similairement pour un flux de priorité 2.

#### 4.2.2 Influence sur les groupes

La notion de priorité va nous obliger à modifier la définition des groupes. En effet, pour effectuer les calculs de bornes par exemple, il faut que tous les flux d'un groupe aient la même priorité.

Un groupe dans un multiplexeur sera formé de flux (éventuellement de groupes):

- qui ont même priorité dans le multiplexeur où se forme le groupe.
- qui ont le même multiplexeur en aval
- qui ont même priorité dans ce multiplexeur aval

Cette définition assure que les calculs sont effectués avec des groupes dont tous les flux ont même priorité. De plus, les enveloppes de ces groupes sont récursivement calculées par le multiplexeur précédent, ce qui explique les deux dernières conditions. On pourrait arguer qu'il n'y a pas toujours de multiplexeur précédent, notamment dans le cas d'un commutateur directement relié à un abonné. Entre autres, dans le cadre de notre modèle, l'émission des abonnés ne prend pas en compte les priorités des VL. On peut alors considérer que tous les flux provenant d'un multiplexeur FIFO ont même priorité, et dans ce cas la création de groupe décrite dans le paragraphe 2.2.2 spécifie l'initialisation de l'enveloppe du groupe.

Par la suite, les groupes créés ainsi se manipulent comme indiqué dans la partie 2.3.

#### 4.3 Application

Pour voir l'influence de cette utilisation des priorités, nous avons appliqué la méthode d'analyse au même réseau que précédemment, avec des priorités aléatoires pour les VL. Pour simplifier l'analyse, nous avons choisi de fixer pour chaque VL une priorité identique de bout en bout.

La figure ci-dessous présente une comparaison des bornes sur les tailles de files d'attente obtenues avec une certaine affectation des priorités, par rapport aux valeurs calculées dans le cas où tous les flux avaient même priorité.

Bien entendu pour effectuer ces comparaisons, on représente la somme des bornes pour les deux priorités. Pour des raisons de clarté, seuls les résultats obtenus pour deux des commutateurs les plus chargés sont présentés.

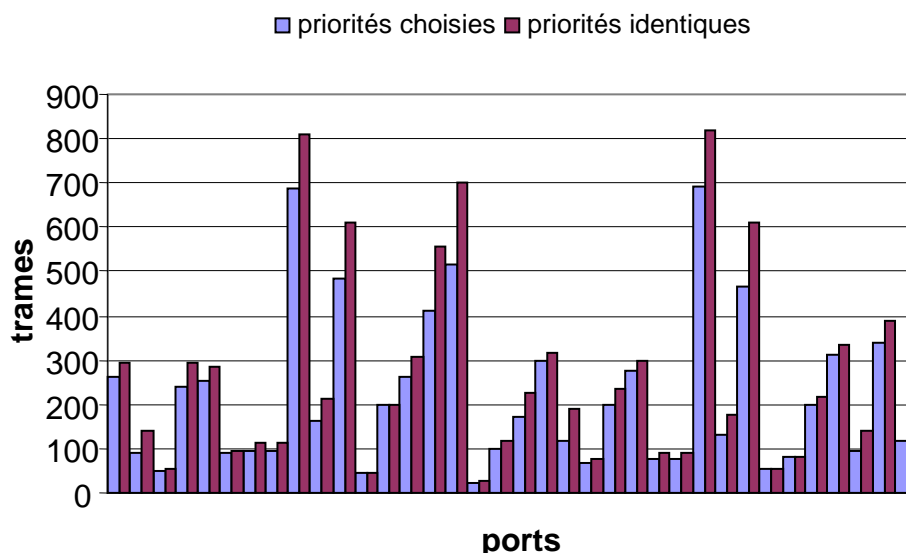


Figure 28 Exemple d'affectation de priorités

On observe sur ces résultats qu'il est possible d'avoir de meilleures bornes lorsqu'on utilise la notion de priorités. Par contre, sur d'autres ports et avec un autre choix de priorités, on obtient de moins bonnes bornes :

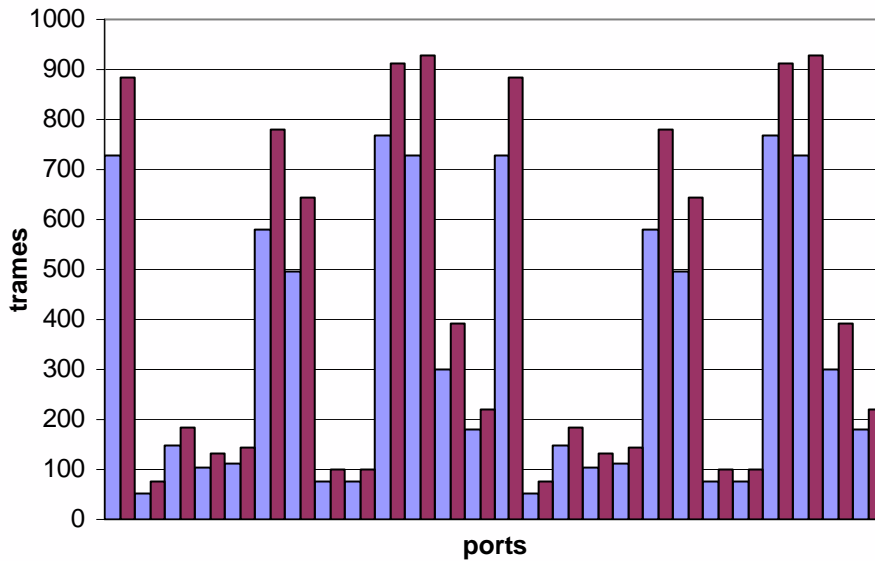


Figure 29 Cas d'une autre affectation de priorités

On montre ainsi que l'affectation des priorités a une réelle influence sur les résultats de notre méthode d'analyse. Il convient donc de choisir de façon adaptée ces priorités, mais ce choix est un problème extrêmement complexe, que nous étudierons plus avant dans le chapitre suivant.

## Conclusion

Dans ce chapitre, nous avons tout d'abord présenté plusieurs améliorations de la méthode d'analyse de réseau, qui permettent d'obtenir des bornes bien meilleures qu'avec la méthode présentée dans le chapitre précédent. Ensuite, nous avons apporté à cette méthode de nouvelles fonctionnalités, comme l'analyse en trames ou la prise en compte de plusieurs niveaux de priorité.

Nous disposons à présent d'un outil permettant d'étudier un réseau avionique. L'application à notre réseau test a démontré que celui-ci satisfait bien les exigences de performance qui ont été fixées. Cependant, les bornes obtenues sont proches de ces contraintes, et laissent peu de marge pour les futures évolutions du réseau. Dans le chapitre suivant, nous présentons plusieurs pistes permettant de mieux utiliser le réseau existant, de façon à ménager un peu plus de marges. Par exemple, un algorithme de choix d'affectation des priorités y est proposé.



## CHAPITRE 4

### Méthodes d'optimisation du réseau

Dans le chapitre précédent, nous avons proposé une méthode d'analyse d'un réseau avionique. Celle-ci permet notamment de vérifier si un réseau étudié est capable de fournir aux flux qui le traversent une qualité de service, caractérisée dans le cadre de notre étude par un délai de bout en bout et une gigue bornés, ainsi que par la garantie de ne perdre aucune trame par congestion d'une file d'attente. Pour garantir ce dernier point, nous obtenons des bornes sur la taille maximale de chacune des files d'attente, bornes qui sont ensuite comparées à la taille physique de la mémoire réservée pour ces files. L'analyse grâce à cette méthode d'un prototype de réseau, mis à notre disposition par notre partenaire industriel, fait apparaître deux éléments importants. Premièrement, le réseau de test est analysable par la méthode, et les résultats de cette analyse indiquent que le réseau satisfait bien aux contraintes rappelées ci-dessus. Deuxièmement, les marges sont faibles : pour certaines files d'attente, les contraintes ne sont respectées qu'à quelques pourcents près seulement. C'est notamment le cas pour les ports les plus chargés du réseau, qui pourraient difficilement accueillir plus de flux.

Nous souhaitons souligner l'importance d'avoir des marges importantes en aéronautique. De manière générale tout d'abord, il est préférable pour des raisons de sécurité que le fonctionnement normal d'un système soit éloigné des limites de son domaine d'utilisation. D'autre part, les avions possèdent des durées de vie très importantes, ce qu'il faut prendre en compte dès la phase de conception, en laissant des marges d'évolution. L'ajout de nouveaux équipements réalisant de nouvelles fonctions est quasiment incontournable au cours des 30 années de la vie de l'appareil. Le réseau avionique devra être capable de supporter les nouveaux besoins de communication induits par ces ajouts. Il faut également se rappeler qu'il existe une réutilisation importante des technologies d'un programme à l'autre. Dans le cadre d'un programme ultérieur, de fortes économies pourraient être réalisées si le réseau avionique étudié était utilisé, ce qui implique qu'il ne soit pas dimensionné au plus juste des besoins de communication actuels.

Nous avons donc essayé de trouver des moyens de ménager cette marge recherchée, c'est-à-dire des modifications du réseau telles que les résultats de l'analyse soient meilleurs.

Nous avons identifié deux domaines susceptibles de recevoir des modifications :

- Le trafic circulant dans le réseau
- L'architecture physique du réseau

En ce qui concerne le trafic, les modifications pourraient être importantes : on pourrait par exemple diminuer la cadence d'émission d'un VL donné, ou décider de scinder en deux les trames des VL dont le  $S_{\max}$  est grand. Cependant, nous avons considéré que ces modifications ne peuvent être effectuées sans une bonne connaissance de l'importance de chaque VL pour les systèmes concernés ; aussi avons-nous décidé de ne pas considérer ce type de changements, et de prendre en compte le trafic tel qu'il nous a été décrit. La seule modification restante porte alors sur la priorité de chacun des VL, non spécifiée actuellement par notre partenaire industriel, et qui peut s'avérer un levier intéressant comme indiqué dans le chapitre précédent. La première partie de ce chapitre présente plusieurs méthodes permettant de choisir ces priorités de manière efficace.

En ce qui concerne l'architecture physique du réseau, nous avons étudié deux axes de recherche. Le premier est relatif à la manière de relier les commutateurs entre eux : nous avons voulu voir si multiplier le nombre de liens permettait de diminuer l'importance des

congestions. Le deuxième axe envisagé porte sur une modification de la politique de service des ports de sortie des commutateurs. Nous avons étudié l'impact sur la méthode d'analyse d'une « régulation » des flux (« reshaping ») dans ces ports.

Note : les résultats présentés dans ce chapitre ont été obtenus avec un policing en bits, présenté dans le chapitre 3. Nous rappelons que le réseau analysé n'est pas un réseau réel, mais un exemple représentatif de la complexité d'un réseau avionique.

## 1 Affectation des priorités aux VL

### 1.1 Complexité du problème

Le réseau avionique que nous avons étudié permet de prendre en compte une priorité différente pour chaque VL. Plus précisément, il est possible de choisir pour chaque VL quelle sera sa priorité dans chaque port de sortie de chaque commutateur du réseau. Cependant, pour des raisons de facilité d'implémentation, il est très probable qu'un VL aura la même priorité dans tous les ports de sortie d'un même commutateur. Le problème que nous avons étudié est celui du choix de ces priorités. (On appelle configuration un choix de toutes les priorités)

Nous avons choisi de mesurer la valeur d'une configuration selon deux critères. Le premier est la valeur du maximum des bornes calculées sur les tailles de files d'attente (priorité haute + priorité basse), en trames. En effet, ce critère est très important puisque c'est lui qui indique s'il existe un risque ou non de perdre des trames par congestion des files d'attente (supposées limitées par un nombre total maximal de trames). Nous voulions cependant nous assurer que diminuer le maximum n'augmente pas la moyenne de façon trop importante. Nous voulions également connaître l'influence de cette optimisation sur les valeurs des bornes sur les délais dans chaque port. Le second critère choisi porte donc sur la moyenne des bornes sur les délais. Cependant, nous avons jugé qu'une moyenne pure a peu de signification : en effet, si le délai  $D_1$  d'un port concerne 200 flux, alors que le délai  $D_2$  d'un autre port ne concerne que

2 flux, il ne serait pas significatif de dire que la moyenne des délais est de  $\frac{D_1 + D_2}{2}$ . Nous

avons préféré à cette moyenne pure la moyenne des bornes sur le délai pondérée par le nombre de flux qui risquent de subir ce délai. Pour raccourcir l'expression de ce second critère, nous l'avons improprement appelé « espérance » du délai. Dans le cas précédent, on

dira donc que l'espérance du délai est de :  $\frac{200.D_1 + 2.D_2}{202}$ .

Du point de vue mathématique, le problème étudié est un problème d'optimisation de grande complexité. En effet, le nombre de variables est très important : en faisant l'hypothèse simplificatrice qu'un VL doit avoir la même priorité partout dans le réseau (ce qui divise par trois environ le nombre de paramètres), il reste autant de paramètres que de VL dans le réseau, soit 1008. Chacun de ces paramètres est binaire, puisque la priorité peut être haute ou basse. L'espace des solutions est donc de dimension 1008, et contient  $2^{1008}$  configurations différentes !

A ce grand nombre de solutions possibles s'ajoute le fait que le réseau est un système très complexe, et que son comportement par rapport à chacune des variables est extrêmement difficile à expliciter. Prenons comme exemple l'effet des groupes : modifier la priorité d'un VL appartenant à un groupe peut avoir très peu d'effet ou bien énormément, notamment dans le cas où le VL est le seul composant de ce groupe, et qu'il peut rejoindre en changeant de priorité un groupe existant (on diminue le nombre total de groupes). Un autre effet est celui induit par la taille variable des trames : il suffit parfois de modifier la priorité d'un seul VL pour modifier la valeur du minimum de  $S_{\min}$ , qui est utilisée pour calculer les bornes en

termes de trames. Un dernier facteur de complexité du problème réside dans le fait que l'analyse du réseau se fait par propagation. Il suffit que la modification d'une priorité ait de l'importance en un point pour automatiquement avoir des répercussions dans tout le réseau : une amélioration locale peut dégrader tout le reste du réseau.

De manière intuitive, on peut toutefois dégager une première règle pour les choix des priorités : en général, il vaudra mieux mettre en priorité haute des trames de petite taille. Si une trame très longue se trouve en priorité haute, elle sera traitée en priorité mais bloquera pendant longtemps le port du commutateur. Si les trames en priorité basse sont courtes, il peut s'en accumuler un grand nombre pendant l'émission d'une seule longue trame. Il vaut mieux donc donner la priorité aux trames courtes, pour désengorger le plus vite possible les files d'attente (dans lesquelles une trame courte compte autant qu'une longue).

Pour conclure la présentation de notre problématique, nous soulignons le fait que nous ne cherchons pas LA solution optimale au problème considéré. En effet, la complexité du problème fait qu'il est impossible de démontrer l'unicité d'une telle solution (l'existence est garantie, puisqu'il n'existe qu'un nombre fini, mais gigantesque, de configurations); il nous est même impossible, connaissant une bonne solution, de vérifier si elle est ou non optimale. Nous nous contenterons de rechercher des configurations qui apportent des gains, compte tenu des critères d'évaluation retenus, par rapport à la configuration de base dans laquelle tous les VL ont la même priorité (basse, par exemple).

## **1.2 Algorithmes classiques**

Pour résoudre ce problème d'optimisation complexe, nous avons envisagé de faire appel à des méthodes classiques ; il nous a cependant semblé qu'aucune ne s'appliquait simplement à notre problème. Nous avons cependant retenu deux méthodes, qui après adaptation pouvaient nous apporter des résultats : l'algorithme de réduction d'espace solution alpha bêta, et la méthode du gradient. L'objectif de nos expériences était de voir si ces algorithmes pouvaient ou non découvrir de bonnes configurations en un temps de calcul acceptable.

### **1.2.1 Une variation sur l'algorithme Alpha bêta**

Dans cet algorithme, on représente les solutions possibles dans un arbre, dans lequel des coupes permettent de restreindre les branches qu'il est nécessaire d'explorer. Dans notre cas, nous créons l'arbre des solutions de la manière suivante : les VL sont ordonnés (le choix de l'ordre étant important, nous le détaillerons par la suite), et un nœud situé à la hauteur  $n$  de l'arbre constituera un choix de priorités pour les  $n$  premiers VL. L'arbre ainsi construit est binaire, puisqu'à chaque nouveau niveau, le  $n+1^{\text{ème}}$  VL peut adopter la priorité haute ou basse. Il comporte autant de niveaux que de VL présents dans le réseau, soit 1008 dans notre cas d'étude. Une branche complète, depuis la racine jusqu'à une feuille, décrit donc une configuration complète.

Nous définissons comme valeur d'un nœud appartenant au  $n^{\text{ème}}$  niveau le maximum de toutes les bornes sur les tailles en trames calculées dans tous les ports du réseau, en supposant que ne circulent dans le réseau que ces  $n$  VL, avec des priorités fixées suivant la position du nœud dans l'arbre. Pour élaguer les branches, on constate que tous les descendants d'un nœud ont forcément des valeurs supérieures à celle du nœud considéré, simplement parce que plus de trafic circule dans le réseau. Si on dispose d'une valeur de référence, par exemple celle obtenue pour une configuration donnée, on élaguera toutes les branches issues d'un nœud qui aurait une valeur supérieure à cette valeur référence.

#### **Première expérience**

La première expérience que nous avons menée possédait comme individu de référence celui pour lequel tous les VL sont de priorité basse. Nous avons essayé de déterminer un ordre de

grandeur de la plus grande coupe qui serait effectuée par l'algorithme, pour essayer de prédire sa vitesse d'exécution. En nous basant sur l'hypothèse qu'il vaut mieux mettre les VL qui possèdent de petites trames en priorité haute, nous avons déduit qu'en mettant en premier les VL qui possèdent les plus grands  $S_{\min}$  en priorité haute, nous allons construire une mauvaise configuration, qui serait d'une longueur proche de celle de la plus petite branche explorée par l'algorithme. Nous avons constaté qu'avec seulement les 536 VL de plus grand  $S_{\min}$  présents dans le réseau, la valeur de l'individu référence était dépassée. Cette constatation nous a conduit à reconsidérer le fonctionnement de l'algorithme. Nous souhaitons en effet choisir une configuration en parcourant une branche de l'arbre, depuis la racine jusqu'à une feuille ou jusqu'à dépasser la valeur de référence. Dans ce dernier cas, nous serions remonté d'un niveau puis aurions parcouru le demi arbre restant. Cependant, sachant qu'il suffit d'à peu près 500 flux pour dépasser cette valeur de référence, le nombre de branches à explorer est immense, et il aurait fallu un temps considérable avant d'aboutir à une seule configuration complète.

### Deuxième expérience

Nous avons alors décidé de partir d'un individu de référence, donnant déjà de « bons » résultats. L'algorithme retenu pour parcourir l'arbre est le suivant :

On examine les configurations qui sont identiques à la configuration de référence depuis la racine jusqu'au niveau  $n$ . Le  $n$  est décroissant, de 1007 à 1. Ainsi, l'étude du niveau  $n$  permettra d'obtenir toutes les configurations similaires à celle de référence pour les  $n$  premiers VL, et différentes pour les  $(1008-n)$  VL restants, et dont la valeur est inférieure à celle de référence.

On note  $H$  un VL de priorité haute,  $B$  sinon. Soit  $S_n = a_1a_2a_3\dots a_n$  la chaîne formée par les priorités des  $n$  VL mises bout à bout. Par exemple, pour  $n=4$ ,  $S_4$  peut valoir  $HBBH$  ou  $BHBH$ .  $S_n$  permet de décrire une configuration partielle de  $n$  VL.

L'étude d'une configuration de niveau  $n$  se déroule comme suit :

On calcule la valeur de la configuration décrite par  $S_n$  grâce à notre méthode d'analyse.

- si sa valeur est moins bonne que celle de référence, l'analyse est terminée, car tous ses descendants seront également moins bons.
- si sa valeur est meilleure,
  - o si la configuration est complète (on a choisi la priorité de chaque VL,  $n=1008$ ) on a trouvé une configuration meilleure que celle de référence.
  - o sinon, on examine récursivement les configurations  $S_n+H$  et  $S_n+B$ .

Cet algorithme présente l'avantage de partir d'une configuration connue, et de découvrir parmi les configurations proches de lui (dont les premiers VL ont la même priorité que pour lui) s'il en existe une meilleure.

La configuration de référence que nous avons choisi pour appliquer cette méthode a été obtenue grâce à une heuristique décrite dans le paragraphe 1.3.1.

Un des points fondamentaux pour la bonne marche de l'algorithme est bien sûr l'ordre dans lequel sont rangés les VL. Si les derniers VL sont tels que la modification de leur priorité ne change pas beaucoup la valeur de la configuration, alors l'algorithme va rechercher en priorité des configurations meilleures que celle de référence, mais dont le gain ne sera pas très important. Si au contraire on place en dernier des VL dont la modification de la priorité change grandement la valeur de la configuration, l'algorithme va rapidement remonter dans les niveaux, si la configuration de référence a déjà bien choisi ces priorités. Il sera difficile et long de trouver une configuration meilleure que celle de référence, mais le gain espéré est plus grand.

Il est toutefois difficile de quantifier si la modification de la priorité d'un VL va ou non modifier grandement la valeur calculée, et donc de classer ces flux. Par exemple, l'effet de

groupe peut faire que la modification d'un VL ne va pas beaucoup changer la valeur, sauf s'il est modifié en même temps que certains autres VL. Pour pouvoir néanmoins appliquer notre méthode, nous avons choisi de classer les VL d'après leur  $S_{\min}$  car il nous a semblé que c'était un critère très important, notamment lors du calcul des bornes en nombre de trames.

Dans une première application, nous avons rangé ces VL par ordre croissant, ce qui devait nous permettre d'assister au premier comportement décrit ci-dessus. En pratique, l'algorithme a découvert assez rapidement (45 niveaux) un individu meilleur que celui de référence, mais avec un gain très faible, inférieur au pourcent. Par la suite, il n'a plus découvert de nouvelle bonne configuration avant que le critère de terminaison que nous avons fixé ne soit satisfait (5 millions de configurations testées).

Dans le cadre d'une deuxième application, nous avons rangé les VL par ordre décroissant, en espérant découvrir de bien meilleures configurations. Malheureusement, avant que le critère de terminaison ne soit satisfait, l'algorithme avait étudié 104 niveaux sans obtenir de configuration complète.

D'après ces expériences, nous avons conclu plusieurs choses. Tout d'abord, la complexité du problème est telle que l'algorithme ne peut pas aboutir à de bonnes configurations si on ne lui donne aucune information complémentaire. Connaître une bonne configuration de référence, ou ranger les VL dans un bon ordre, permettraient de mieux diriger l'algorithme, mais le temps de calcul reste rédhibitoire. D'autre part, la configuration de référence choisie est assez bonne, puisque l'algorithme n'a pas pu découvrir de configuration réellement meilleure, malgré un long temps de calcul (5 millions de configurations).

Il semble donc que cet algorithme, fondé sur la coupe alpha bêta, ne permet pas de choisir comment affecter les priorités des VL ; il permet tout au plus de vérifier si on ne peut pas gagner un petit peu en modifiant légèrement une configuration connue.

### 1.2.2 Une variation sur la méthode du gradient

L'application de la méthode du gradient classique est délicate dans notre cas, où toutes les variables sont binaires. Nous proposons donc l'algorithme suivant, qui peut être vu comme une variation de la méthode de plus forte pente.

```
booléen trouvé_mieux = vrai
configuration ref

tant que (trouvé_mieux) {
    trouvé_mieux = faux
    pour (i=1 à nb_VLs) faire {
        obtenir un nouvelle configuration à partir de ref en
changeant la priorité du VL i
        calculer la valeur de la nouvelle configuration
        si elle est meilleure que ref:
            trouvé_mieux = vrai
            ref = nouvelle configuration
    }
}
```

Comme précédemment, nous définissons ici la valeur d'une configuration comme étant la plus grande des bornes (exprimée en nombre de trames) calculées sur toutes les files d'attente du réseau. On remarque que dans cette méthode, l'ordre des VL est beaucoup moins

important que pour la méthode précédente, puisque les VL sont scrutés de manière successive.

Nous avons mené de nombreuses expériences en partant de configurations aléatoires, mais aucune ne s'est arrêtée malgré un temps de calcul important. Ceci est très certainement dû à la taille très importante du problème, et à la non linéarité par rapport aux différentes variables. Par contre, utiliser cet algorithme à partir de configurations connues pour être assez bonnes nous a permis de découvrir de nouvelles configurations, très légèrement meilleures car assez « proches » de celle de départ. Nous avons donc décidé de retenir cet algorithme, qui nous permettra de « finaliser » les résultats d'autres recherches, en recherchant autour des solutions trouvées s'il existe des configurations légèrement meilleures. Cette recherche doit être assez rapide si les solutions sont déjà bonnes, grâce à la simplicité de l'algorithme.

D'après ces deux expériences, nous avons conclu que le problème était trop complexe, et ne permettait pas d'appliquer des méthodes d'optimisation classique. Nous nous sommes alors penchés sur deux types d'approches différentes. Tout d'abord nous avons décidé d'utiliser des heuristiques, qui permettent d'utiliser la connaissance que nous avons du fonctionnement du réseau pour trouver de bonnes configurations. Dans un deuxième temps, nous avons essayé d'utiliser des algorithmes d'évolution, qui sont généralement bien adaptés aux problèmes de grande dimension et de complexité élevée.

### **1.3 Heuristiques d'affectation**

#### **1.3.1 Heuristique « $S_{\min}$ »**

La première heuristique que nous présentons est directement reliée à l'intuition que nous avons déjà décrite, selon laquelle il vaut mieux mettre les petites trames en priorité haute. De cette façon, on espère diminuer le nombre de trames présentes dans un port de sortie à tout moment, en évitant le blocage de nombreuses petites trames le temps d'en émettre une très longue. Du point de vue de la méthode de calcul déterministe, on retrouve bien l'influence de ce facteur  $S_{\min}$ . Dans le calcul des bornes sur les tailles de files d'attente, qui peut être effectué selon deux méthodes présentées dans le chapitre précédent, on retrouve systématiquement le facteur  $S_{\min}$  au dénominateur. La borne calculée est donc inversement proportionnelle à la plus petite taille d'une trame. Plus précisément, le facteur impliqué est le minimum des  $S_{\min}$  pour tous les flux d'une même priorité. Ceci nous fait penser qu'il vaut mieux que tous les flux possédant le même  $S_{\min}$  aient la même priorité. Si par exemple on choisit la priorité haute pour tous les VL possédant le plus petit  $S_{\min}$ , on est sûrs que parmi les flux restant en priorité basse, il n'y aura que des  $S_{\min}$  strictement supérieurs, et donc on diminuera la borne calculée sur la file d'attente de priorité basse.

#### Application

Nous avons appliqué le raisonnement précédent en utilisant l'heuristique suivante, que nous avons baptisée heuristique  $S_{\min}$  :

- La configuration initiale a tous les VL en priorité basse
- Les VL sont triés par ordre de  $S_{\min}$  croissant. (les ex-aequo rangés aléatoirement)
- La première configuration est obtenue en mettant le premier VL en priorité haute
- La deuxième en mettant les deux premiers VL en priorité haute, etc.

On obtient ainsi autant de configurations qu'il y a de VL présents dans le réseau, soit 1008. Le graphique suivant présente les résultats obtenus, où pour chaque configuration on indique le maximum des bornes sur les tailles de files d'attente, maximum pour tous les ports de tous les commutateurs du réseau (borne priorité haute + borne priorité basse).

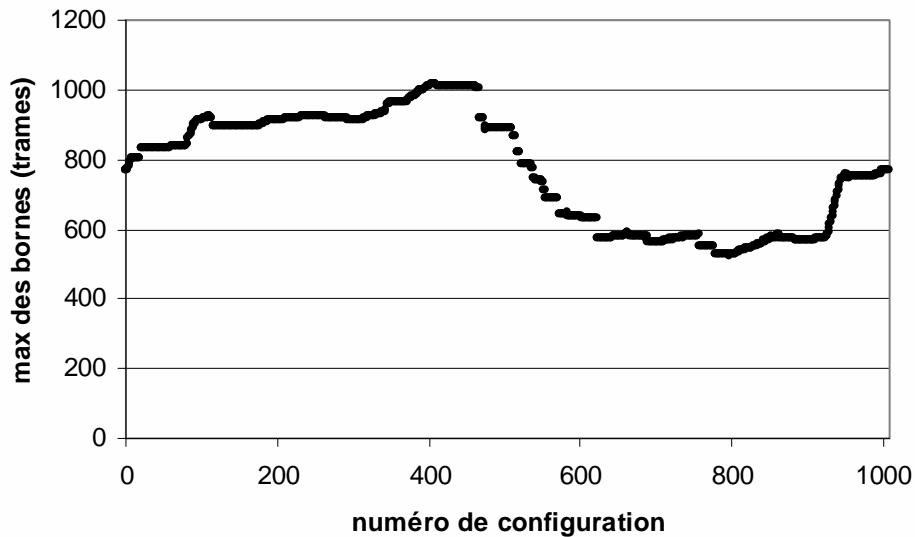


Figure 30 Evolution de la taille max durant l'heuristique

On peut remarquer plusieurs choses sur cette courbe. Tout d'abord, les valeurs obtenues pour la première et la dernière configuration sont identiques. Ceci est tout à fait normal puisque dans ces deux cas tous les VL ont même priorité (tous basse dans le premier cas, haute dans le second), et dans ces cas les ports d'attente des commutateurs se comportent comme une simple file FIFO. La deuxième constatation est que la courbe obtenue présente un certain nombre de paliers. Ces discontinuités peuvent s'expliquer par la présence dans les commutateurs de plusieurs VL qui présentent le même  $S_{\min}$ . Tant qu'ils n'ont pas tous changé de priorité, il reste des VL de ce  $S_{\min}$  là en priorité basse, et la discontinuité se rencontre lorsque le dernier VL de cet ensemble change de priorité.

Pour les tendances générales, on constate que la courbe est croissante dans un premier temps, avant de redescendre pour présenter un minimum, puis de remonter jusqu'à la valeur initiale. La croissance initiale s'explique en se rappelant qu'on groupe les flux provenant du même port, allant dans le même port, et de même priorité. Lorsqu'on passe un VL en priorité haute au début, on crée un nouveau groupe, de priorité haute, sans pour autant supprimer celui de priorité basse. En effet, il n'y a aucune raison que tous les VL appartenant à un même groupe aient tous un  $S_{\min}$  faible. Dès lors, on perd un peu du bénéfice apporté par la notion de groupe, en multipliant le nombre de ceux-ci, comme l'explique la figure ci-dessous. On peut y lire que pendant les 500 premières configurations, on ajoute 2000 groupes de priorité haute, alors que dans le même temps on n'en supprime que 700 de priorité basse.

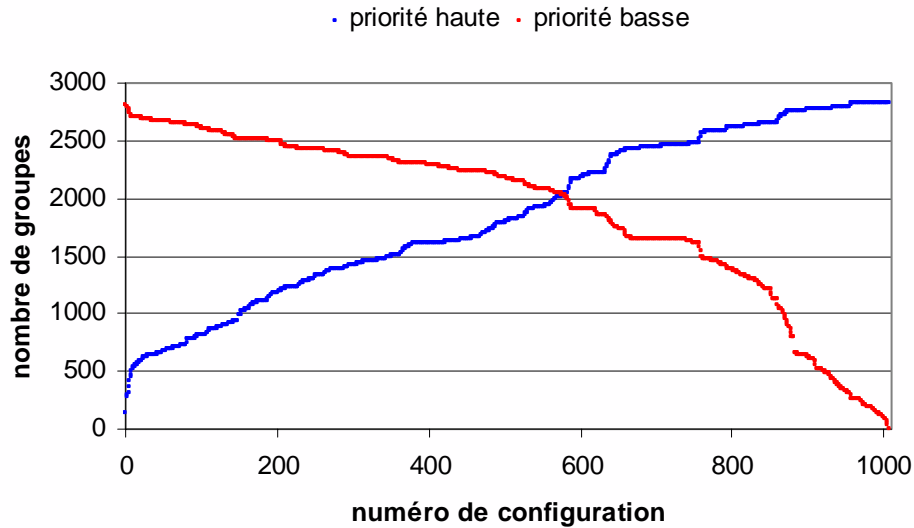


Figure 31 Evolution du nombre de groupes durant l'heuristique

La décroissance constatée au bout de 500 configurations s'explique en partie par le fait qu'on crée moins de groupes de priorité haute, alors qu'on en supprime de plus en plus de priorité basse (ils ne contiennent plus de VL). Cependant, le facteur principal est le fait que la taille minimale de trames pour les groupes de priorité basse restants commence à augmenter fortement ; en effet, on a enlevé de ces groupes les flux possédant un  $S_{\min}$  faible :

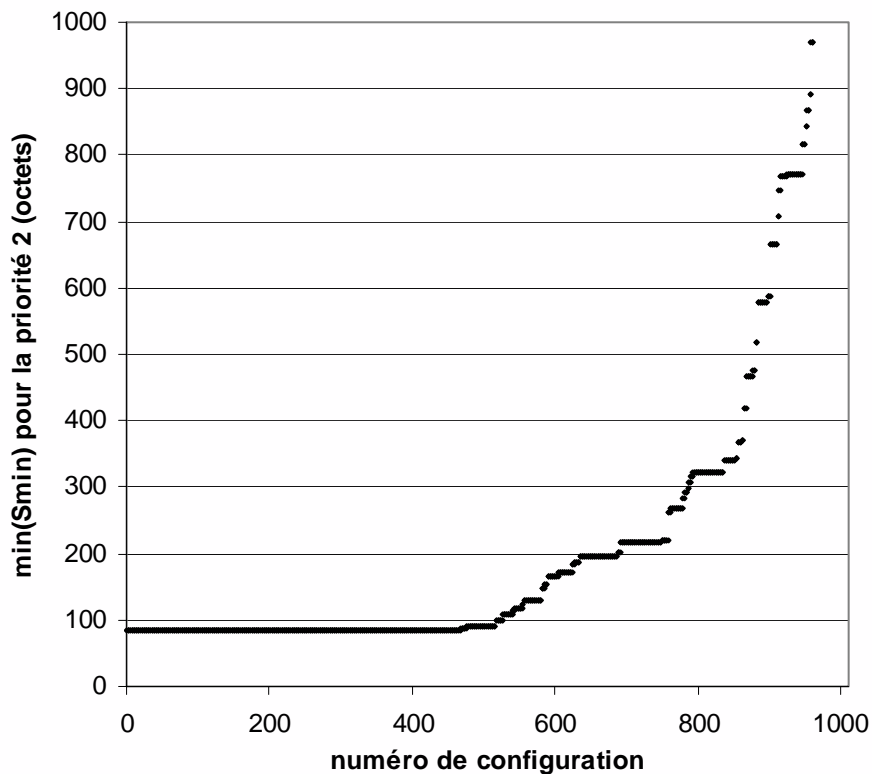


Figure 32 Evolution de la taille min pour les groupes de priorité basse durant l'heuristique

Comme le calcul de tailles en trames utilise ce facteur, on constate une forte diminution de la contribution de la priorité basse à la taille totale.



Ce phénomène continue jusqu'à la fin de l'heuristique, mais comme le nombre de groupes restants en priorité basse diminue de plus en plus, l'effet est moins important, ce qui explique la remontée du maximum des tailles vers la fin de l'heuristique. Pour illustrer cette remontée, la figure ci-dessous retrace la somme, dans tout le réseau, de tous les  $S_{\min}$  de tous les groupes. On peut y lire que l'effet dû à l'augmentation du minimum des  $S_{\min}$  est très largement compensé par l'effet contraire, relatif à la diminution du nombre de groupes de priorité basse.

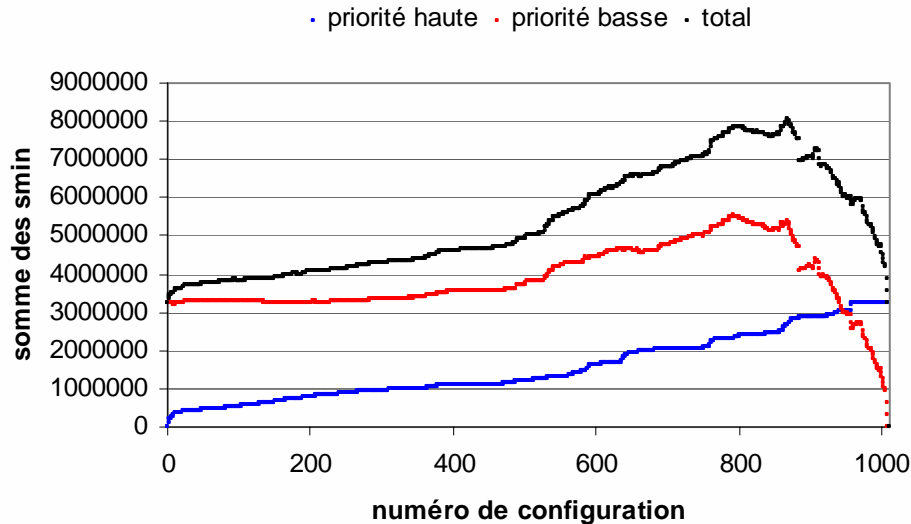


Figure 33 Evolution de somme des  $S_{\min}$  durant l'heuristique

La dernière remarque a trait à la valeur de la configuration pour laquelle le minimum est atteint. On obtient en effet une valeur de 526 trames, contre 769 pour la configuration initiale, soit une baisse d'environ 30%, ce qui est très positif. C'est cette configuration que nous avons choisie comme configuration de référence, par exemple pour les algorithmes du gradient et de coupe alpha bêta présentés ci-dessus.

#### Analyse complémentaire

Nous soulignons tout d'abord que les bons résultats de cette heuristique sont très liés à la méthode de calcul des bornes sur les tailles en trames, c'est-à-dire à la déduction de bornes en termes de trames, à partir des bornes calculées en termes de bits. En effet, au cours de l'analyse, la borne maximale en bits est loin de se comporter de la même manière :

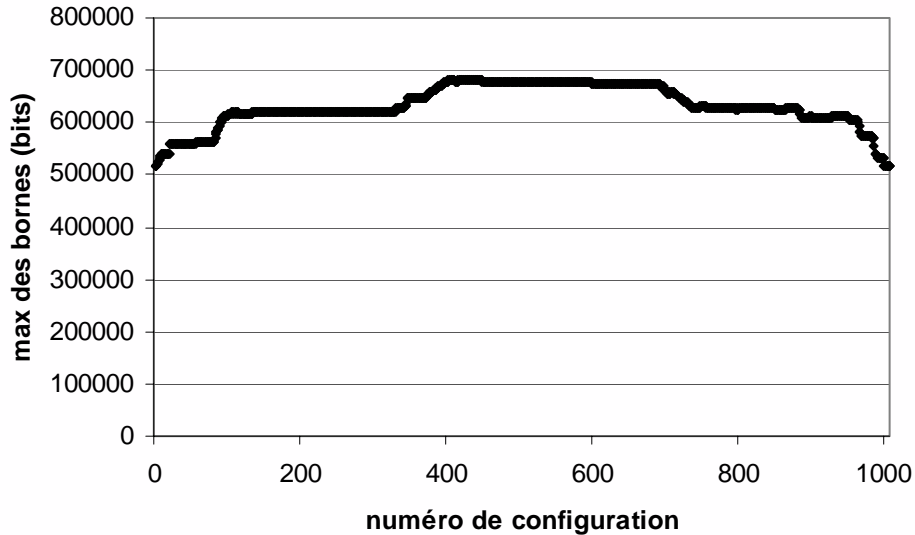


Figure 34 Evolution de la taille max en bits durant l'heuristique

Ce qui permet d'expliquer les bons résultats en terme de trames, alors même que les tailles en bits sont dégradées, c'est que la manière de calculer la taille en trames à partir de la taille en bits, pour une priorité donnée, fait intervenir le minimum des  $S_{\min}$  pour les VL de cette priorité. Or, comme nous l'avons vu, l'heuristique permet de trouver des configurations pour lesquelles les  $S_{\min}$  des VL de priorité basse sont de plus en plus élevés, ce qui permet de réduire la contribution de la priorité basse à la taille totale. La taille en bits dépend plus directement du nombre de groupes présents dans le réseau, que l'on a représenté dans la figure suivante :

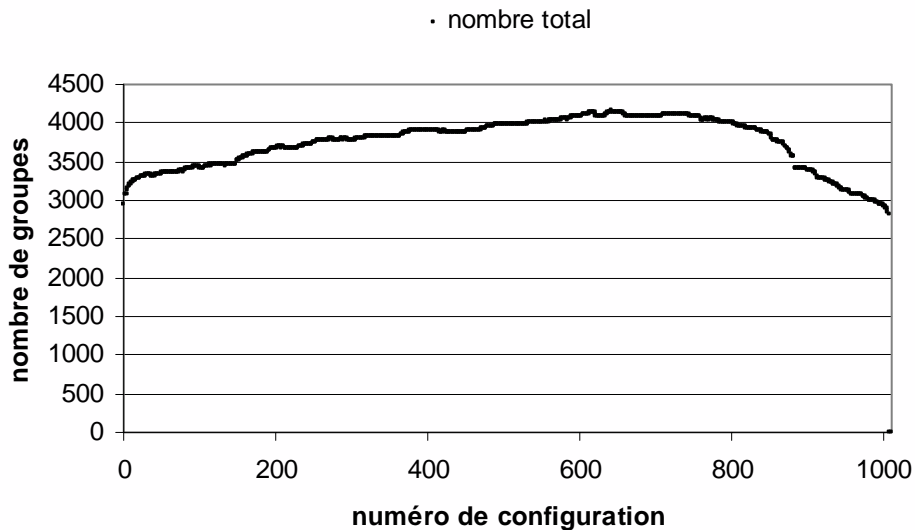


Figure 35 Evolution du nombre total de groupes durant l'heuristique

Nous avons également voulu vérifier l'importance de notre choix de ne regarder que le maximum des bornes calculées, ce qui nous a conduit à suivre l'évolution au cours de l'heuristique de la moyenne de ces bornes (pondérée par le nombre de flux).

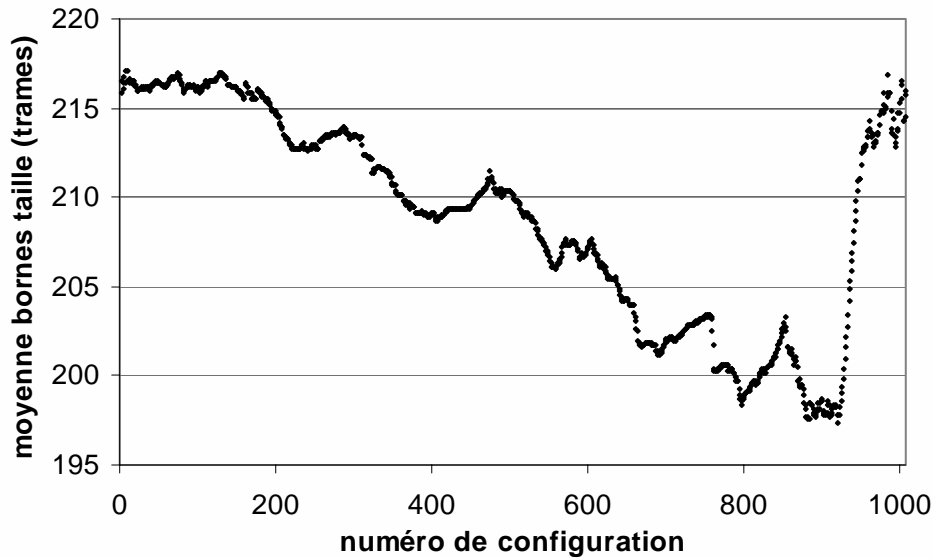


Figure 36 Evolution de la moyenne des bornes sur les tailles durant l'heuristique

On peut constater avec satisfaction que l'heuristique permet de diminuer le maximum des bornes calculées, sans pour autant en augmenter la moyenne.

Dans le même ordre d'idée, nous avons voulu voir si cette amélioration du maximum des bornes sur les tailles se faisait au détriment des bornes calculées sur le délai. Le graphique suivant montre l'évolution de notre second critère au cours de l'heuristique.

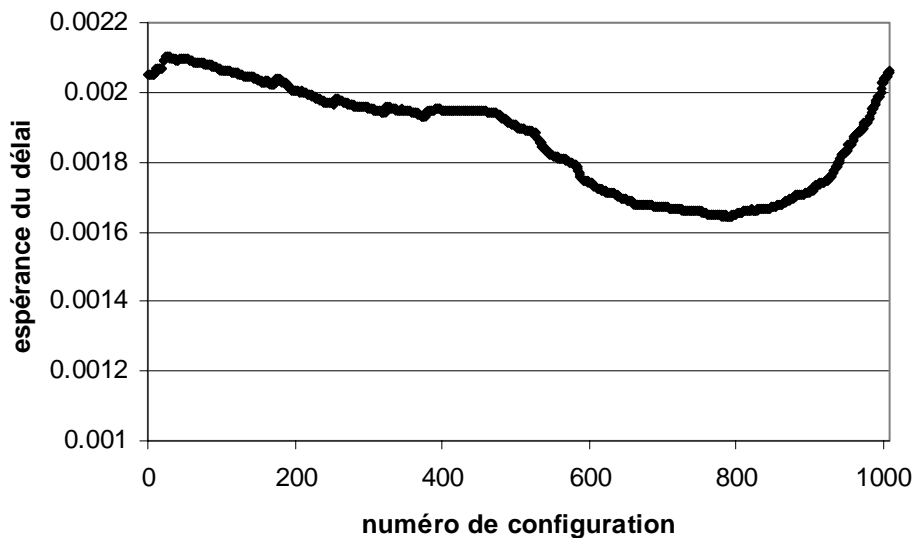


Figure 37 Evolution de l'espérance du délai durant l'heuristique

Encore une fois, on peut constater heureusement que l'heuristique ne dégrade pas les bornes calculées sur le délai.

En conclusion, l'heuristique  $S_{\min}$  que nous avons présentée permet de trouver des configurations bien meilleures que celle de référence où toutes les priorités sont fixées à basse. Cette heuristique présente également l'avantage de ne pas nuire aux bornes sur le délai, et de ne pas augmenter non plus la moyenne des bornes sur les tailles.

### 1.3.2 Autres heuristiques envisagées

Nous avons constaté qu'outre la taille minimale des trames, plusieurs autres facteurs influent sur le calcul des valeurs des configurations, comme l'effet de groupe ou la taille maximale des trames. Nous avons alors essayé d'appliquer le même type d'heuristique que précédemment, mais en modifiant le critère utilisé pour classer les VL.

#### Effet de groupe

Nous avons mis en avant l'importance dans l'heuristique  $S_{\min}$  des groupes qui sont scindés puis reconstitués. Ceci nous a donné l'idée d'utiliser comme critère de classement des VL le nombre de fois où un VL est associé aux autres dans des groupes. Plus formellement, on définit la « corrélation » du VL  $i$  avec le VL  $j$  comme étant le nombre de groupes dans le réseau où  $i$  et  $j$  apparaissent ensemble. Nous trions ensuite les VL d'après la somme de leurs corrélations avec tous les autres VL. Ainsi, nous faisons en sorte de déplacer en premier les VL qui sont très « groupés » avec d'autres ; on s'attend donc à ce que les valeurs des premières configurations se dégradent plus vite que précédemment, puisqu'on scinde un grand nombre de groupes dès le début de l'heuristique.

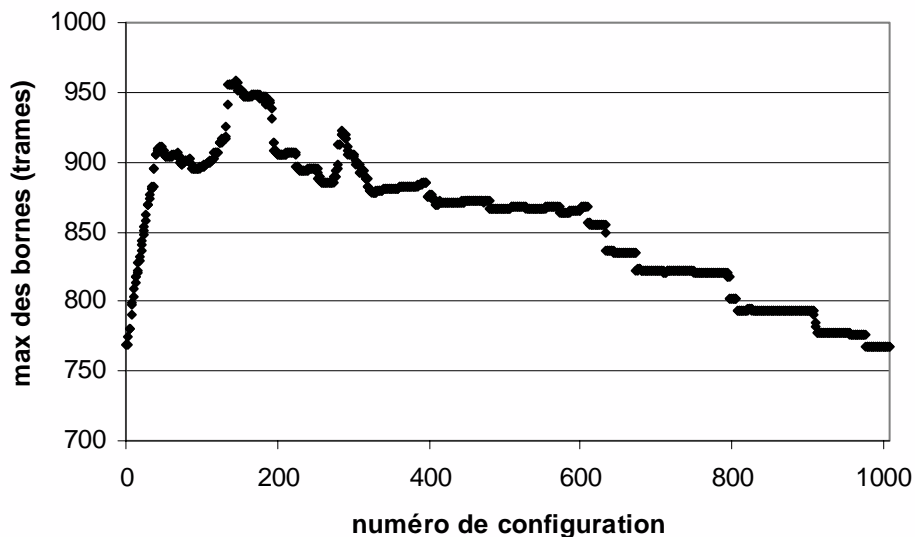


Figure 38 Evolution de la taille max durant l'heuristique utilisant l'effet de groupe

Les résultats de cette heuristique sont bien conformes à nos prédictions, en ce qui concerne l'allure générale de la courbe, ce qui conforte bien l'importance du facteur « effet de groupe » sur l'analyse (nous avons vérifié par ailleurs que déplacer les VL dans le sens contraire ne donne pas de meilleurs résultats). Cependant, cette heuristique ne nous permet pas d'exhiber de configuration meilleure que la configuration initiale.

#### Effet de $S_{\max}$

Un autre facteur important de la méthode d'analyse du réseau est la taille maximale des trames de chaque VL. En effet, ce paramètre sert à la définition des enveloppes initiales (on rappelle que l'ordonnée à l'origine de l'enveloppe initiale est fixée égale à ce  $S_{\max}$  pour chaque VL). Il sert également lors du calcul des courbes de service dans les multiplexeurs, car le traitement est non préemptif et donc les trames de priorité haute peuvent avoir à attendre la fin du traitement d'une trame de priorité basse. Nous avons donc fait une nouvelle expérience en triant les VL par valeur de  $S_{\max}$  décroissant.

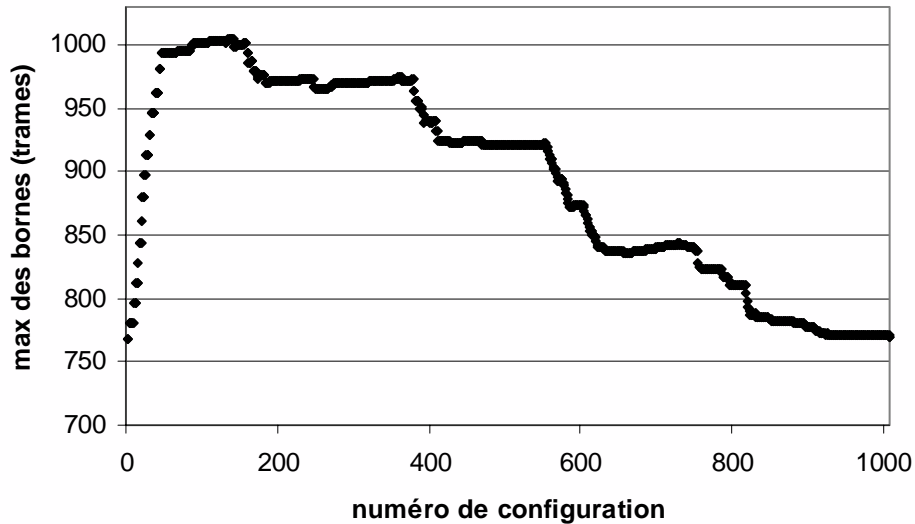


Figure 39 Evolution de la taille max durant l'heuristique  $S_{max}$

Comme dans le cas précédent, on constate que l'analyse est sensible à ce facteur  $S_{max}$ , mais qu'on ne peut pas en déduire une heuristique efficace pour réduire la borne maximale calculée.

### 1.3.3 Améliorations de l'heuristique $S_{min}$

Les expériences précédentes nous ont montré que le facteur  $S_{min}$  est prépondérant pour trouver de bonnes configurations. Elles ont également montré que les facteurs  $S_{max}$  et effet de groupe sont importants pour l'analyse. Nous avons alors décidé d'utiliser ces facteurs pour raffiner l'heuristique  $S_{min}$ . En effet dans celle-ci, les VL sont triés par ordre croissant de  $S_{min}$ , mais ceux de même  $S_{min}$  sont triés aléatoirement. Notre idée a donc consisté à séparer ces ex-aequo sur la base des deux autres critères. Nous représentons ci-dessous les résultats des deux expériences correspondantes.

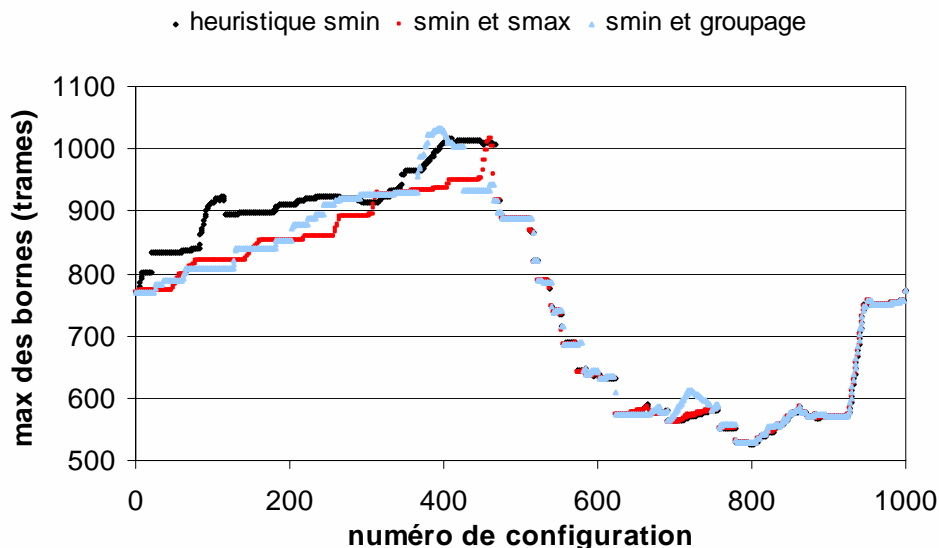


Figure 40 Evolution de la taille max pour d'autres heuristiques

Ces modifications portées à l'heuristique  $S_{\min}$  modifient surtout le comportement de celle-ci pour les premières configurations calculées. Malheureusement ce ne sont pas celles qui nous intéressent, car elles possèdent des valeurs assez élevées. Parmi les configurations obtenues, la meilleure améliore de moins de 1% celle obtenue par l'heuristique  $S_{\min}$ , ce qui ne représente pas un gain très intéressant.

#### 1.3.4 Synthèse

L'heuristique  $S_{\min}$  que nous avons présentée permet de trouver de bons individus, améliorant jusqu'à 30% le maximum des bornes calculées dans le réseau, sans pour autant en augmenter la moyenne ou les bornes sur les délais. Elle présente également l'avantage d'être relativement peu coûteuse en calculs, car il suffit d'analyser un millier de configurations différentes, ce qui représente une dizaine de minutes seulement. Par contre, raffiner cette heuristique pour obtenir des individus encore meilleurs est délicat, et n'apporte pas beaucoup de gains.

### 1.4 **Algorithmes d'évolution**

Devant la complexité du problème, nous avons pensé avoir recours à des algorithmes d'évolution, qui sont réputés pouvoir trouver des solutions même dans le cas d'espaces de solutions de très grande taille. De plus, dans notre cadre d'étude, il est relativement facile d'analyser une configuration donnée, puisque le calcul ne dure que quelques dixièmes de seconde.

Le principe général de ces algorithmes est d'imiter le processus de l'évolution naturelle, en espérant que la sélection naturelle au sein d'une population permettra d'obtenir (de converger vers) de bons individus. Nous présentons ici cette famille d'algorithmes et leur principe général, puis nous présenterons les adaptations nécessaires de ces algorithmes au contexte d'optimisation multicritères dans lequel nous nous situons. Les paragraphes suivants présenteront nos différentes expériences, ainsi que les résultats obtenus.

#### 1.4.1 Généralités sur les algorithmes d'évolution [45]

Cette famille d'algorithmes est relativement récente, bien que des travaux préliminaires aient été effectués dès les années 50 ([46][47][48][49]). La principale raison du développement actuel de ce type d'algorithme est la mise à disposition des chercheurs de moyens de calcul suffisamment puissants pour permettre l'utilisation pratique de ce genre d'approche, qui nécessite de nombreux calculs. Le principal avantage de ces algorithmes, outre leurs performances souvent intéressantes, est de pouvoir facilement s'adapter à de nombreux types de problèmes, ce qui justifie des applications de plus en plus nombreuses, y compris dans l'industrie.

On peut classer les algorithmes d'évolution en trois grandes familles, qui ont été à l'origine développées séparément, mais qui présentent de nombreux points communs.

Les algorithmes génétiques ont été présentés par Holland [50][51][52], avant d'être étudiés par de nombreux groupes de recherche. Leur principe est de faire l'analogie avec la théorie de la génétique, en manipulant des gènes qui décrivent les individus. Comme la plupart des algorithmes d'évolution, leur principal champ d'application réside dans les problèmes d'optimisation.

La programmation évolutive a été utilisée dans le cadre de la recherche sur l'intelligence artificielle, comme dans les travaux de Fogel [53][54] ou Burgin [55][56], Atmar [57]. Il s'agit principalement d'une extension des automates à éléments finis. Dans ce cadre, un automate composé d'un nombre fini d'états réagit à une séquence d'entrée, qui utilise un ensemble fini

de symboles. Selon le symbole lu, l'automate change d'état, conformément à un nombre fini de règles de transition, et produit des symboles de sortie. L'idée originelle était de faire évoluer l'automate, en modifiant ses états ou ses règles de transition, de sorte que les symboles de sortie prédisent les symboles suivant d'entrée, ou du moins de mesurer grâce à une fonction de comparaison l'écart entre la prédiction et la réalité.

La dernière famille est celle des stratégies d'évolution. Les travaux pionniers de ce domaine, comme ceux de Rechenberg [58][59] et Schwefel[60][61], ont présenté ce type de méthode d'optimisation de paramètres.

Le principal point commun de ces trois familles d'algorithmes réside dans leur structure. Celle-ci veut copier le comportement de l'évolution naturelle, et se sert pour cela du modèle darwinien classique. Dans celui-ci, un individu est soumis à la concurrence de ses semblables (nourriture, prédateurs, reproduction), ainsi qu'à son environnement (climat). Plus un individu est adapté à ces conditions, plus il a de chances de produire une descendance, qui héritera de ses caractéristiques. Au fil des générations, la population est alors de plus en plus constituée d'individus performants. Mais, la reproduction des individus étant soumise à des phénomènes aléatoires, il existe toujours un renouvellement de l'information génétique totale, qui permet de préserver la diversité de la population. Ce modèle est décrit dans l'algorithme suivant (tiré de [45]), où  $P(t)$  représente la population totale de la génération  $t$  (de taille  $\mu$ ),  $Q$  un ensemble d'individus choisis selon des critères qui varient suivant l'algorithme et  $P'(t)$  un ensemble de  $\lambda$  descendants de la population  $P(t)$ .

```
t = 0;
Initialiser P(t);
Évaluer P(t);
Tant que (!critère_de_termination) {
    P'(t) = variation(P(t));
    Évaluer P'(t);
    P(t+1) = sélection[P'(t) U Q(t)] ;
    t = t + 1;
}
```

Un algorithme d'évolution sera déterminé par la donnée de tous les opérateurs utilisés ci-dessus : représentation des individus, variation, évaluation, et sélection. C'est principalement dans ces opérateurs que l'on peut différencier les trois grandes familles présentées. Cependant, la majeure partie des implémentations actuelles utilise des approches hybrides, aussi les trois familles ont-elles tendance à être de moins en moins distinctes.

Le critère de terminaison peut être lié à la durée d'exécution de l'algorithme (1 heure de calcul, 1000 générations, etc.), ou bien à des critères portant sur la population. On peut décider d'arrêter l'algorithme lorsque tous les individus sont « bons » en moyenne, ou bien si on obtient un individu satisfaisant, etc.

L'évaluation des individus est généralement effectuée grâce à une fonction de coût, dont le bon choix détermine bien sûr le fonctionnement de l'algorithme, et son adéquation au problème étudié.

#### 1.4.1.1 Représentation des individus

En général, on cherche à étudier des objets appartenant au monde réel, comme par exemple des composants électroniques, des actionneurs, des produits chimiques, etc. Les différents paramètres de ces objets constituent ce qu'on appelle le phénotype. Les opérateurs dont nous avons parlé opèrent sur des représentations symboliques de ces objets, qui forment un espace

nommé génotype. L'opérateur de représentation des individus consiste donc à faire le lien entre phénotype et génotype. Deux types d'approche sont possibles pour cette représentation. On peut décider de coder les individus de façon à ressembler le plus possible aux paramètres des objets appartenant au phénotype. Cette approche court le risque d'utiliser dans les calculs des individus très complexes ; par contre, il est facile de voir l'individu correspondant dans le phénotype. A l'inverse, on peut choisir d'utiliser dans le calcul des individus de type bien connu, comme des chaînes de caractères ou des vecteurs de nombres flottants. L'inconvénient de ce type de représentation est de complexifier la correspondance avec les objets réels. Mathématiquement, selon la complexité de la fonction de codage/décodage, il peut apparaître des phénomènes perturbateurs comme des non linéarités ou des singularités. Dans le cas général, il n'existe pas de résultat théorique permettant d'affirmer la supériorité d'une approche par rapport à l'autre[45]. En pratique cependant, on a plus tendance à utiliser la deuxième approche, principalement pour les gains de temps de calcul qu'elle implique. Un autre facteur à ne pas négliger est qu'ainsi on peut réutiliser des algorithmes déjà écrits sur des objets de type bien connu.

Pour les algorithmes génétiques, la notion centrale est celle de gène. On va donc représenter un individu par une chaîne de caractère, censée représenter l'intégralité de son génome. On retrouve bien l'analogie avec la génétique, où les chaînes ADN sont représentées par la séquence de leurs peptides A T C G. De façon classique, on ne considère que des chaînes constituées de caractères binaires, 0 ou 1. Ceci peut conduire à des fonctions de décodage très complexes : dans le cas où il faut coder un vecteur de nombre réels appartenant à  $\mathbf{R}^n$ , on commence par scinder le gène en  $n$  fragments de longueur identique ; chacun de ces fragments correspond au codage classique d'un nombre flottant en binaire.

Pour les deux autres familles d'algorithmes, les individus sont représentés par des vecteurs de réels (stratégies d'évolution) ou des matrices de réels (programmation évolutive, la matrice représente les transitions entre états).

#### 1.4.1.2 Opérateurs de mutation

Les opérateurs de mutation font partie des opérateurs de variation. Leur rôle consiste à trouver un descendant à partir d'un individu donné, par analogie avec les mutations aléatoires du génome qui interviennent dans la nature.

Pour les algorithmes génétiques, cette opération revient simplement à modifier aléatoirement un des caractères de la chaîne. En général, on fixe la probabilité de mutation d'un caractère à une valeur basse, 0,0001 par exemple, ce qui est une valeur beaucoup plus forte que pour l'évolution naturelle, mais qui permet de voir rapidement des modifications intervenir au sein de la population. Certains travaux ont cependant montré que choisir une probabilité très forte à l'origine mais décroissante au fil des générations, pouvait permettre de converger plus rapidement vers de bons individus[62][63].

Dans le cadre de la programmation évolutive, la mutation peut être effectuée par la création ou suppression aléatoires d'états, ou bien la redéfinition de transitions.

Pour les stratégies évolutives, la mutation d'un individu représenté par le vecteur de réels  $x = [x_i]_{i \in 1..n}$  s'effectue généralement de la façon suivante :

$$x'_i = x_i + \sigma_i \cdot N_i(0,1)$$

Le vecteur  $\sigma$  est appelé vecteur des paramètres de la stratégie. La mutation consiste donc à ajouter à chaque composante  $x_i$  une valeur d'espérance nulle et de variance  $\sigma_i^2$ . De nombreux travaux récents [64][65] font de  $\sigma_i$  un des paramètres à optimiser par l'algorithme, ce qui porte le nom d'auto évolution.



#### 1.4.1.3 Opérateurs de recombinaison

Comme les opérateurs de mutation, les opérateurs de recombinaison sont des opérateurs de variation. Dans l'analogie avec l'évolution naturelle, ils représentent la reproduction sexuée, puisque leur rôle est de déterminer un descendant à partir de deux (ou plus) parents.

Pour les algorithmes génétiques, la recombinaison est le plus souvent effectuée en choisissant aléatoirement un point de croisement des gènes (position dans la chaîne). Le gène résultant est obtenu en concaténant la partie gauche du gène d'un parent avec la partie droite de celui de l'autre parent.

Pour les stratégies d'évolution le choix est très ouvert. On peut choisir des opérateurs qui sélectionnent aléatoirement les composantes de l'un ou l'autre des parents, ou bien utiliser des moyennes entre ces composantes.

Des travaux récents ont montré que l'utilisation de plusieurs parents permet d'accélérer la recherche de bons individus [66]-[68], mais la compréhension du fonctionnement de ces opérateurs de recombinaison est encore un problème de recherche très ouvert.

#### 1.4.1.4 Sélection

Contrairement aux opérateurs précédents, la sélection ne s'appuie pas sur la représentation des individus dans le génotype, mais sur leur valeur calculée grâce à la fonction d'évaluation choisie. On dispose de nombreuses méthodes pour choisir certains individus, pour la recombinaison par exemple. Une méthode consiste à assigner à chaque individu une probabilité de sélection directement proportionnelle à sa valeur. Une autre assigne cette probabilité en fonction du classement de l'individu dans la population totale (le meilleur individu a plus de chances d'être choisi que les autres, et ainsi de suite). Une méthode fréquemment utilisée, car elle est efficace et très simple à programmer, est celle du tournoi. Dans celle-ci, on choisit au hasard un certain nombre d'individus, et on choisit parmi ceux-ci le meilleur. On reconduit cette méthode jusqu'à avoir sélectionné le bon nombre d'individus. Le nombre d'individus choisis pour chaque tournoi permet de fixer le niveau d'exigence de la sélection : plus il sera important, meilleurs seront les individus choisis.

#### 1.4.1.5 Quelques perspectives

Plusieurs pistes ont été envisagées pour améliorer ces différents algorithmes. On pourrait par exemple prendre en compte l'âge des individus au moment de choisir les individus pour la reproduction. Une autre approche serait d'effectuer en parallèle l'évolution de plusieurs races, ce qui impliquerait de nouvelles interactions. Il serait également possible de considérer des générations décalées, ou bien des évolutions de populations en des endroits différents, avec des effets liés aux migrations (ceci permettrait de répartir le calcul). Enfin, certains envisagent d'utiliser un mode de représentation ou une fonction d'évaluation qui évolueraient en fonction du temps.

#### 1.4.1.6 Synthèse

Le principal point fort des algorithmes d'évolution réside dans la quantité de choix disponibles, à chaque niveau de la démarche. Cela permet d'obtenir facilement un algorithme très adapté au problème étudié. Par rapport aux algorithmes d'optimisation classique, des études [69] ont montré que les résultats sont souvent aussi bons, mais qu'en plus les algorithmes d'évolution s'appliquent à des cas pour lesquels il n'existe pas de méthode classique.

Comme souvent, cette force constitue également la grande faiblesse de ces algorithmes. En effet, le grand nombre de paramètres à fixer les rend difficiles à utiliser, d'autant plus qu'il manque encore de connaissances théoriques sur leur fonctionnement (ce qui permettrait de mieux choisir les bons paramètres et opérateurs).

### 1.4.2 Généralités sur les algorithmes multicritères [70]

Le fait que les algorithmes d'évolution donnent de bons résultats pour des problèmes d'optimisation complexes en a fait des candidats de choix pour les problèmes d'optimisation multicritère. Contrairement aux algorithmes précédents où la fonction d'évaluation jugeait les individus selon un critère prédéfini, dans cette classe de problèmes les individus sont jugés selon plusieurs critères. Dans le reste de cet exposé, nous nous contenterons du cas de deux critères, sans perte de généralité. On se trouve immédiatement confronté au problème suivant : soient deux individus A1 (valeurs  $u_1$  et  $v_1$ ) et A2 ( $u_2$  et  $v_2$ ), comment les comparer ? On utilise la notion de domination au sens de Pareto : A1 domine A2 si et seulement si  $u_1 > u_2$  et  $v_1 > v_2$ . On constate immédiatement que cette relation n'est pas une relation d'ordre total, puisque deux individus peuvent très bien ne pas être comparables avec cette relation. Nous illustrons cette notion dans le schéma suivant, qui représente l'espace des objectifs : chaque point représente un individu et a comme coordonnées les valeurs de l'individu par rapport aux différents critères.

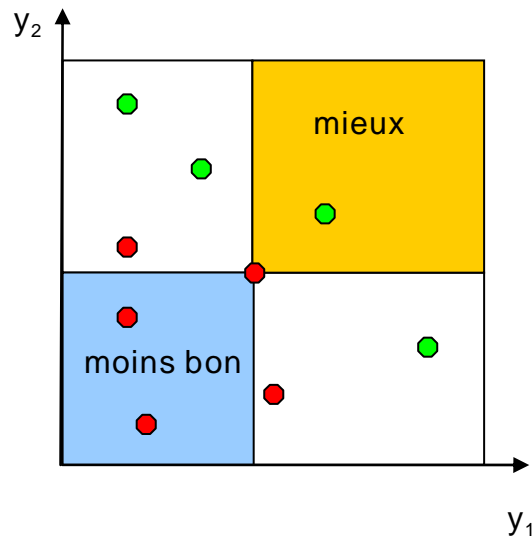


Figure 41 Plan des objectifs, notion de « mieux »

Plutôt que la recherche d'un individu optimal, le but de la recherche multicritère est remplacé par la recherche du front de Pareto. Le front de Pareto est une courbe dans le plan des objectifs, qui constitue la frontière entre les individus dominés et non dominés : si un individu se situe sur le demi-plan supérieur, il sera non dominé. Au sein d'une population de taille donnée, l'ensemble des individus non dominés fournit une estimation du front de Pareto.

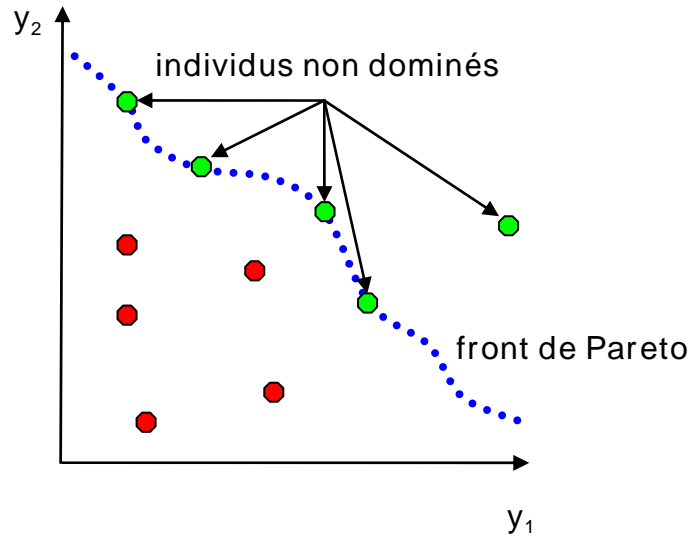


Figure 42 Evocation du front de Pareto

La signification physique de ce front de Pareto est évidente : il s'agit de la courbe d'échange entre les deux critères. Elle indique par exemple, si on veut gagner sur le critère 1, de combien on va perdre sur le critère 2. La connaissance de ce taux d'échange est souvent primordiale lors des phases de conception, pendant lesquelles on cherche le meilleur compromis entre plusieurs critères.

Cet aspect multicritère va créer plusieurs difficultés pour les algorithmes d'évolution, et nous détaillons les principales dans les trois paragraphes suivants.

#### 1.4.2.1 Evaluation des individus

Dans le cas des algorithmes monocritères, il est facile de juger de la qualité d'un individu donné. Quand plusieurs critères sont présents, il faut déterminer une méthode permettant de juger cette qualité, en tenant compte des différents critères, notamment pour les phases de sélection. On distingue généralement trois méthodes.

La première méthode permet de se ramener au cas précédent, puisqu'il s'agit simplement de faire une moyenne pondérée des valeurs sur chacun des critères. Pour conserver l'aspect multicritère, les poids utilisés varient au cours de l'évolution.

La deuxième méthode est de regarder les critères à tour de rôle : le premier individu est choisi en fonction d'un critère, le second en fonction d'un autre critère, etc. Une variante de cette méthode consiste à affecter à chaque critère une probabilité d'être le critère pris en compte pour le choix de l'individu.

La dernière famille de méthodes utilise la notion de domination au sens de Pareto. La qualité d'un individu peut être établie en comptant combien d'individus il domine, ou combien le dominant, ou alors en tenant compte de ces deux chiffres, comme par exemple dans l'algorithme SPEA2 [74] que nous détaillons plus loin. On remarque que les deux dernières méthodes jugent la qualité d'un individu par rapport à la population totale, alors que la première fixe cette qualité uniquement sur la base des différents critères.

#### 1.4.2.2 Préservation de la diversité

Un des points clés des méthodes multicritères est de préserver la diversité génétique au sein de la population. Ceci permet en principe d'éviter que tous les individus ne ressemblent au meilleur, c'est-à-dire que la population ne converge que vers un point du front de Pareto, au lieu de le décrire. La technique utilisée dans la plupart des algorithmes consiste à diminuer la

probabilité de sélection d'un individu (en diminuant sa qualité calculée précédemment), s'il est dans une zone « dense » de la population. On distingue ici aussi trois grandes méthodes permettant d'estimer cette densité.

La première est la plus consommatrice de puissance de calcul : pour estimer la densité autour d'un individu donné elle commence par calculer toutes les distances (dans l'espace des objectifs) de cet individu à tous les autres. Ces distances sont alors utilisées comme arguments d'une fonction « densité ». Ce peut-être une simple moyenne, ou d'autres fonctions plus complexes.

La deuxième méthode est celle du plus proche voisin. Ici encore, on calcule toutes les distances, et on estime la densité par la distance au  $k^{\text{ème}}$  voisin le plus proche. En général,  $k$  est fixé à 1.

La dernière méthode consiste à découper l'espace des objectifs en petites zones, généralement grâce à une grille régulière. La densité locale est estimée simplement en comptant combien d'individus se trouvent dans la même case.

#### 1.4.2.3 Elitisme

Au cours de l'évolution, on ne souhaite pas perdre de bons individus. Ceci implique en général le stockage de ces individus dans une archive, et la consultation de cette archive au moment de constituer les nouvelles populations. Le problème se pose lorsqu'on considère que pour des raisons d'efficacité d'implémentation, cette archive doit être de taille limitée. Pour cette raison, on se limite généralement à ne stocker que des individus non dominés. Mais cette précaution est loin d'être suffisante, puisque il peut en principe exister un nombre infini d'individus non dominés. Un tri de l'archive est donc nécessaire à chaque génération. Pour ce faire, les algorithmes peuvent par exemple utiliser des informations sur la densité : si deux individus sont proches, on peut ne garder qu'un des deux. Malheureusement, cette méthode peut conduire à une perte de diversité génétique, car ce n'est pas parce que deux individus ont des valeurs proches pour les critères choisis, que leurs gènes se « ressemblent ». D'autres algorithmes utilisent « l'âge » des individus comme critère de choix, en décidant de ne conserver que les individus les plus récemment ajoutés dans l'archive, ou au contraire les plus anciens.

Cette gestion de l'archive reste un point très délicat des algorithmes multicritères, car il est source d'une perte de diversité génétique, et qu'il peut également introduire l'effet suivant : un individu présent dans l'archive à la génération  $t$  peut être dominé par un individu ayant été dans l'archive au temps  $t' < t$ , mais ayant été supprimé depuis. Les performances d'un algorithme multicritère sont donc très souvent mesurées au regard de la qualité de leur gestion d'archive [71][72].

### 1.4.3 Application des algorithmes génétiques multicritères au problème étudié

Pour développer notre application d'un algorithme d'évolution adapté à notre contexte, nous avons retenu l'architecture présentée par Bleuler, Laumanns, Thiele et Zitzler sous le nom de projet PISA [73]. Cette architecture a été construite sur la base de deux constatations : d'une part, compte tenu de la complexité importante des algorithmes utilisés, il devient difficile pour un non spécialiste de développer et d'utiliser ce type d'approches, pour résoudre un problème donné. D'autre part, les développeurs d'algorithmes d'évolution ont du mal à trouver des problèmes concrets leur permettant de tester les performances de leurs algorithmes. Les auteurs de [73] ont donc pris le parti de créer une interface simple pour relier ces deux mondes. Ils séparent d'un côté ce qui dépend directement du problème étudié, c'est-à-dire les opérateurs de mutation, de recombinaison, ainsi que la fonction d'évaluation, et de l'autre ils placent tout ce qui est indépendant de la nature du problème, comme la sélection

qui n'utilise que les valeurs calculées pour les individus. L'idée est donc que les spécialistes d'optimisation peuvent développer des modules spécifiques indépendants du problème, que les développeurs d'applications n'auront plus qu'à connecter au module représentant leur problème. Le schéma ci-dessous, tiré de [73], illustre cette architecture.

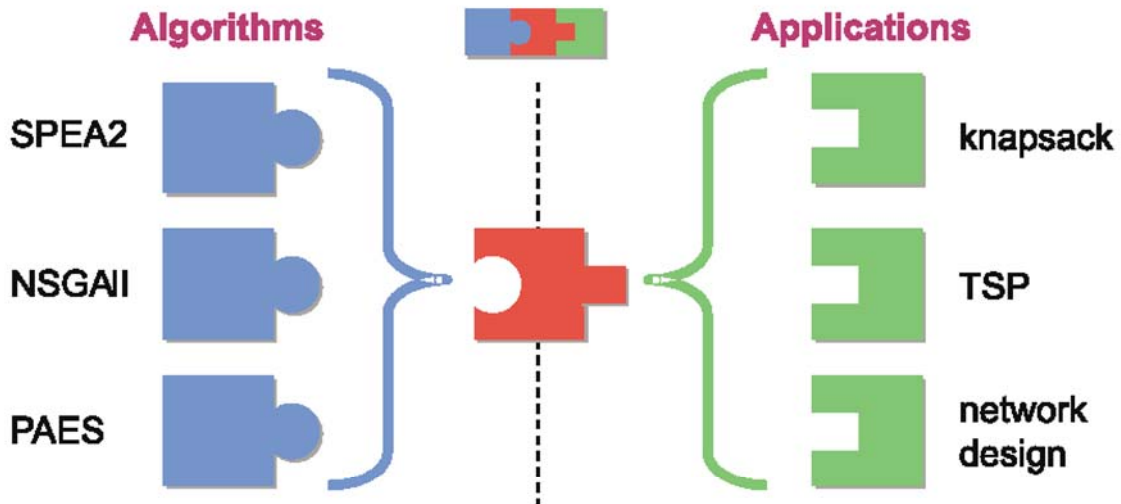


Figure 43 Architecture PISA

L'architecture ainsi développée présente l'avantage de permettre la réutilisation de modules existants, avec une certaine souplesse. Le lien entre les deux modules est effectué grâce à une interface simple (mode texte). Les données échangées sont soit des données de contrôle (par exemple : « le variateur a fini, c'est au tour du sélecteur »), soit des données relatives aux individus (valeurs pour chaque critère, index des individus choisis dans la population). Certains paramètres communs aux deux modules choisis (taille des populations, archives, pression sélective) sont regroupés au sein d'un fichier texte supplémentaire.

#### Paramètres globaux choisis

Les paramètres nécessaires dans notre cas ont été fixés comme suit :

Paramètre	Description	Valeur choisie
$\alpha$	Taille de la population initiale et de l'archive	1000
$\mu$	Nombre d'individus sélectionnés pour la reproduction	100
$\lambda$	Nombre de nouveaux individus à chaque génération	10
P	Pression sélective (nombre d'individus dans chaque tournoi)	3
Dim	Dimension du problème : nombre de critères d'évaluation des individus	2
Max_gen	Nombre maximal de générations	15000

Il s'agit ici des valeurs fixées en général dans toutes nos expériences. Cependant, il nous a parfois été nécessaire d'utiliser des valeurs différentes, par exemple pour juger de l'influence respective de ces paramètres.

La taille de l'archive retenue peut sembler faible au regard de la complexité du problème, puisque elle est même inférieure à la longueur des gènes. Cependant, certaines études montrent que les algorithmes d'évolution sont peu sensibles à cette taille d'archive. De plus, une taille de population relativement faible rend l'occupation mémoire du programme plus raisonnable, et l'exploitation des résultats plus simple. Nous avons choisi de sélectionner pour la reproduction une portion importante de la population (10%), mais de n'utiliser que 20% de ces individus pour la reproduction (deux parents pour un enfant, choisis aléatoirement parmi  $\mu$ ). Nous pensons ainsi permettre de préserver la diversité de la population en ne sélectionnant pas exclusivement les meilleurs individus. Pour la même raison, nous avons choisi une pression sélective relativement faible. Ces choix ont été établis en fonction de notre compréhension du problème ; l'article [45] déclare d'ailleurs qu'il n'existe pas de « guide théorique du choix des paramètres », car ce choix est intimement lié au problème étudié.

### Sélecteur choisi

Pour notre application, nous avons choisi pour le sélecteur l'algorithme SPEA2 [74]. En effet, une étude de plusieurs algorithmes d'optimisation multicritère [72] a conclu qu'il était aujourd'hui le meilleur sur la plupart des critères : préservation de la diversité, élitisme, distance moyenne au front de Pareto.

On rappelle ici les principales caractéristiques de cet algorithme :

- la qualité d'un individu est jugée d'après des critères de domination dans la population globale. Plus précisément, on affecte à chaque individu  $i$  une valeur  $S(i)$ , qui représente le nombre d'individus de la population qui sont dominés par  $i$ . La qualité d'un individu  $j$  est alors obtenue en additionnant les  $S(i)$  pour tous les individus  $i$  qui dominent  $j$ .
- l'estimation de la densité est effectuée par une méthode inspirée de celle de plus proche voisin : les distances à tous les voisins sont calculées, et on ajoute à la qualité de l'individu (décrite dans le point précédent) l'inverse de la distance au  $k$ -ième voisin (en général,  $k$  est fixé à 1).
- l'algorithme utilise une archive d'individus de taille fixe. En cas de besoin, le tri de l'archive est effectué en enlevant l'individu qui a le plus proche voisin (en cas d'égalité, on regarde le second plus proche voisin).

Le déroulement général de SPEA2 est explicité par l'algorithme suivant :

Données :  $M$  taille de la population engendrée,  $N$  taille de l'archive,  $T$  nombre maximum de générations.

Etape 1 : Initialisation. Création d'une population initiale  $P_0$ , d'une archive initiale vide  $A_0$ .

Etape 2 : Evaluation des individus. On calcule la qualité de chacun des individus (population  $P_t$  et archive  $A_t$ ) comme indiqué ci-dessus.

Etape 3 : Sélection. Recopie de tous les individus non dominés de  $P_t$  et  $A_t$  dans  $A_{t+1}$ . Si la taille de  $A_{t+1}$  est supérieure à  $N$ , on la trie selon le critère cité précédemment. Si elle est inférieure à  $N$ , on la remplit d'individus dominés.

Etape 4 : Terminaison. Si un critère de terminaison est vérifié, ou que le nombre de générations calculées est supérieur  $T$ , fin de l'algorithme. Le résultat est formé

de l'ensemble des individus non dominés de  $A_{t+1}$ .

Etape 5 : Sélection pour la reproduction. On choisit un ensemble de « parents » par tournois successifs au sein de  $A_{t+1}$ .

Etape 6 : Variation. On applique les opérateurs de recombinaison et de mutation sur l'ensemble des parents choisis. Les individus résultants constituent  $P_{t+1}$ .  $t = t + 1$ , et on boucle sur l'Etape 2.

#### Variateur choisi

Conformément à l'architecture PISA, notre variateur représente notre problème. Nous avons fait les choix suivants :

- les individus représentent des configurations différentes, c'est-à-dire qu'une configuration correspond à un choix de priorités pour tous les VL du réseau. Chaque individu est représenté par son gène, codé dans notre application par une chaîne de caractères « 1 » ou « 2 ». Les VL sont ordonnés lors de l'initialisation de l'algorithme, et un « 1 » en  $i^{\text{ème}}$  position indique que le VL numéro  $i$  est de priorité haute. On obtient des gènes de longueur le nombre de VL dans le réseau, soit 1008.
- Chaque individu est analysé par une nouvelle exécution de notre programme d'analyse. Les critères de mesure retenus sont ceux présentés plus tôt, c'est-à-dire d'une part la plus grande borne sur les tailles de files d'attente calculées (pour tous les multiplexeurs du réseau), et d'autre part l'espérance des bornes sur le délai calculées dans tout le réseau.
- La variation proprement dite est effectuée parmi les  $\mu$  individus choisis pour la reproduction, en appliquant  $\lambda$  fois l'opérateur de recombinaison. Pour les  $\lambda$  individus ainsi obtenus, on fixe une probabilité de mutation 0,4.
- l'opérateur de mutation choisi est classique : pour un individu donné, la mutation est effectuée en modifiant les priorités de 2 pourcents des VL, choisis aléatoirement.
- l'opérateur de recombinaison est également classique : A partir de 2 individus parents, on obtient un nouvel individu en concaténant les parties droites d'un parent et gauche de l'autre parent, avec un point de croisement aléatoire (l'index dans la chaîne où l'on coupe les gènes).
- Le choix de la population initiale est très important, comme nous l'illustrerons par la suite. Dans un premier temps, les  $\alpha$  premiers individus sont obtenus en tirant aléatoirement des chaînes de caractères binaires de la bonne longueur (celle du génotype).

#### 1.4.4 Résultats obtenus sans choix de la population initiale

Nous présentons ci-dessous les résultats de notre première utilisation de l'algorithme ainsi constitué pour déterminer de bonnes affectations de priorité.

Chaque point représente une configuration différente, avec en abscisse l'espérance du délai, et en ordonnées la borne maximale calculée (en trames). Les individus appartenant à une même génération sont représentés de la même couleur ; les nuages de points représentent donc les populations à des instants différents. Dans notre application, nous avons choisi de minimiser les deux critères, les meilleurs individus sont donc dans le coin inférieur gauche du graphique.

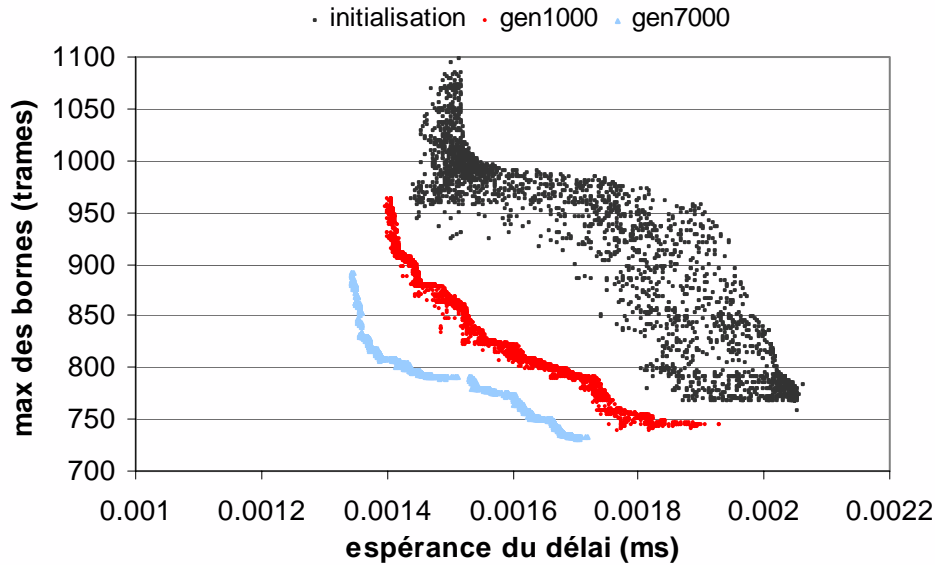


Figure 44 Evolution de la population au cours de l'algorithme

La première constatation qu'on peut formuler à propos de ce graphique est que l'algorithme converge bien, vers une population restreinte, composée d'individus nettement meilleurs que ceux initialisés aléatoirement. Pour confirmer cette impression, nous présentons sur le graphique suivant l'évolution de la « dispersion » des nuages au fil des générations. (On définit cette dispersion comme la somme des distances euclidiennes au carré de chaque point à tous les autres.)

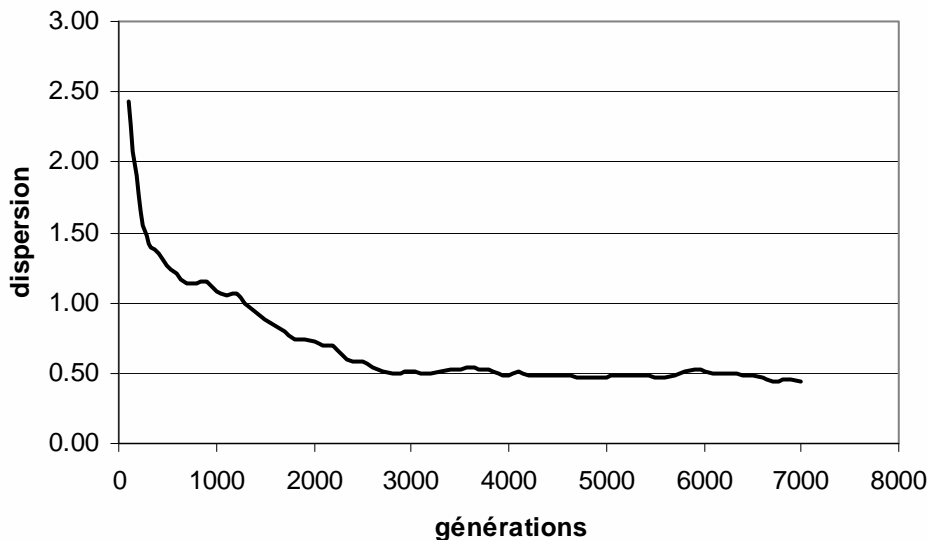


Figure 45 Evolution de la dispersion de la population au cours de l'algorithme

On peut bien vérifier avec cette courbe que la population est de plus en plus concentrée. Les performances moyennes des individus sur le premier critère (taille maximale) sont présentées sur le graphique suivant :



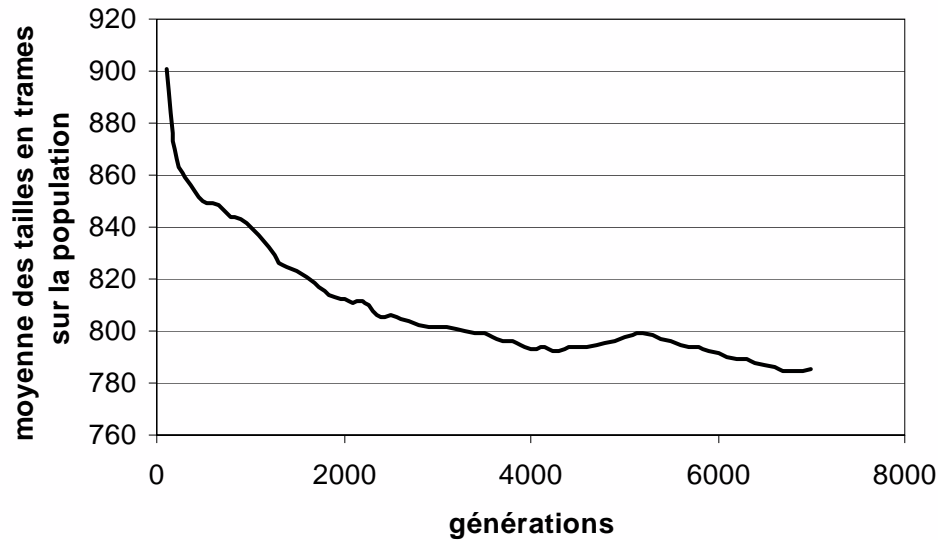


Figure 46 Evolution de la valeur moyenne des individus au cours de l'algorithme

On peut observer sur ce graphique que les individus des dernières populations sont en moyenne 15% meilleurs que ceux de la population initiale en ce qui concerne le premier critère (on obtiendrait le même ordre de grandeur sur le deuxième critère).

Ceci nous permet de conclure que notre algorithme fonctionne correctement, puisqu'il permet d'obtenir de bons individus, et qu'il converge de façon satisfaisante.

Cependant, ces résultats nous semblent insuffisants, dans le sens où ils ne nous permettent pas d'exhiber des individus aussi performants que ceux trouvés par l'heuristique  $S_{\min}$ , présentées dans la section 1.3.1. On rappelle en effet que cette méthode nous a fait découvrir des individus qui auraient sur le premier graphique les coordonnées (0.0016 ; 524). Les individus de la population finale sont même équivalents à l'individu de référence, pour lequel tous les VL ont une priorité basse : (0.002 ; 768).

Nous avons interprété ce manque de performance comme une perte de diversité dans nos populations, ou du moins un manque de recherche d'innovations génétiques : il semblerait que notre algorithme ait convergé vers des « points attracteurs » de l'espace des solutions.

Pour essayer de contourner ce problème, nous avons pensé à augmenter les paramètres responsables de la création de nouveaux individus : probabilité de mutation, sévérité de ces mutations, nombre de descendants par génération. Le graphique suivant présente les résultats obtenus en utilisant ce nouveau jeu de paramètres :

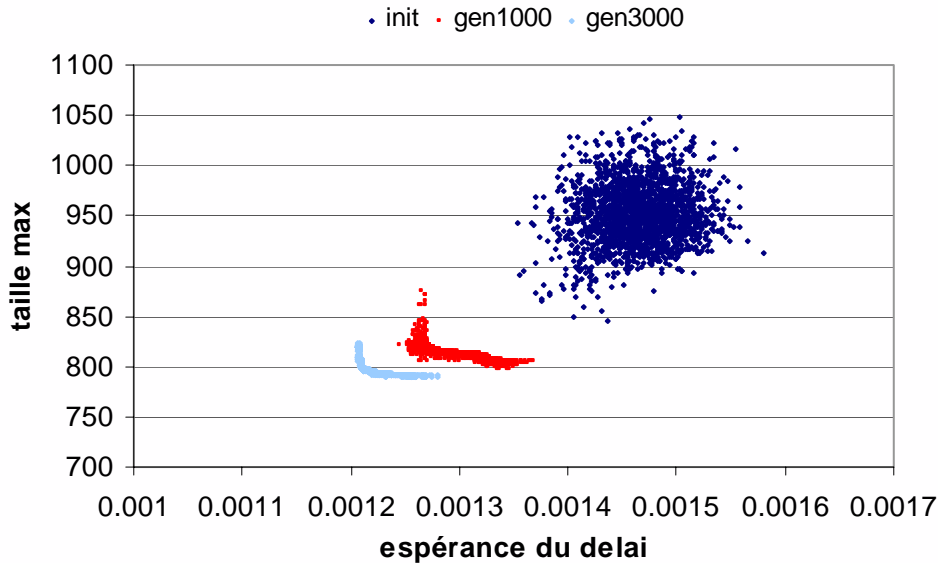


Figure 47 Evolution de la population au cours de l'algorithme, nouveaux paramètres

Il apparaît que les résultats obtenus avec ces paramètres ne sont pas meilleurs que dans le cas précédent. En fait, nous avons déduit que l'impact de cette modification des paramètres a simplement permis d'accélérer la convergence de l'algorithme, sans pour autant améliorer la valeur des individus des dernières générations. On peut constater cette accélération de convergence en reprenant l'évolution de la moyenne des valeurs sur le premier critère, au fil des générations :

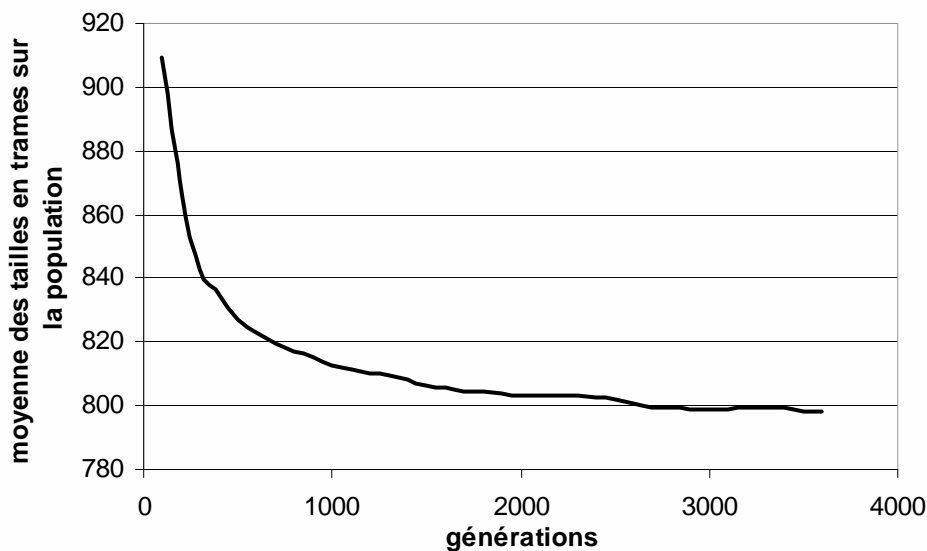


Figure 48 Evolution de la valeur moyenne des individus avec ces nouveaux paramètres

On constate immédiatement une convergence bien plus rapide, puisqu'on se rapproche presque deux fois plus vite que précédemment d'une valeur supérieure à 1% à la limite. Nous avons alors tenu à vérifier que le fait d'optimiser par rapport à deux critères ne limite pas la performance de l'algorithme par rapport à l'un d'entre eux. Nous avons donc choisi d'utiliser notre algorithme pour optimiser seulement la borne maximale sur les tailles de files d'attente.

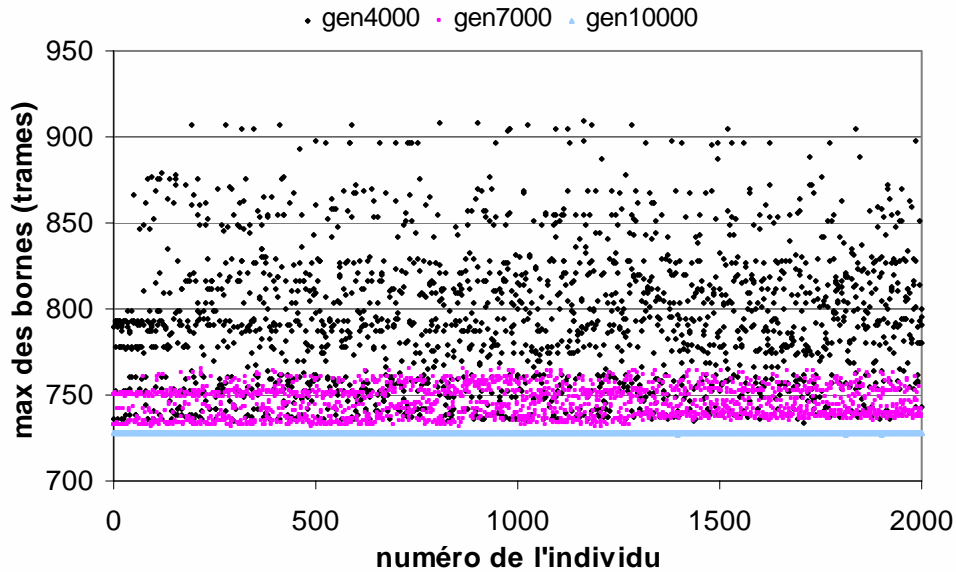


Figure 49 Evolution de la population au cours de l'algorithme mono-objectif

Encore une fois, on constate que l'algorithme converge vers des individus meilleurs que les individus initiaux, par rapport au seul critère retenu. La valeur de ces individus, proche de 727, est légèrement meilleure que dans le cas de la recherche multicritère, d'environ 5%. On touche là une limite de l'algorithme multicritère : il n'a pas permis de découvrir des individus de ce type, c'est-à-dire légèrement meilleurs pour un critère donné, mais bien moins bons par rapport à l'autre (on peut le vérifier par ailleurs). Cependant, même en optimisant par rapport à un seul critère, on n'est pas parvenu à découvrir des individus aussi bons que ceux exhibés par l'heuristique  $S_{\min}$ .

Nous concluons de ces expériences que l'algorithme, initialisé avec des individus choisis aléatoirement, tend à converger vers des individus dont les valeurs ne sont pas excellentes ; il ne permet pas de se diriger au sein de l'espace solution vers de meilleurs individus. Nos tentatives de modifier le comportement de l'algorithme, en changeant les opérateurs de variation, ou en essayant de favoriser un critère par rapport à l'autre, n'ont pas permis de mieux diriger l'algorithme, ce qui semble indiquer que les « bons » individus occupent une portion très restreinte de l'espace solution. Compte tenu de la complexité du problème, il ne nous a pas semblé abordable d'étudier les propriétés topologiques de l'espace solution, de manière à « guider » les recherches de l'algorithme.

#### 1.4.5 Résultats obtenus avec population initiale choisie

Il nous alors semblé que la meilleure façon de guider l'algorithme était de ne pas le laisser chercher « en aveugle », mais d'y inclure nos connaissances sur le fonctionnement du réseau. Pour ce faire, nous avons choisi de jouer sur la population initiale. L'idée est de donner à l'algorithme des individus initiaux qui possèdent déjà de bonnes propriétés. On sait en effet qu'une des caractéristiques de l'algorithme est de favoriser les individus performants, et de faire en sorte de répandre dans la population entière leurs propriétés.

Nous avons donc choisi de remplir la population initiale avec les configurations étudiées dans l'heuristique  $S_{\min}$ , parmi lesquelles on sait se trouver de très bonnes configurations. On complète cette population initiale avec des individus dont les gènes sont choisis aléatoirement.

Les résultats obtenus sont représentés sur le graphique suivant.

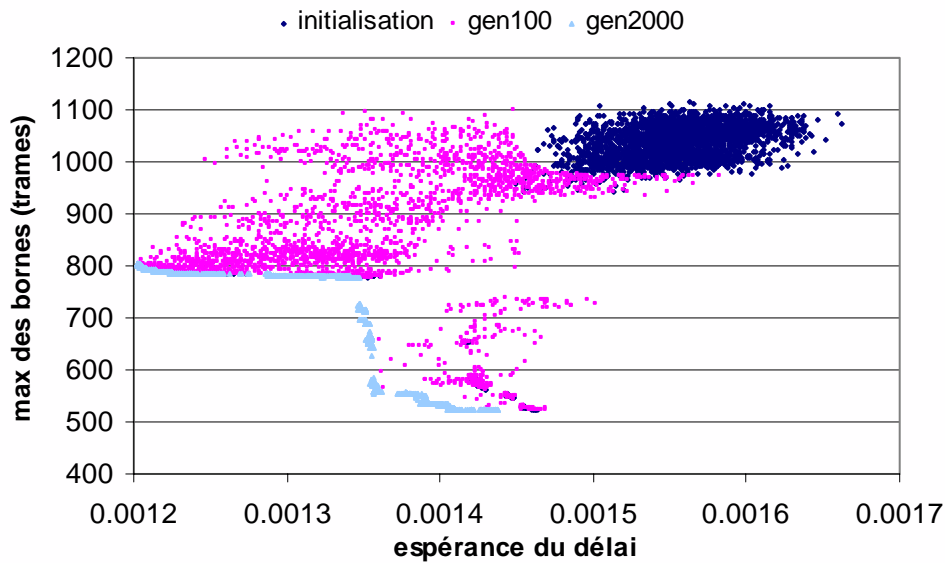


Figure 50 Evolution de la population au cours de l'algorithme, initialisation choisie

On voit facilement sur ce graphique l'évolution des populations au cours de l'algorithme, et surtout comment les bonnes propriétés des quelques bons individus initiaux se propagent au sein de la population. Conformément à nos attentes, la population finale est constituée d'individus bien meilleurs que dans l'expérience précédente, où la population initiale était choisie aléatoirement : on constate en effet des gains jusqu'à 25% pour le critère de borne maximale. Pour mieux voir les individus de la population finale, nous effectuons un zoom sur la partie inférieure gauche du graphique, en ne représentant que la première et la dernière génération.

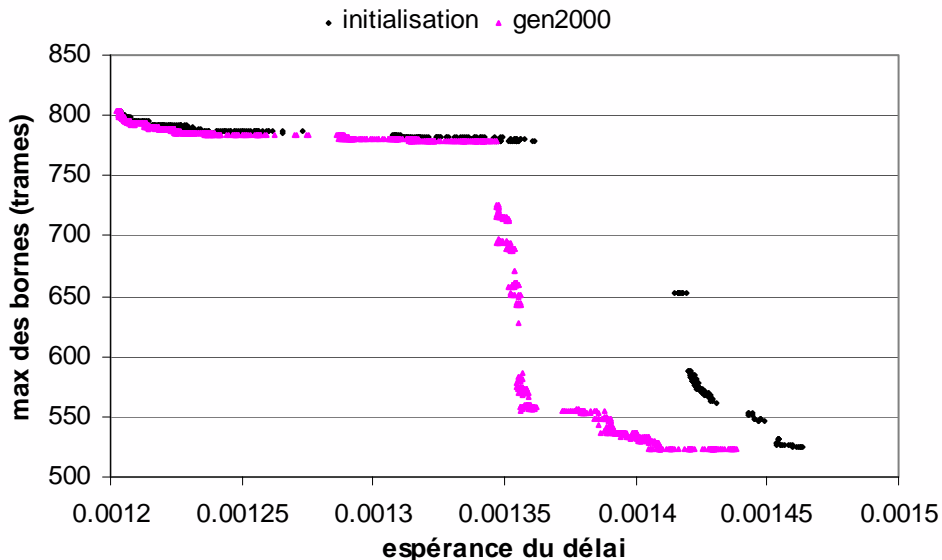


Figure 51 Zoom sur la population finale

On voit assez distinctement deux types de points sur ce graphique. Le premier groupe est formé d'individus pour lesquels l'espérance du délai est inférieure à 0.00135 ms. Ils ont en général des valeurs sur l'autre critère comprises entre 780 et 800 trames. Le « taux

d'échange » entre les deux critères semble ne pas être très important, puisqu'il est possible d'améliorer l'espérance du délai sans trop modifier la borne maximale calculée. On remarque que ce groupe est formé en grande partie des individus initiaux, ce qui indique que notre algorithme n'a pas trouvé de configurations bien meilleures que celles données par l'heuristique. La seconde catégorie d'individus regroupe ceux qui sont meilleurs pour le critère des tailles. Le « taux d'échange » est plus difficile à définir ici, mais on constate bien que pour gagner sur un critère il faut perdre sur le second. On remarque également qu'on n'a pas pu trouver d'individu significativement meilleur que ceux donnés par l'heuristique, vis-à-vis de ce critère de taille, mais par contre on trouve des individus tout aussi bons pour ce critère, mais 5% meilleurs pour l'autre critère. L'allure générale de la population finale nous donne une bonne idée du front de Pareto de ce problème, et peut permettre de mieux comprendre quels sont les termes de l'échange entre diminuer la borne maximale calculée et augmenter le délai moyen dans le réseau.

#### 1.4.6 Apport des algorithmes génétiques au problème

Les algorithmes d'évolution se posent a priori comme de bons candidats pour trouver des solutions à des problèmes d'optimisation multicritères de forte complexité. L'application d'une de ces techniques à notre problème nous a cependant confrontés aux problèmes connus de ce type de démarche : convergence de l'algorithme vers des bassins d'attraction, difficulté de régler les différents paramètres utilisés. Nous avons montré toutefois que le fait de guider l'algorithme en lui indiquant dès l'origine quelques bonnes solutions permet d'en améliorer grandement les performances. Au final, ce couplage heuristique  $S_{\min}$  + algorithme génétique se présente à nous comme l'outil le plus efficace pour découvrir de bonnes solutions à notre problème. Outre le fait qu'il est plus performant que nos différentes optimisations de l'heuristique  $S_{\min}$ , il permet d'esquisser l'allure du front de Pareto, ce qui peut s'avérer très utile pour le responsable du réseau. Cela peut en effet lui permettre de choisir en toute connaissance de cause un compromis entre les deux critères.

### 1.5 **Synthèse sur l'affectation des priorités aux VL**

Le problème du choix des priorités pour le millier de VL présents dans le réseau est un problème extrêmement complexe, de par sa taille mais aussi à cause des multiples effets, parfois contraires, qui influencent la méthode d'analyse.

Pour résoudre ce problème, nous avons présenté une heuristique relativement simple, qui permet d'obtenir rapidement de bonnes configurations. Nous avons également présenté une méthode à base d'algorithme génétique permettant d'améliorer cette heuristique, au prix toutefois d'une bien plus grande complexité.

Les résultats obtenus par ces méthodes sont intéressants à deux titres. Tout d'abord, on arrive à choisir de bonnes affectations pour les priorités des VL, qui permettent d'améliorer les résultats de l'analyse du réseau de plus de 30% (par rapport à la configuration tous les VL de priorité basse) : on passe par exemple d'une borne maximale de 769 trames à seulement 525, tout en réduisant la moyenne des bornes sur le délai dans les ports du réseau de 25% environ. D'autre part, on met en évidence les termes du compromis entre ces deux critères, ce qui permet de choisir la bonne configuration en fonction des besoins.

Plusieurs études intéressantes permettraient de compléter notre étude. La première porterait sur une meilleure compréhension du déroulement de l'évolution au cours de l'algorithme génétique. Il serait notamment intéressant de quantifier l'importance relative de chacun des paramètres que nous avons fixés. Une autre étude plus axée sur le réseau pourrait se focaliser sur la caractérisation des bonnes configurations trouvées. Il serait en effet intéressant de

découvrir leurs points communs, ce qui permettrait peut-être à terme de mieux comprendre le fonctionnement global du réseau, et de son analyse par notre outil. On pourrait notamment essayer de quantifier les « corrélations » entre plusieurs VL, c'est-à-dire comprendre quels sont les VL dont les priorités doivent être modifiées en même temps, pour améliorer le plus possible les résultats. Cette connaissance permettrait alors de créer une nouvelle heuristique, ou d'améliorer l'heuristique  $S_{\min}$  proposée.

## 2 Optimisation de l'architecture matérielle du réseau

Nous disposons maintenant d'une méthode d'analyse complète du réseau, qui permet de juger si un réseau avionique respecte les contraintes liées à la qualité de service exigée. Sachant que le réseau sera jugé de cette façon, nous cherchons à l'optimiser vis-à-vis de cette méthode d'analyse. Dans la première partie de ce chapitre, nous avons modifié les caractéristiques du trafic passant dans le réseau, plus précisément les priorités affectées à chaque VL. Nous étudions maintenant des modifications plus profondes du réseau, qui touchent à son architecture physique. Dans un premier temps, nous nous pencherons sur une modification de la manière de relier les commutateurs entre eux, puis nous envisagerons les gains éventuels apportés par une modification de la politique de service des ports de sortie de ces mêmes commutateurs.

### 2.1 Mode d'interconnexion des commutateurs

Les travaux présentés ci-dessous ont été réalisés en collaboration avec des étudiants de l'ENSEEIH, dans le cadre de leur projet long. Ils sont détaillés dans le rapport de projet rédigé à cette occasion [81].

#### 2.1.1 Motivations

Au cours de nos analyses précédentes, nous avons pu constater que les ports qui posent problème, c'est-à-dire au sein desquels la borne calculée est très élevée, sont très peu nombreux : moins d'une dizaine sur plus de 300 ports. Les problèmes sont donc localisés en ces points, ce qui nous a donné l'idée de leur apporter une solution locale.

La solution que nous avons envisagée est relativement simple : puisque les ports de sortie constituent des points d'engorgement, nous avons pensé à doubler ces ports. Dans le cas d'un port qui relie deux commutateurs par exemple, nous avons ajouté un nouveau lien entre deux autres de leurs ports. De manière intuitive, on divise ainsi le trafic qui passe dans les ports, ce qui implique automatiquement une réduction des bornes calculées.

Bien entendu, nous faisons l'hypothèse qu'il y a au moins un port disponible sur chacun des commutateurs impliqués.

#### 2.1.2 Réalisation

La mise en pratique de cette idée est relativement aisée. La principale question qui se pose porte sur le choix des VL à déplacer sur le nouveau lien. Considérant l'expérience acquise précédemment avec le choix des priorités, nous avons décidé de déplacer d'abord les flux de plus petit  $S_{\min}$ . Plus précisément, nous en avons fait deux heuristiques : pour un port donné que nous doublons, la première heuristique déplace un par un les flux, par ordre croissant de  $S_{\min}$ . La deuxième heuristique déplace quant à elle les groupes, toujours selon le même ordre. Les paragraphes suivants présentent les résultats obtenus.

### 2.1.3 Résultats locaux

Pour pouvoir juger de l'efficacité de nos heuristiques, nous avons bien sûr observé la somme des bornes calculées pour les deux ports, celui d'origine et le dupliqué.

Les deux graphiques suivants présentent l'évolution de la somme des deux bornes, au cours des deux heuristiques présentées. Le port que nous avons choisi de dupliquer est l'un des plus chargé du réseau: nous avons calculé pour ce port une borne sur la taille maximale de 730 trames.

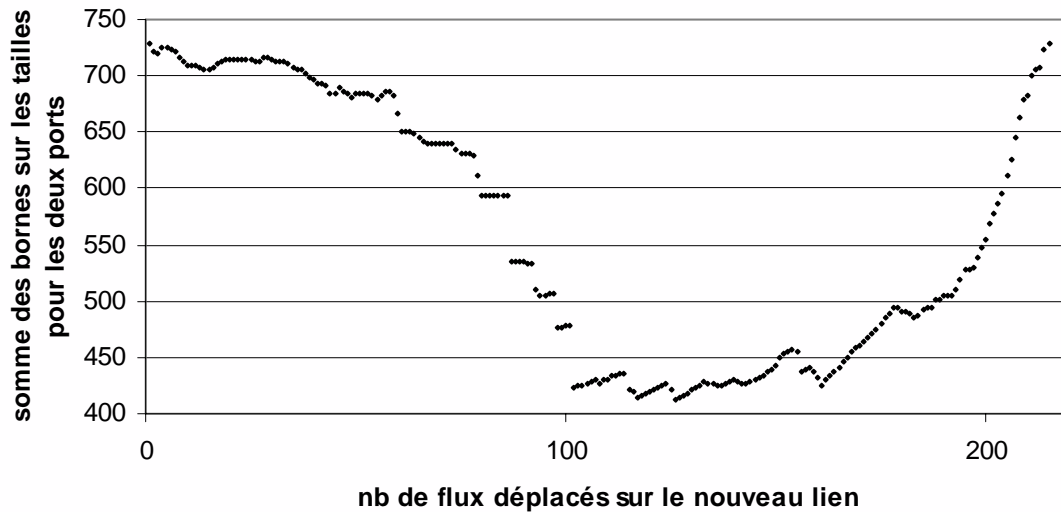


Figure 52 Heuristique Smin en utilisant les flux passant par le port doublé

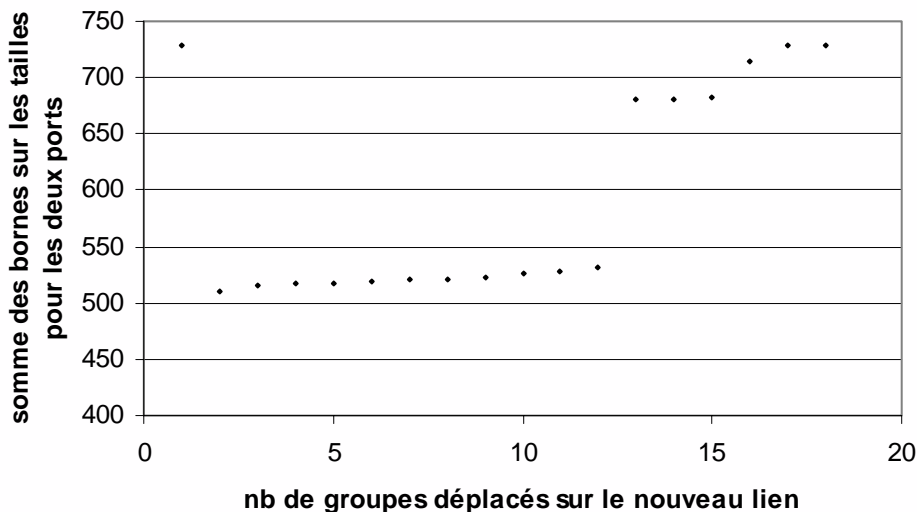


Figure 53 Heuristique Smin utilisant les groupes passant par le port doublé

Ces deux graphiques appellent un certain nombre de commentaires. Tout d'abord, l'évolution au cours de l'heuristique est très marquée, puisque la différence entre l'instant de départ et la valeur atteinte au minimum peut atteindre 40%. On constate bien sûr que les valeurs initiales et finales sont égales, puisque elles correspondent aux cas où tous les flux sont dans un port ou dans l'autre. Il est très intéressant de noter que la somme des deux bornes a tendance à diminuer lorsqu'on répartit le trafic dans les deux ports. Nous interprétons ceci en pensant que lorsque le trafic est moindre dans un port, les congestions sont largement moins importantes :

la somme des valeurs atteintes lors de deux petites congestions est nettement inférieure à celle atteinte lors d'une grosse congestion.

En ce qui concerne la comparaison entre les deux heuristiques, on peut surtout constater que la première permet de diminuer de manière plus importante la somme des bornes que la seconde, de près de 15%. On peut penser que ceci provient du fait que la première heuristique permet de choisir plus librement les flux à déplacer.

On voit donc grâce à ces résultats qu'on peut largement diminuer la taille totale des files d'attente nécessaire dans un commutateur, en répartissant le trafic entre plusieurs ports.

#### 2.1.4 Influence sur le reste du réseau

Nous avons vu dans le paragraphe précédent que localement, la solution proposée présente bien des avantages. Nous avons cependant voulu étudier l'impact de ces modifications locales sur le reste du réseau. Pour ce faire, nous avons regardé la somme des bornes sur tous les ports de sortie du commutateur qui reçoit le trafic par le lien doublé.

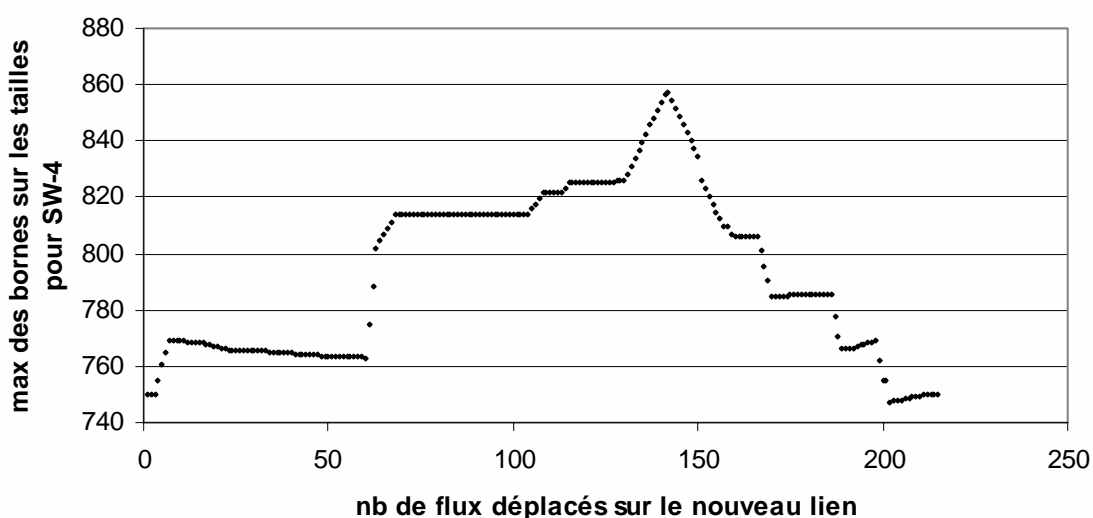


Figure 54 Maximum des bornes sur SW-4 au cours de l'heuristique déplaçant les flux

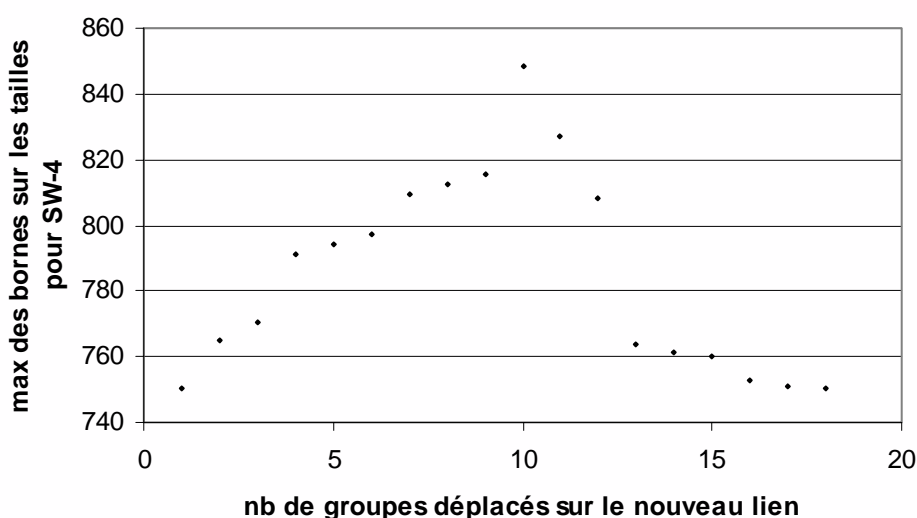


Figure 55 Maximum des bornes sur SW-4 au cours de l'heuristique déplaçant les groupes



Il convient ici de différencier l'analyse suivant l'heuristique considérée. Pour l'heuristique qui déplace les flux, on voit clairement que l'impact sur le commutateur suivant est néfaste, puisque la somme des bornes augmente de près de 150 trames. Ceci s'explique très simplement par le fait que cette heuristique casse des groupes, ce qui limite donc les gains apportés par le groupage des flux.

Pour l'heuristique qui déplace des groupes entiers, ce phénomène n'existe pas. Contrairement à l'autre heuristique, elle permet de conserver les groupes, et de bénéficier de la réduction du délai dans le premier commutateur. En effet, si les groupes subissent moins de congestions dans le commutateur précédent, leur régularité sera moins affectée, et donc leur enveloppe plus basse, ce qui permet d'expliquer les gains constatés dans ce cas, environ 5%.

### 2.1.5 Synthèse

Les résultats de nos deux heuristiques nous permettent de dire que doubler certains liens permet de diminuer la somme des bornes calculées dans un commutateur. On a là une solution locale à des problèmes d'engorgement très localisés. Par contre, ces modifications ne sont pas sans conséquences sur le reste du réseau, et peuvent conduire à en dégrader le fonctionnement en réduisant l'effet de groupage. Nous pensons donc que c'est une solution à utiliser avec précaution, en ayant pleinement conscience de l'impact global de ce genre de modifications.

## **2.2 Politique de service des ports de sortie des commutateurs**

Les travaux présentés ci-dessous ont également été réalisés en collaboration avec un groupe d'étudiants de l'ENSEEIH, dans le cadre de leur projet long de troisième année. Ils sont détaillés dans le rapport de projet rédigé à cette occasion [82].

### 2.2.1 Motivations

Dans les chapitres précédents, nous avons montré que les congestions proviennent des rafales de trafic, qui s'amplifient au fur et à mesure de la progression des flux dans le réseau. La cause de cette amplification est le fonctionnement des ports de sortie des commutateurs, qui implémentent des politiques de service conservatives. Dans celles-ci, les trames sont émises dès que possible, et donc si elles arrivent en rafale, elles seront également émises en rafale. La longueur de celle-ci sera augmentée des nouvelles trames qui se seront accumulées durant le temps d'émission de la rafale.

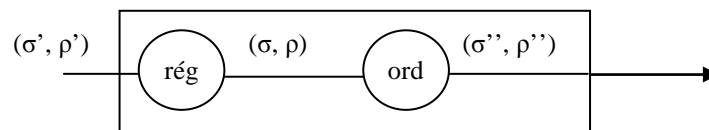
Ce type de politique de service a été choisi pour deux raisons principales. D'une part, les politiques de service conservatives sont souvent d'implémentation plus simple que les politiques non conservatives, principalement parce qu'il n'y a pas besoin de calculer à quelle date retransmettre les trames reçues, ce qui implique l'utilisation d'horloges, d'ordonnanceurs, etc. D'autre part, le fait de retransmettre au plus tôt les trames semble intuitivement être la meilleure solution pour limiter le délai moyen subi par les trames. En effet, le principe des politiques non conservatives est de retarder l'émission de certaines trames, pour améliorer la régularité de l'émission des flux. De plus amples détails sur les différentes politiques de service peuvent être trouvés dans le rapport technique [80], dont une partie est consacrée à un état de l'art sur le sujet.

Dans le cadre de notre étude, nous avons jugé intéressant d'étudier l'apport d'une politique de service non-conservative. Une telle politique permettrait d'améliorer la régularité des flux, donc de diminuer les enveloppes des flux, et donc les bornes calculées. De plus, notre étude s'intéresse aux valeurs maximales du délai, et pas au délai moyen ; il se pourrait alors que nos résultats ne soient pas trop dégradés par ce changement de politique.

D'un point de vue pratique, nous avons choisi comme politique de service non conservative un régulateur de trafic, utilisant le paradigme du seau percé, suivi d'un ordonnanceur de type SP (Static Priority). On peut observer que la politique ainsi obtenue n'est autre qu'un exemple

particulier de politique RCSP, modèle général qui a été étudié par Zhang dans [75]-[78]. De plus, nous soulignons que ce type de politique de service est facilement analysable par le Network Calculus : le régulateur proposé peut se modéliser grâce à un élément appelé « shaper »[79]. Nous rappelons que pour ce type d'élément, la courbe de service offerte au flux est égale à l'enveloppe induite par le dispositif à seuil percé. Cette courbe de service permet alors d'effectuer l'analyse en suivant la méthode d'analyse proposée dans le chapitre précédent. Cependant, le but de ce paragraphe est de mettre en lumière l'apport éventuel d'une telle politique dans notre contexte. En ce sens, nous nous limitons à l'analyse de flux de priorités identiques et nous n'utiliserons pas la notion de groupe, pour ne pas subir d'interférences dues aux effets de priorités différentes ou de groupage.

Le schéma général de la politique de service retenue, pour un flux donné, sera le suivant :



Il existe plusieurs façons de mettre en place cette politique de service ; elles se différencient principalement par le réglage du régulateur à seuil percé. Nous avons retenu deux approches différentes : dans la première, nous régulons tous les flux mais l'importance de cette régulation est configurable. Dans la deuxième approche, nous avons choisi de ne réguler que les flux dont l'enveloppe a beaucoup augmenté (« beaucoup » dépendant d'un seuil que nous fixons) lors de la traversée du réseau ; la régulation est alors totale, c'est-à-dire qu'on force le flux à avoir la même enveloppe que lorsqu'il a pénétré dans le réseau.

## 2.2.2 Première approche : régulation atténuée

### 2.2.2.1 Principe

Le principe de cette approche est de réguler tous les flux, mais cette régulation dépendra de la valeur d'un coefficient de régulation,  $K$ . Pour  $K=0$ , il n'y a pas de régulation : les enveloppes entrant dans le régulateur sont conservées. Inversement,  $K=1$  correspond à une régulation totale, c'est-à-dire que les enveloppes des flux sont régulées jusqu'à être exactement égales à l'enveloppe qu'avaient ces flux au moment où ils pénétraient dans le réseau.

Plus précisément, en respectant les notations présentées ci-dessus, et avec  $(\sigma_0, \rho_0)$  l'enveloppe initiale du flux considéré :

$$\begin{aligned}\sigma &= K \cdot \sigma_0 + (1 - K) \cdot \sigma' \\ \rho &= \rho' = \rho_0\end{aligned}$$

La courbe de service du régulateur sera donc :

$$\beta(u) = K \cdot \sigma_0 + (1 - K) \cdot \sigma' + \rho_0 \cdot u$$

### 2.2.2.2 Réalisation

Pour étudier cette politique de service, nous avons utilisé dans notre outil de calcul un nouvel élément, de type « shaper », dont la courbe de service vient d'être explicitée. Ce type d'élément est décrit en détail dans les travaux de Le Boudec [79]. Pour pouvoir comparer les résultats de cette méthode avec nos résultats précédents, il convient de faire la somme des bornes obtenues, à la fois sur le régulateur et sur l'ordonnanceur. En effet, la mémoire nécessaire pour le commutateur se décompose en deux parties, celle pour stocker les trames que le régulateur retarde, et celle pour stocker les trames qui attendent d'être émises par

l'ordonnanceur. De même, le délai subi par une trame sera égal à la somme du délai subi dans le régulateur et du délai subi dans l'ordonnanceur.

Cependant, faire cette somme pourrait être injustement néfaste à la politique non conservative. En effet, en ce qui concerne la mémoire nécessaire, on considère en faisant ce calcul que les VL sont régulés dans chaque port de sortie. Or une implémentation judicieuse consisterait à réguler les VL non pas dans chaque port de sortie, mais au niveau des ports d'entrée, avant la phase de commutation, et surtout la duplication des trames. Pour une telle implémentation, la mémoire totale nécessaire dans le commutateur est bien moindre qu'avec une régulation dans tous les ports de sortie. En ce qui concerne les délais, on majore le délai subi par une trame dans un port de sortie par la somme : délai subi dans l'ordonnanceur + maximum des délais subis dans leur régulateur par les trames utilisant ce port de sortie.

Nous donnerons par les suites les résultats obtenus dans le cas d'une régulation « avant la commutation ».

### 2.2.2.3 Résultats

Le graphique suivant présente les résultats de notre étude.

En abscisse de ce graphique figurent les différentes valeurs du coefficient de régulation  $K$ .

La première série de données représente la moyenne du délai de bout en bout subi par les flux (on considère chaque flux unicast).

La seconde série représente le maximum, pour tous les ports du réseau, de la borne sur la taille de files d'attente. Comme indiqué dans le paragraphe ci-dessus, la valeur est obtenue en additionnant la borne calculée dans le port (avec les enveloppes régulées), plus la somme des bornes sur les tailles des régulateurs (un régulateur par VL).

On a ainsi une idée de l'impact global de la politique de service étudiée sur le réseau.

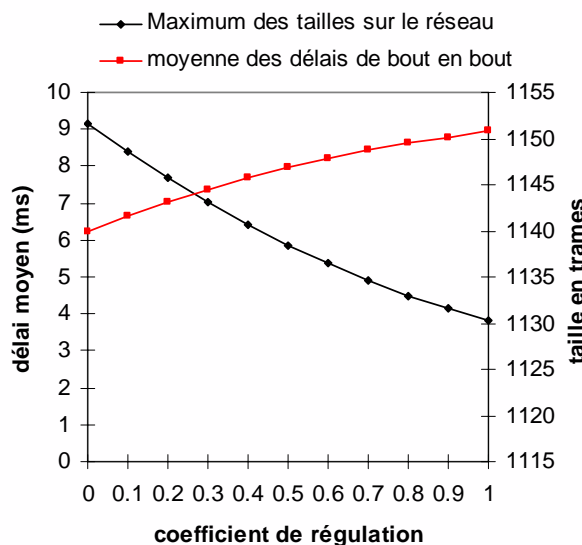


Figure 56 Evolution des bornes en fonction du seuil de régulation

Ce graphique montre clairement que plus la régulation est importante, plus les bornes calculées sur les tailles de files d'attente diminueront, ce qui était bien le but poursuivi par cette étude. Nous comprenons cependant ici le prix de cette amélioration des tailles : les latences calculées augmentent d'autant plus que la régulation est plus importante. On voit notamment que les bornes sur les délais peuvent augmenter en moyenne de 45%, pour un gain sur les tailles de files d'attente de quelques pourcents (environ 2%), obtenu dans le cas d'une régulation totale.

Nous avons voulu vérifier si les résultats ne seraient pas meilleurs dans le cas d'un réseau plus chargé. Pour ce faire, nous avons multiplié le trafic du réseau par un facteur trois, en triplant le débit de chaque VL. Les résultats présentés ci-dessous montrent clairement que la nouvelle politique de service est plus efficace dans le cas d'un réseau plus chargé, car on peut atteindre ici dans le cas d'une régulation totale ( $K=1$ ) avant la commutation des gains proches de 10%, en ce qui concerne les tailles. Cependant, la moyenne des délais augmente encore plus dans ce cas : elle est presque doublée dans le cas d'une régulation totale.

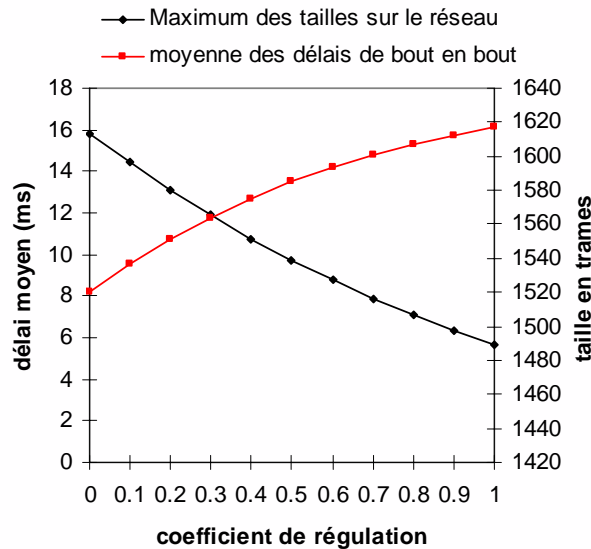


Figure 57 Evolution des bornes en fonction de la régulation, trafic triplé

## 2.2.3 Deuxième approche : régulation adaptative

### 2.2.3.1 Principe

Le principe de cette deuxième approche est de ne réguler que les flux dont l'enveloppe a beaucoup augmenté. Nous espérons de cette façon diminuer le nombre et l'importance des régulations, et peut-être ainsi diminuer la mémoire nécessaire aux régulateurs.

Plus précisément, on caractérise l'augmentation de l'enveloppe d'un flux de la façon suivante, avec  $(\sigma_0, \rho_0)$  l'enveloppe initiale du flux considéré et  $(\sigma', \rho')$  son enveloppe courante:

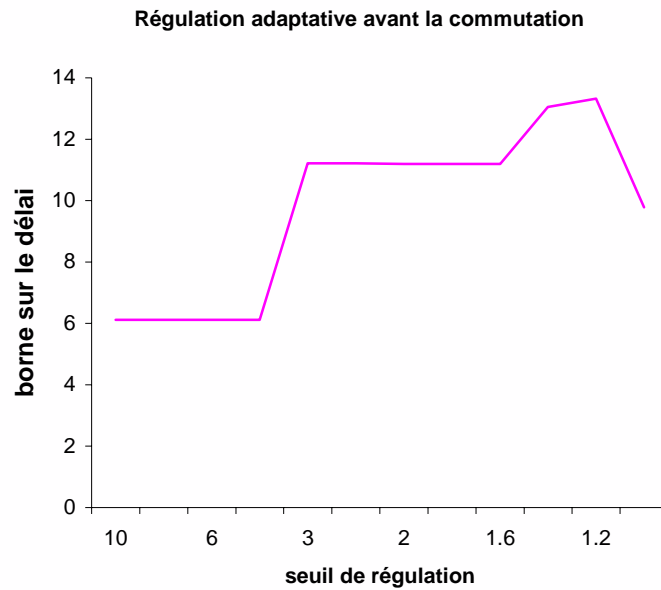
$$\text{augmentation} = \frac{\sigma'}{\sigma_0}$$

On règle le régulateur en fonction d'un seuil  $R$  : si l'augmentation est plus grande que  $R$ , le VL sera régulé totalement (retour à l'enveloppe initiale) ; dans le cas contraire, pas de régulation.

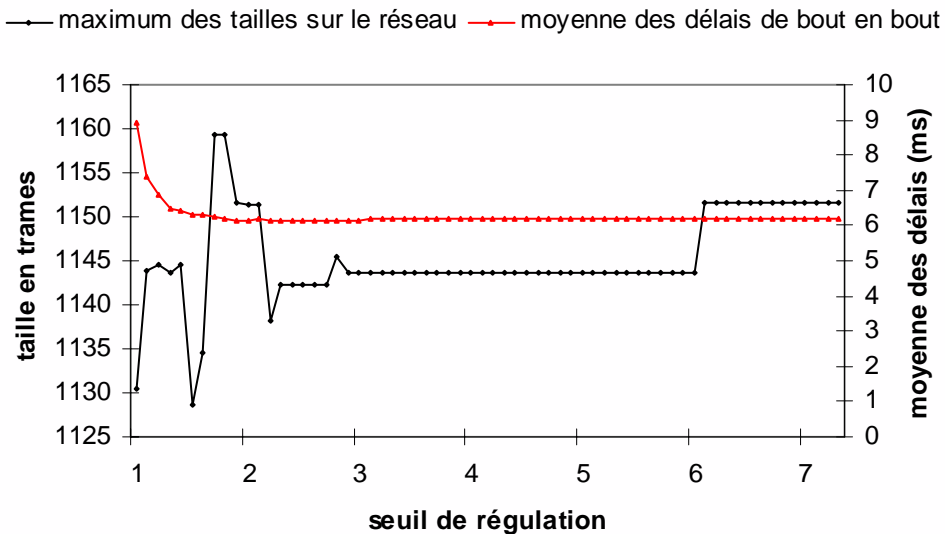
Comme précédemment, nous présentons les résultats obtenus en implantant la régulation avant la commutation.

### 2.2.3.2 Résultats

Pour commencer, nous représentons pour un port donné l'impact de cette politique de service, vis-à-vis de la borne sur le délai. En abscisse figure la valeur du seuil de régulation  $R$  :  $R=10$  correspond à une régulation de très peu de flux, mais dont l'enveloppe aurait beaucoup augmenté.  $R=1$  correspond à la régulation de tous les VL dans tous les commutateurs.



Ce graphique nous montre que pour un port donné, il est difficile de prédire l'évolution de la borne sur le délai en fonction de la valeur du seuil. On peut expliquer cette irrégularité par le fait que suivant le seuil, le flux peut avoir ou non été régulé dans un port précédent. En ce qui concerne l'impact global sur le réseau, le graphique suivant présente les résultats obtenus:



**Figure 58 Evolution des bornes en fonction du seuil de régulation**

Puisque le cas où le seuil est égal à 1 correspond à celui où le coefficient de régulation  $K$  de l'approche précédente est égal à 1, et que le cas où le seuil est supérieur à 10 est similaire à celui où le coefficient de régulation est égal à 0, il est logique de retrouver les mêmes variations entre les extrémités des courbes. Ainsi, on retrouve la baisse de 2 % pour la taille maximale de files d'attente dans le cas de la régulation avant commutation. Cependant pour des valeurs intermédiaires, on constate que l'évolution n'a pas la même allure. Pour cette approche, les valeurs sont peu modifiées tant que le seuil est supérieur à 3. Ceci s'explique par le fait que, dans notre réseau, la très grande majorité des flux subit (de bout en bout) une augmentation inférieure à 3, comme l'indique le graphique ci-dessous :

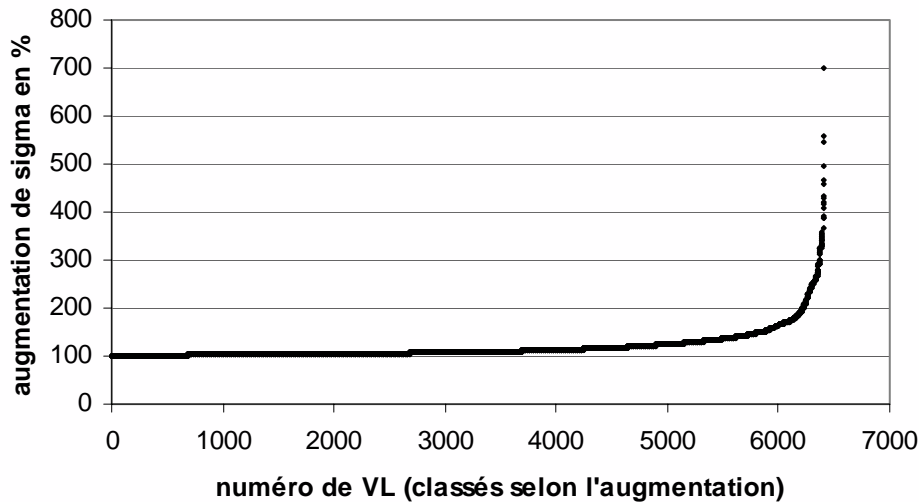


Figure 59 Augmentation de l'ordonnée à l'origine des enveloppes lors de la traversée du réseau

Pour des valeurs plus faibles du seuil de régulation, on voit clairement que l'évolution de la taille maximale est peu régulière, ceci étant dû au phénomène mis en évidence plus tôt, à savoir que les LV peuvent avoir été régulés dans un port précédent le port où est atteinte la valeur maximale.

On peut conclure de ces résultats que pour le réseau que nous avons étudié, cette seconde approche n'est pas particulièrement intéressante, puisqu'elle permet un réglage moins fin qu'avec l'approche précédente. Pour un autre réseau, où l'augmentation des flux est plus répartie, elle pourrait peut-être aboutir à de meilleurs résultats.

#### 2.2.4 Synthèse

Nous avons présenté dans ce paragraphe deux approches dans lesquelles les flux sont partiellement régulés dans les commutateurs. Nous avons souligné que pour le réseau étudié la première approche, consistant à réguler tous les flux dans tous les commutateurs suivant un coefficient de régulation, s'avère plus intéressante. La régulation ainsi effectuée permet de limiter la taille des rafales, et donc de diminuer les bornes sur les tailles calculées jusqu'à 10%. Cependant, ce gain intervient au détriment des bornes sur le délai, qui peuvent presque être doublées, du fait que l'on retarde des trames pour conserver la régularité des flux.

La conclusion de notre étude semble être que le choix de la meilleure régulation dépend du réseau étudié, de sa charge, mais surtout de l'objectif poursuivi par le responsable du réseau : est-il prêt à augmenter le délai de traversée du réseau, pour y diminuer les risques de congestion ? Notre étude permet de mieux comprendre les termes de cet échange.

### 2.3 Synthèse sur l'optimisation de l'architecture matérielle

Dans cette partie, nous avons étudié certaines modifications touchant à l'architecture physique du réseau, toujours pour chercher à créer plus de marge. Nous avons vu que la duplication de certains liens entre commutateurs pouvait diminuer localement les bornes calculées sur les tailles. Cependant, l'impact de cette modification sur le reste du réseau peut être négatif. D'autre part, nous avons montré que l'utilisation de politiques de service non conservatives pouvait permettre de diminuer globalement les bornes sur les tailles, mais au détriment des bornes sur le délai.

Dans les deux cas, il s'agit de méthodes qui permettent d'améliorer des bornes au détriment d'autres. Nous pensons que notre étude permet de mieux choisir s'il faut ou non utiliser ces

méthodes et la manière de les utiliser, en fonction des objectifs poursuivis et en comprenant les répercussions de ces choix.

## Conclusion

L'objectif de ce chapitre était de présenter des méthodes permettant de dégager de la marge pour une éventuelle augmentation du trafic dans le réseau. Nous avons distingué deux domaines où apporter des modifications : le trafic, et l'architecture physique du réseau. En ce qui concerne le trafic, nous avons montré que choisir convenablement les priorités des flux permettait des gains importants, en diminuant par exemple les bornes sur les tailles calculées de 30%. Nous avons présenté, pour aider l'utilisateur dans sa démarche, une heuristique simple permettant un bon choix pour ces priorités, ainsi qu'un algorithme plus complexe qui permet d'obtenir des bornes légèrement meilleures.

Parmi les modifications relatives aux éléments matériels du réseau, nous avons montré qu'on pouvait localement gagner un peu de marge en reliant les commutateurs par plus d'un lien physique. Nous avons cependant souligné les conséquences éventuellement néfastes d'une telle modification sur le reste du réseau (report de congestions). D'autre part, nous avons montré que l'utilisation dans les commutateurs de fonctions de régulation (« reshaping ») pouvait améliorer les bornes sur les tailles, au détriment cependant des bornes sur les latences. Au final, nous avons présenté dans ce chapitre plusieurs méthodes permettant d'améliorer sensiblement les bornes, en essayant d'indiquer les éventuelles conséquences de ces modifications, pour que leur utilisateur puisse le faire en toute connaissance de cause.

Bien sûr, les méthodes présentées pourraient faire l'objet d'études ultérieures pour encore améliorer ces résultats, mais nous pensons qu'une plus grande source d'amélioration se trouve dans l'adoption d'une démarche de certification différente, que nous présentons dans le chapitre suivant.





## CHAPITRE 5

### Une approche combinée déterministe stochastique

#### 1 Motivations

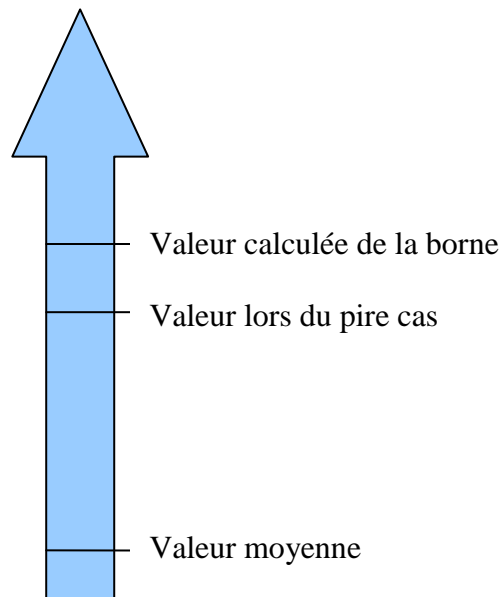
Les chapitres précédents ont montré que les résultats du Network Calculus peuvent s'appliquer au cas d'un réseau avionique, et que leur application est réellement intéressante. Ils s'avèrent même particulièrement bien adaptés pour deux raisons. D'une part la notion d'enveloppe, qui est au centre de la théorie, modélise de façon très exacte les contraintes qui sont imposées aux flux en bordure de réseau lorsqu'ils sont mis en forme pour correspondre aux spécifications du VL. D'autre part, le calcul réseau fournit des valeurs sur les tailles de files d'attente ou les valeurs des délais dont on est sûr qu'elles bornent bien les valeurs réelles. Or ceci correspond exactement à ce qui est exigé par le processus de certification du système avionique retenu en accord avec les autorités de certification.

Cependant, cette dernière exigence est directement reliée au fait que la technologie Ethernet commuté est d'utilisation récente dans le domaine, et qu'on cherche par conséquent à la mettre en pratique en diminuant au maximum les risques. Il est envisageable dans un programme futur, si la technologie est jugée assez mûre, de relâcher cette contrainte de déterminisme, et d'effectuer un autre choix de méthode de certification du réseau. Les prochaines parties visent à mieux comprendre quelles sont les limitations introduites par l'exigence forte du déterminisme, puis présenter des approches différentes qui permettraient de tirer parti d'une modification de cette contrainte.

##### **1.1 Le caractère pessimiste des bornes déterministes**

Le véritable objectif du calcul réseau est de fournir des bornes, c'est-à-dire des valeurs qui ne seront jamais dépassées, même dans le pire cas envisageable. Pour ce faire, et puisqu'il est extrêmement difficile de déterminer la configuration qui pourra aboutir à un cas pire, la méthode impose de considérer à chaque étape du calcul des hypothèses qui sont peut-être excessivement pessimistes.

En pratique, on s'attend à se retrouver dans le cas de figure suivant :



**Figure 60** Borne calculée et valeurs réelles

Le schéma ci-dessus peut illustrer par exemple la situation pour la taille d'une file d'attente particulière, dans un port d'un commutateur donné. En moyenne, il est très probable que la taille de cette file soit beaucoup plus faible que dans le cas pire, puisque ce cas pire ne peut arriver que dans des situations très précises, correspondant par exemple à des arrivées simultanées de trames en provenance de plusieurs sources. D'autre part, la borne calculée doit majorer la valeur réelle de la taille de la file d'attente, même dans le cas pire. Il est raisonnable de penser que la borne calculée est strictement supérieure à la valeur réelle, puisque de nombreuses approximations ont été faites au cours de la démarche de calcul, et ce toujours de manière conservatrice. On peut classer les origines de ces approximations en plusieurs catégories, suivant leur origine :

- Modélisation du réseau : comme toujours lors d'un calcul, l'analyse s'effectue sur un modèle du réseau, qui par définition ne peut pas prétendre être une représentation totalement exacte de la réalité. Il faut donc lors de l'étape de modélisation ne retenir que des hypothèses conservatrices, pour s'assurer que le modèle n'engendrera pas de résultats trop optimistes. La modélisation du commutateur qui a été retenue dans cette étude en est un bon exemple : les différents processus mis en œuvre dans le commutateur ont été modélisés au moyen d'un élément de type délai borné. Cette modélisation est sûre, puisque elle fait partie des spécifications du commutateur, mais elle est certainement pessimiste par rapport au fonctionnement réel du commutateur.
- Modélisation du trafic : le trafic est modélisé conformément à la notion de VL ; or celle-ci est intrinsèquement pessimiste, car elle ne décrit que le trafic maximum qui peut être émis par un abonné ; celui-ci peut en réalité en émettre moins. Il faut également avoir conscience qu'il n'est pas toujours évident de dimensionner correctement les VL, faute de connaître assez précisément le fonctionnement des applications embarquées. Les VL définis réservent donc certainement trop de ressources dans le réseau par rapport aux besoins réels de communication.
- Méthode de calcul elle-même : il a été démontré que les bornes données par les théorèmes du Network Calculus sont serrées, c'est-à-dire qu'il est possible de créer un système qui atteigne ces bornes. Cependant, l'application des théorèmes au cas particulier d'un réseau avionique peut engendrer des majorations supplémentaires. Un tel exemple de majoration peut être trouvé, dans la présente étude, lors des conversions effectuées entre le domaine des bits et le domaine des trames. Pour

s'assurer de la validité du calcul, les conversions doivent prendre pour hypothèse que toutes les trames ont la pire taille possible, minimale ou maximale suivant le cas.

L'ensemble de ces hypothèses conservatives explique pourquoi notre méthode d'analyse ne permet pas de réellement étudier le fonctionnement du réseau, mais seulement de borner certaines valeurs.

D'après l'analyse précédente de la situation, un premier axe de recherche consiste à rapprocher la valeur calculée de la valeur réelle mesurable lors d'un cas pire. Pour ce faire, il convient de diminuer les majorations effectuées au cours de chaque étape de l'analyse du réseau. On peut notamment améliorer le modèle du réseau, par exemple en améliorant la modélisation du commutateur que nous avons adoptée. Ceci pose cependant la question de l'indépendance de l'analyse par rapport aux équipements matériels mis en place. Le modèle existant a en effet l'avantage de ne seulement prendre pour hypothèse que le commutateur utilisé est conforme aux spécifications. Pour affiner la modélisation actuelle, il faudrait développer un modèle qui se rapproche plus de l'implémentation choisie par le fabricant de matériel. Le modèle serait certes plus représentatif de la réalité, mais ne serait plus valable dès qu'un autre commutateur (pourtant conforme aux spécifications) serait utilisé. On pourrait également penser à s'assurer que les VL sont bien déclarés conformément aux besoins réels des abonnés, mais cette démarche est d'une part difficile à accomplir d'un point de vue technique (difficulté d'évaluer les besoins réels actuels ou à court terme), et elle relève également d'un caractère industriel lié à la problématique de l'organisation des différents services de l'avionneur. Enfin, la source d'amélioration qui apparaît potentiellement la plus profitable est de modifier la méthode de calcul proprement dite. Cela a été en partie réalisé dans le chapitre 3 de cette étude, avec notamment l'utilisation de groupes et la réflexion menée sur les conversions entre les domaines des bits et des trames, mais d'autres améliorations sont toujours envisageables. Toutes ces améliorations permettraient certainement de diminuer la valeur calculée, mais afin de connaître les marges de progression existantes, il est fondamental de mieux connaître la valeur réelle mesurable dans le cas pire. Pour ce faire, on peut envisager des campagnes de test sur le réseau réel, ou bien avoir recours à des simulations de ce réseau. Les deux démarches possèdent le point commun de devoir mesurer des phénomènes qui peuvent être extrêmement rares, ce qui les rend toutes deux difficiles à accomplir de manière rigoureuse.

## **1.2 Apport des méthodes stochastiques**

Le second axe de recherche envisagé consiste à se rapprocher non pas de la valeur dans le cas pire précédente, mais bien de la valeur « probable ». On entend par valeur probable une valeur plus faible que la valeur dans le cas pire, mais suffisamment élevée pour qu'elle ne soit dépassée qu'en de très rares occasions. Cette notion de valeur probable fait bien entendu surgir la notion de probabilité d'occurrence des cas pires, dont on peut avoir l'intuition qu'elle est très faible. En effet, même pour un élément isolé, les scénarios qui conduisent au cas pire supposent qu'un ensemble de conditions très précises soit réuni, ce qui est peu probable dans un système réel, et encore moins probable lorsqu'on considère un réseau entier. Cependant, cette probabilité est très difficile à évaluer de manière directe, tant les scénarios aboutissant au cas pire sont difficiles à déterminer. On pourrait utiliser les méthodes statistiques classiques pour étudier ce problème, et découvrir ces probabilités. En effet, ces méthodes possèdent l'avantage d'être bien connues, et d'avoir été fréquemment utilisées. Le modèle utilisé pour le réseau peut facilement être converti en un modèle par files d'attente, qui est bien adapté à ce genre de démarches. Malheureusement, il manque dans ce cas particulier un élément fondamental pour utiliser une approche statistique classique, qui est la caractérisation des entrées. En effet, ce type d'approche nécessite une connaissance du comportement statistique

des entrées, notamment par le biais de la définition de leur loi. On peut par exemple trouver de nombreux résultats si on suppose un modèle d'entrée de type Markovien modulé fluide [105][106] ou « fractional brownian [107] ». Le livre [108] recense un grand nombre de ces études. Dans le cas d'étude considéré, ce genre de connaissance fine du comportement des flux n'est pas disponible. Les seules données connues sont fournies par la notion de VL, qui caractérise l'émission maximale des flux d'une certaine manière, et qui ne contraint pas le comportement statistique de ces flux : par exemple, la donnée d'un temps inter émission minimum ne spécifie en rien le temps moyen inter émission.

Une autre approche envisageable pour étudier le problème est d'avoir recours à des simulations du réseau. Nous avons cependant jugé la tâche hors de la portée de cette thèse, pour plusieurs raisons. Tout d'abord, la complexité de telles simulations est très importante, à cause de la taille du réseau, mais surtout à cause de la granularité très fine qu'il faut employer pour pouvoir suivre des événements très rares. D'autre part, il nous a semblé que des résultats de simulations pouvaient difficilement servir de base à une nouvelle méthode de certification, qui pourrait éventuellement être mise en place pour des programmes futurs. Enfin, cette approche nécessite également de faire certaines hypothèses sur le comportement statistique des flux en entrée.

A partir de ces constatations est née l'idée d'utiliser une approche combinée stochastique déterministe, qui offre l'important avantage de conserver la notion d'enveloppe déterministe, particulièrement adaptée à notre cas d'étude, et celui d'offrir des garanties de type statistique.

## **2 Principe d'une approche combinée déterministe stochastique**

### **2.1 Etat de l'art sur les garanties statistiques**

Un certain nombre de travaux ont porté sur l'établissement de résultats de type stochastique. La principale difficulté rencontrée est de trouver un modèle qui soit valable à l'intérieur du réseau, et pas seulement pour les éléments périphériques. On peut par exemple distinguer les approches de type « mieux que Poisson », introduites par Bonald [96]. Dans cette approche, on suppose que les flux en entrée du réseau sont indépendants et périodiques. La méthode proposée repose sur la conjecture, vérifiée par simulation mais non démontrée, que les tailles des files d'attente d'un nœud à priorité statique sont majorées par celles du même nœud, mais qui aurait comme entrées un flux poissonien de paquets de taille constante. En utilisant cette conjecture, les auteurs démontrent que chaque nœud peut être modélisé par une file M/D/1. Ce faisant, ils établissent la répartition du délai de bout en bout.

D'autres approches prennent seulement comme hypothèse que les différentes sources sont indépendantes, mises en forme avant leur entrée dans le réseau, et stationnaires.

#### **2.1.1 Hypothèse des sources on-off**

Les travaux dans cette catégorie [91][92][95] ont en commun d'utiliser l'heuristique suivante : on considère que le cas pire pour un multiplexeur est atteint lorsque les flux qu'il reçoit possèdent un trafic de type « on-off ». Ceci signifie que les sources vont périodiquement émettre à leur débit maximum durant une certaine période, puis ne plus rien émettre. Cette hypothèse permet d'obtenir de nombreux résultats, bien qu'on ne puisse pas garantir qu'elle soit rigoureusement exacte dans de nombreux cas, comme le souligne Rajagopal, Reisslein et Ross dans [94].

#### **2.1.2 Inégalités de Hoeffding**

Dans [89][90], les auteurs étudient le problème d'un multiplexeur à taux constant, dont les arrivées (identiques) sont contraintes par deux régulateurs successifs de type seuil percé. Cette

contrainte signifie simplement que les sources ont un comportement crête et un comportement moyen connus. A partir de ces hypothèses, en utilisant un argument de type inversion de Palm, ils expriment une borne sur la répartition de la taille de la file d'attente.

Dans [83], les auteurs étudient le même problème, et parviennent à généraliser les résultats précédents au cas de flux caractérisés par des enveloppes, au sens du calcul réseau déterministe.

Ces deux travaux ont en commun d'avoir été généralisés par Vojnović et Le Boudec dans [84][85] ; il a même été démontré que les deux méthodes développées pouvaient être considérées comme des cas particuliers d'application de bornes de Hoeffding [88], comme nous le verrons en détail dans la partie 2.2.

### 2.1.3 Enveloppes à $\epsilon$ près

Une autre approche a été présentée par Boorstyn, Liebeherr et Burchard dans [97]. Cette démarche, basée sur les bornes de Chernoff et le théorème central-limite, définit le concept d'enveloppes « à  $\epsilon$  près ». Ces enveloppes ne sont plus déterministes, mais telles que la probabilité d'être dépassées est très faible. De façon duale, les auteurs définissent dans [98] la notion de courbe de service « à  $\epsilon$  près ». Munis de ces deux éléments, ils démontrent à nouveau des théorèmes similaires à ceux du calcul réseau déterministe. L'inconvénient majeur de cette méthode est qu'elle est difficile d'application. En effet, les résultats ne sont pas fournis sous forme d'expressions explicites, et nos tentatives d'application se sont trouvées confrontées à des problèmes de type résolution d'extremums.

On peut également trouver dans la littérature [109][110] [111] des travaux fondés sur la méthode dite « exponential bounded burstiness ». Selon Le Boudec et Fioriu [86] ceux-ci peuvent être considérés comme des cas particuliers d'enveloppes « à  $\epsilon$  près ».

## 2.2 Présentation des travaux de Vojnović

### 2.2.1 Contexte

Les travaux présentés dans [84][85] par Vojnović et Le Boudec ont pour cadre les réseaux de type IETF Differentiated Services (DiffServ) [102]. Dans ce type d'architecture, on offre aux flux plusieurs types de qualité de service, et le PHB (Per Hop Behaviour) pour chaque service spécifie la façon dont chaque routeur va traiter les paquets appartenant à des flux ayant demandé cette qualité de service. Une grande différence de DiffServ par rapport à IntServ est que les routeurs ne traitent pas chacun des flux individuellement, mais plutôt les flux agrégés formés de tous les flux individuels appartenant à la même classe de service.

Vojnović et Le Boudec s'intéressent particulièrement au PHB Expedited Forwarding (EF) [99][100]. La définition de EF repose sur un modèle abstrait de nœud, inspiré de GPS et appelé Packet Scale Rate Guarantee (PSRG). Un nœud offre cette garantie au flux agrégé EF avec un taux  $r$  et une latence  $e$  si et seulement si, la date de départ  $d_n$  du  $n^{\text{ième}}$  paquet, compté selon l'ordre d'arrivée, satisfait :

$$d_n \leq f_n + e$$

où  $f_n$  est défini par récurrence, avec  $f_0=0$  et

$$f_n = \max \left\{ a_n, \min \{ d_{n-1}, f_{n-1} \} + \frac{l_n}{c} \right\}, n \geq 1$$

avec  $a_n$  et  $l_n$  respectivement la date d'arrivée et la taille du  $n^{\text{ième}}$  paquet.

Le but de PSRG est de définir formellement l'idée suivante : le nœud garantit au flux agrégé au moins le taux de sortie  $r$ , avec une tolérance  $e$ . Un cas particulier de PSRG est un ordonnanceur qui accorde une priorité haute au flux EF par rapport aux autres flux, mais en

restant non préemptif. Dans ce cas, le taux  $r$  est la capacité de sortie de l'ordonnanceur, et la tolérance correspond au temps d'émettre un paquet non EF de taille maximale.

Ce modèle abstrait PSRG, outre le fait qu'il permet de formaliser proprement l'idée précédente, permet aussi d'obtenir facilement des bornes sur les délais de bout en bout dans un réseau. Il permet également d'obtenir des bornes sur le délai à partir de bornes sur les tailles de files d'attente, ce qui serait impossible avec une caractérisation du service d'un élément avec la notion de courbe de service [104].

Le point qui permet d'effectuer le rapprochement avec notre contexte d'étude est que les flux EF, tout comme les flux appartenant à des VL, sont mis en forme avant leur entrée dans le réseau conformément à un contrat de trafic. A ceci s'ajoute le fait que ces travaux utilisent le même formalisme que celui adopté dans la méthode de calcul réseau déterministe, y compris la notion de courbe de service. On verra enfin que certains de ces résultats nécessitent une analyse déterministe préalable du réseau.

### 2.2.2 Principales hypothèses

Les hypothèses sur les flux sont les suivantes:

(A1) A l'entrée dans le réseau, les flux sont statistiquement indépendants. Plus précisément, pour toute amplitude d'intervalle  $\Delta t$ , les variables aléatoires définies comme étant l'accroissement du premier flux de chaque VL durant  $\Delta t$  seront indépendantes. Cette hypothèse a été retenue dans d'autres travaux de recherche, notamment ceux de Bonald, Proutière et Roberts[96]. On note  $R_i(t)$  l'accroissement du flux  $i$  entre 0 et  $t$ , et  $R(t)$  la somme de tous les  $R_i(t)$ . Plus généralement, l'indice  $i$  désigne un flux particulier, l'absence d'indice désignant le flux agrégé.

(A2) A l'entrée dans le réseau, les flux traversent un régulateur. Pour le flux  $i$ , on note  $\alpha_i$  l'enveloppe induite.

(A3) On a pour chaque flux:

$$E(R_i(t) - R_i(s)) \leq \rho_i(t - s), \forall s \leq t$$

où:  $\rho_i = \lim_{t \rightarrow +\infty} \frac{\alpha_i(t)}{t}$ . On peut remarquer que ceci est vrai en particulier pour des flux ergodiques ayant des accroissements stationnaires. On notera  $\rho$  la somme de ces  $\rho_i$ .

(A4) Chaque nœud offre au flux agrégé une courbe de service  $\beta$ . Certains résultats sont démontrés dans le cas particulier d'une courbe de service super additive, c'est-à-dire telle que  $\beta(t + s) \geq \beta(t) + \beta(s)$ ,  $\forall t, s \geq 0$ .

Dans le paragraphe suivant, on retrouvera également les notations:

- $R_i^*(t)$  le trafic cumulé du flux  $i$  en sortie de l'élément considéré.
- $Q(t) = R(t) - R^*(t)$  la quantité de bits présents dans le nœud à l'instant  $t$ .
- $\tau$  la borne sur la durée d'une période d'activité, définie par:

$$\tau = \inf \{u \geq 0 / \alpha(u) \leq \beta(u)\}$$

On remarque que  $\tau$  est de manière immédiate une borne sur le délai au sein de l'élément. Plus précisément, il s'agit de la borne établie par Cruz pour un élément de type multiplexeur générique conservatif.

- $\lambda_c(t)$  la droite d'équation  $\lambda_c(t) = c.t$ .
- $v(f, g) = \sup_{u \geq 0} \{f(u) - g(u)\}$  la distance verticale entre les courbes  $f$  et  $g$ .

- De même,  $h(f, g)$  est la distance horizontale entre  $f$  et  $g$ .

### 2.2.3 Méthode

La méthode proposée par Vojnović permet de fournir des résultats de type stochastique sur les tailles de files d'attente dans un élément particulier, sous les hypothèses présentées précédemment. Le schéma général permettant d'aboutir à ces bornes est le suivant :

1. majorer la taille de la file par une somme de processus indépendants,
2. utiliser les résultats du calcul réseau déterministe pour majorer ces processus,
3. appliquer les bornes de Hoeffding [88] pour obtenir le résultat recherché.

Ces dernières sont un outil important fréquemment utilisé dans la conception d'algorithmes aléatoires, ou dans le cadre de méthodes probabilistes [101]. Elles portent sur les sommes de variables aléatoires indépendantes et bornées, et constituent une généralisation des bornes de Chernoff.

Plus précisément, soient  $X_1, X_2, \dots, X_N$  des variables aléatoires indépendantes, on note  $X = \sum_i X_i$ ,  $\mu = E[X]$ ,  $a = (1 + \delta) \cdot \mu$ ,  $\delta \geq 0$ . On cherche à majorer la probabilité

$P(X \geq a)$ . L'idée présente derrière ces résultats est de considérer que, selon l'inégalité de Markov,

$$P(X \geq a) = P(e^{tX} \geq e^{at}) \leq \frac{E[e^{tX}]}{e^{at}}$$

Ensuite, il faut calculer une borne  $u(t)$  sur  $E[e^{tX}]$ , puis minimiser  $\frac{u(t)}{e^{at}}$  pour  $t > 0$  pour obtenir la borne sur la probabilité  $P(X \geq a)$ .

La première des inégalités de Hoeffding peut s'exprimer de la façon suivante avec  $a_i \leq X_i \leq b_i$  :

$$P[X - \mu \geq t] \leq e^{-\frac{2t^2}{\sum_i (b_i - a_i)^2}}$$

### 2.2.4 Principaux résultats

Vojnović a utilisé la méthode décrite précédemment de deux manières différentes. La première permet de généraliser les résultats obtenus par Kesidis et Konstantopoulos [89][90], à des enveloppes et des courbes de service quelconques. La seconde manière permet de généraliser les résultats de Chang, Song et Chiu [83] au cas d'une courbe de service super additive. Dans les deux cas, Vojnović fournit des résultats explicites même dans le cas de variables aléatoires non uniformément bornées.

#### 2.2.4.1 Bornes de Kesidis et Konstantopoulos

On ne présentera dans cette partie que la méthode s'appliquant au cas hétérogène, c'est-à-dire le cas où tous les flux ne sont pas bornés par les mêmes quantités. Le cas homogène se traite résolument de la même façon. L'idée consiste à considérer l'arriéré de travail qui est dû à chacun des flux séparément. Plus formellement, on considère que  $Q(t) \leq \sum_i Q_i(t)$ , où  $Q_i(t)$

représente la partie de l'arriéré de travail qui serait due au flux  $i$ , si celui-ci recevait une fraction  $\gamma_i$  du service total offert au flux agrégé :

$$Q_i(t) = \sup_{s \leq t} \{R_i(t) - R_i(t-s) - \gamma_i \cdot \beta(t-s)\}, \quad \forall \gamma_i / \sum_i \gamma_i = 1.$$

Selon le calcul réseau déterministe,  $0 \leq Q_i(t) \leq v(\alpha_i, \gamma_i \cdot \beta)$ .

Ensuite, en appliquant la borne de Hoeffding, on obtient :

$$P(Q(t) > b) \leq \exp \left( - \frac{2(b - \rho h(\alpha, \beta))^2}{\inf_{\gamma} \sum_i v(\alpha_i, \gamma_i \beta)^2} \right)$$

#### 2.2.4.2 Bornes de Chang, Song et Chiu

Les résultats suivants sont valides pour une courbe de service super additive. Encore une fois, on ne montre ici que l'obtention des résultats dans le cas hétérogène, qui est celui qui se rapproche le plus de notre contexte.

Dans cette approche, on commence par remarquer que :

$$Q(t) \leq \sup_{0 \leq s \leq \tau} \{R(t) - R(t-s) - \beta(s)\}$$

$\tau$  étant une borne déterministe sur la durée d'une période d'activité.

Au lieu de considérer l'arriéré de travail dû à chacun des flux, comme précédemment, on va utiliser dans cette approche l'arriéré de travail qui peut se rajouter dans des intervalles temporels plus petits que  $[0, \tau]$ . Pour appliquer la méthode, on va définir une partition  $s_k$  de cet intervalle, et prendre comme variable aléatoire les  $R_i(t) - R_i(t - s_{k+1})$  qui représentent l'accroissement de  $Q(t)$  sur chacun des  $[s_k, s_{k+1}]$ . Plus précisément, par passage aux probabilités on réécrit la formule précédente en :

$$P(Q(t) > b) \leq \sum_k P\{R(t) - R(t - s_{k+1}) > b + \beta(s_k)\},$$

D'après la définition des enveloppes, les variables aléatoires choisies sont comprises entre 0 et  $\alpha_i(s_{k+1})$ . On peut donc appliquer les bornes de Hoeffding pour obtenir :

$$P(Q(t) > b) \leq \sum_k \exp \left( - \frac{2 \cdot [(b + \beta(s_k) - \rho \cdot s_{k+1})^+]^2}{\sum_i \alpha_i(s_{k+1})^2} \right)$$

De la même façon, on peut considérer comme variables aléatoires les  $Z_i(t) = \tilde{Q}_i(t) - \tilde{Q}_i(0)$ , où  $\tilde{Q}_i(t) = \sup_{s \leq t} \{R_i(t) - R_i(s) - \rho_i(t-s)\}$ . On a de manière immédiate:

$$-v(\alpha_i, \lambda_{\rho_i}) \leq Z_i(t) \leq v(\alpha_i, \lambda_{\rho_i}).$$

D'après une nouvelle inégalité de Hoeffding,

$$P(Q(t) > b) \leq \sum_k \exp \left( - \frac{[(b + \beta(s_k) - \rho \cdot s_{k+1})^+]^2}{2 \cdot \sum_i v(\alpha_i, \lambda_{\rho_i})^2} \right)$$

En prenant le minimum des deux expressions, on trouve finalement:

$$P(Q(t) > b) \leq \sum_k \exp \left( - \frac{2 \cdot [(b + \beta(s_k) - \rho \cdot s_{k+1})^+]^2}{\min \left( \sum_i \alpha_i(s_{k+1})^2, 4 \cdot \sum_i v(\alpha_i, \lambda_{\rho_i})^2 \right)} \right)$$

Il convient de remarquer que cette formule est valable pour toute subdivision  $s_k$ ; la question du choix de la subdivision sera abordée dans la partie 3.

Dans la suite de nos travaux, il a été choisi de ne considérer que la dernière borne, qui est celle qui donne les meilleurs résultats dans la plupart des cas, comme le démontre Vojnović dans [84].



### 2.2.5 Autres résultats

#### 2.2.5.1 Bornes aux instants d'arrivée

Les résultats précédents peuvent être interprétés comme étant la fraction de temps pendant laquelle la taille de la file d'attente est supérieure à un certain niveau. Il peut être intéressant, notamment dans le cas d'un modèle PSRG, de connaître la probabilité que la file d'attente soit supérieure à un certain niveau, **au moment** de l'arrivée d'une trame (probabilité de Palm). En notant  $P_A$  cette probabilité, et  $\hat{\beta}$  la pente maximale de  $\beta$ , Vojnović démontre :

$$P_A(Q(t) > b) \leq \frac{\hat{\beta}}{\rho} P(Q(t) > b)$$

Dans le cas d'un modèle PSRG, et contrairement au cas où la seule connaissance sur l'élément est qu'il offre une certaine courbe de service, il est possible d'obtenir une borne sur le délai à partir de la borne sur l'arriéré de travail. En utilisant la formule précédente, on peut déterminer une borne sur la répartition du délai [85].

#### 2.2.5.2 Extension du calcul à plusieurs éléments

Les résultats démontrés jusqu'à présent sont tous valables pour des éléments dont les entrées sont indépendantes. Ceci peut être valide pour les éléments périphériques du réseau, mais n'a plus de raison d'être pour les éléments centraux. Nous proposons deux méthodes pour effectuer tout de même ce calcul au cœur du réseau.

La première est inspirée du fait que les flux ne sont plus indépendants dès qu'ils ont été multiplexés sur un lien physique commun. Cet effet a déjà été pris en compte dans notre analyse grâce à l'utilisation de groupes ; l'idée est donc d'utiliser ces groupes, et non les flux individuels, dans l'application des résultats ci-dessus.

La seconde méthode est suggérée par Chang, Song et Chiu dans [83], et fait encore une fois appel aux résultats de l'analyse déterministe. On remarque que  $\forall s, t \ R_i(t) - R_i(s) \leq R_i^0(t) - R_i^0(s - \Delta)$ , où  $R^0$  désigne la fonction cumulative décrivant le flux en bordure du réseau, et  $\Delta$  représente une borne sur la gigue maximum subie par le flux, depuis son entrée dans le réseau jusqu'à l'élément considéré. On va ainsi se ramener à mesurer l'accroissement des flux initiaux, qui sont par hypothèse indépendants.

En pratique, cette technique de majoration appelée « fresh traffic majorization » revient à remplacer dans les différentes formules  $s_{k+1}$  par  $s_{k+1} + \Delta$ .

#### 2.2.5.3 Borne sur la fraction de pertes à long terme

Les résultats précédents nous offrent des bornes sur la probabilité instantanée de congestion de la file d'attente. En pratique il peut être intéressant de savoir quelle est la fraction du trafic total qui va être perdue par congestion, à cause de la taille physique de la file d'attente et pendant une longue période de fonctionnement du réseau. En notant  $L(t)$  la quantité de trafic perdue durant  $[0, t]$ , on peut définir la fraction de pertes par  $l(t) = \frac{L(t)}{R(t)}$ , et on s'intéresse à

$\bar{l} = \lim_{t \rightarrow \infty} l(t)$ . D'après Chuang et Chang [103], on connaît une borne déterministe sur cette fraction de perte durant un intervalle de temps de longueur  $t$ , pour une file d'attente de taille  $B$  :  $\hat{l}(t) = 1 - \inf_{s \leq t} \left( \frac{\beta(s) + B}{\alpha(s)} \right)$ . En utilisant ce résultat, Vojnović démontre, avec l'hypothèse supplémentaire d'un temps discret,

$$E[L(t) - L(t-1)] \leq \hat{l}(1)\alpha(1).P(Q(t) > B)$$

Ce qui lui permet de démontrer :

$$\bar{l} \leq \frac{\hat{l}(1)\alpha(1)}{\rho} P(Q(t) > B)$$

Il convient de bien noter que ces résultats sont valides pour un temps discret, et donnent des probabilités de perte de bits. Il serait intéressant dans notre contexte de vérifier la validité de ces résultats en temps continu, et d'en déduire des probabilités de perte de trames.

### 3 Etude des bornes obtenues par Vojnović

#### 3.1 Etude des facteurs influents de ces bornes

Les formules présentées dans la partie précédente ne sont pas de lecture aisée, et ne se prêtent pas à une interprétation physique évidente. Nous avons donc décidé de proposer une première approche de ces résultats de type « boîte noire », c'est-à-dire en faisant varier un à un les différents paramètres intervenant dans ces formules, pour quantifier leur importance relative.

Les résultats suivants ont été obtenus pour un multiplexeur FIFO à capacité de sortie 100Mbps, dont les entrées possèdent des enveloppes affines  $(\sigma_i, \rho_i)$  telles que  $\sum_i \sigma_i = 31744 \text{ bits}$ , et  $\sum_i \rho_i = 36,6.10^6 \text{ bits/s}$ . Dans ces conditions, on peut calculer une

borne déterministe sur le délai,  $D=317.4 \mu\text{s}$ , ainsi que sur la taille maximale de la file d'attente,  $Q=31744 \text{ bits}$ . Le calcul fait aussi intervenir une borne sur la longueur maximale d'une période d'activité,  $\tau=32,9\text{ms}$ . Parmi les différents paramètres, nous avons étudié particulièrement l'influence du nombre d'intervalles dans la subdivision de  $[0, \tau]$ , du nombre de sources, de leur charge et de leur homogénéité.

### 3.1.1 Influence du nombre de pas de subdivision

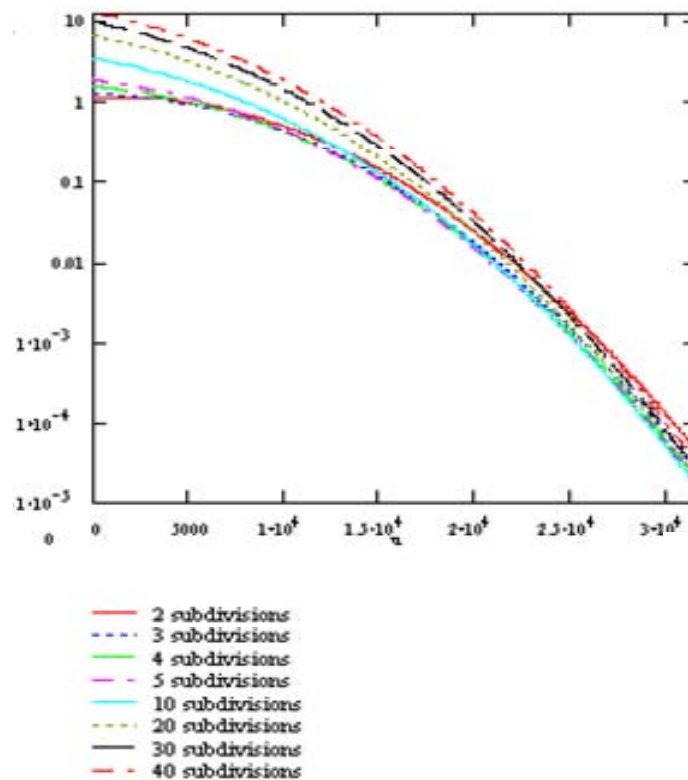


Figure 61 Effet du nombre de subdivisions

La figure ci-dessus, comme toutes celles de cette section, montre en abscisse une taille de file d'attente, et en ordonnée une majoration de la probabilité de dépasser cette valeur. Bien évidemment, plus la taille de file d'attente est grande, plus la probabilité de la dépasser est faible. D'autre part, les graphiques sont représentés en échelle logarithmique, ce qui est plus adapté au vu des formules utilisées.

Les subdivisions choisies ici sont régulières ; d'autres études ont été menées sur des subdivisions irrégulières. Il apparaît cependant qu'il est très difficile de choisir la régularité de la subdivision dans le but d'améliorer les résultats, à nombre de subdivisions constant. La première constatation que l'on peut faire est que la borne dépasse la valeur 1 pour les faibles valeurs de taille de file d'attente. Il est évident que nous aurions pu choisir de ne représenter que  $\min(1, P(Q(t) > b)$ . Néanmoins, cela n'aurait pas apporté de réelle amélioration, puisque les ordres de grandeur intéressants n'apparaissent que pour des  $b$  beaucoup plus grands.

Il apparaît assez clairement ici que le nombre de pas de subdivision choisi n'est pas un facteur de premier ordre. La différence entre les courbes est relativement faible, on peut donc penser utiliser ce paramètre uniquement dans le but d'optimiser des résultats. Par la suite, nos études utiliseront un faible nombre de subdivisions (3 ou 5) pour diminuer le nombre de calculs.

### 3.1.2 Influence du nombre de sources

Le graphique ci-dessous montre l'évolution des bornes en fonction du nombre de sources qui entrent dans l'élément ; nous avons choisi ici de rester à charge totale constante quel que soit le nombre de flux.

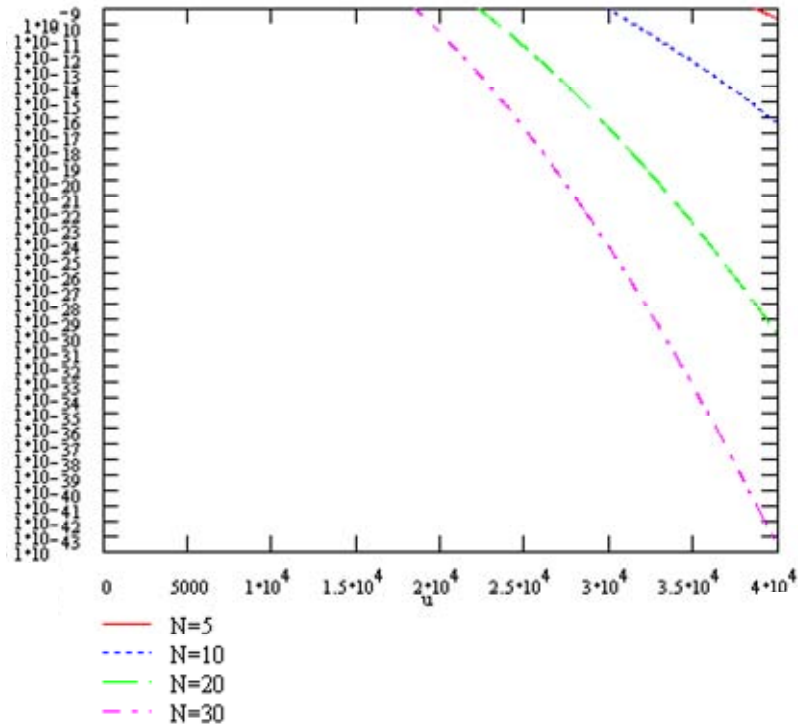


Figure 62 Effet du nombre de flux

Ce graphique fait apparaître clairement que le nombre de flux pris en compte par le calcul est lui un critère très important : plus il y aura de flux, et meilleures seront les bornes. On peut tout à fait justifier cette observation en pensant au phénomène suivant : pour obtenir de fortes tailles de files d'attente, il faut que les flux suivent un scénario précis, exigeant par exemple que les arrivées de trame de plusieurs flux soient simultanées. Or, plus les flux sont nombreux plus il est probable que des arrivées de trames soient simultanées, mais moins il est probable qu'elles le soient toutes. Le comportement des bornes est donc tout à fait satisfaisant compte tenu de la compréhension du fonctionnement réel des éléments.

### 3.1.3 Influence de la charge des sources

On représente dans la figure ci-dessous l'influence de la charge des sources. Pour ce faire, on pose qu'un flux est plus chargé qu'un autre si ses paramètres  $\sigma_i$  et  $\rho_i$  sont supérieurs aux paramètres de l'autre flux (bien sûr ce n'est pas une relation d'ordre total). Dans notre étude, nous avons considéré le cas initial, puis des cas pour lesquels on a multiplié par un facteur entier les enveloppes de ces flux, ce qui correspond à une augmentation de charge.

Les résultats obtenus montrent bien que ce paramètre de charge des flux est très important, puisque les bornes se dégradent très vite en fonction de l'augmentation de la charge. Ici encore, ce résultat est conforme à l'intuition, puisqu'il est bien évident que plus les flux sont chargés, plus la probabilité d'obtenir des tailles importantes est élevée.

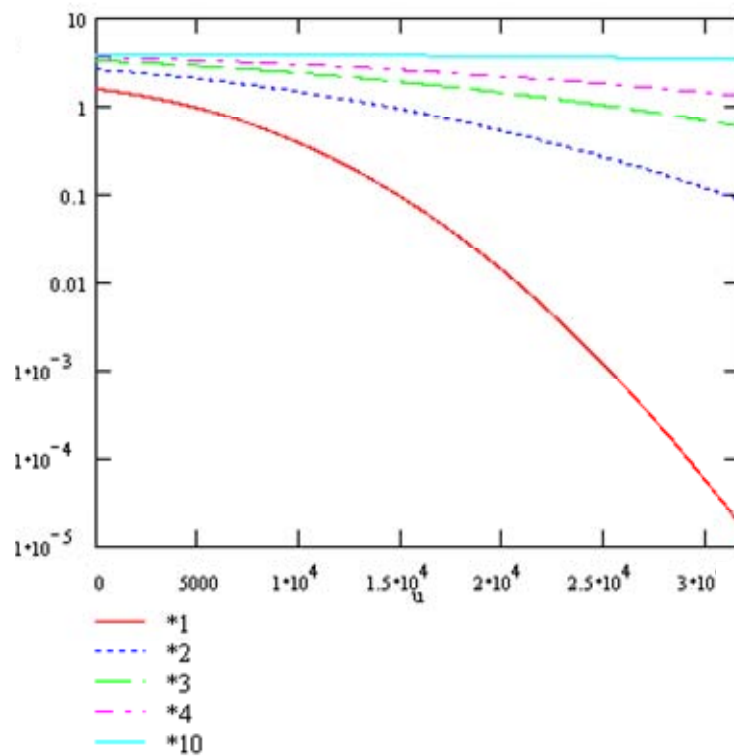


Figure 63 Effet de la charge

### 3.1.4 Influence de la dispersion des sources

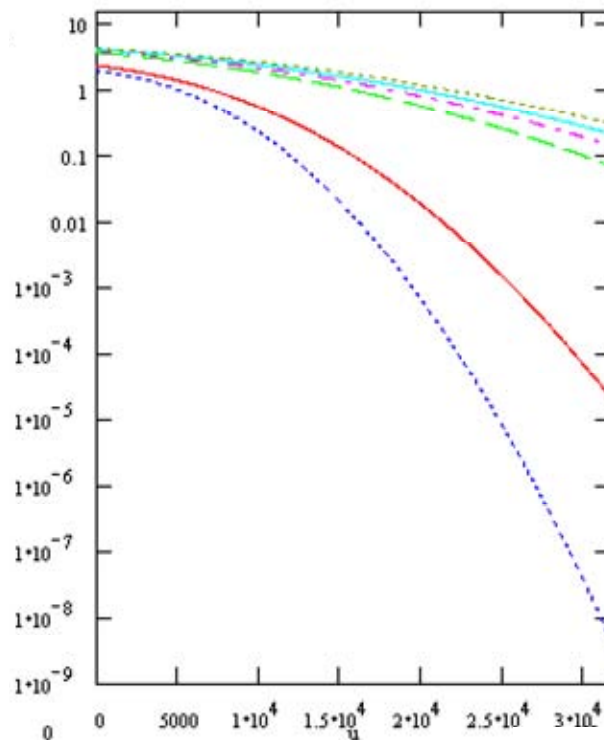


Figure 64 Effet de la dispersion

Nous avons ici voulu étudier l'importance de la « ressemblance » des flux. En effet, les travaux de Vojnović comportent systématiquement deux types de formules, un pour le cas homogène (tous les flux ont même enveloppe), et un autre pour le cas hétérogène. Il semble donc intéressant de voir l'influence de ce paramètre sur les bornes.

Pour ce faire, nous avons décidé de multiplier les paramètres  $\sigma_i$  et  $\rho_i$  de chacun des flux par respectivement : 1, 1,4, 1,8, 2,2, 2,6, 1/2,6, 1/2,2, 1/1,8, 1/1,4 et 1. Nous obtenons ainsi la deuxième courbe (indice 1). Les courbes suivantes sont obtenues en répétant  $j$  fois ces opérations de multiplication pour la courbe d'indice  $j$ . On peut facilement constater que le résultat de ces multiplications successives est bien de rendre les flux de moins en moins semblables, tout en gardant une charge totale constante sur l'élément.

Les résultats sont assez significatifs du fait que plus les flux sont hétérogènes, moins bonnes sont les bornes. Ceci peut être expliqué comme étant directement relié aux bornes de Hoeffding, qui sont au coeur de la démarche. En effet, ces bornes se comportent elles aussi moins bien dans le cas de flux hétérogènes.

Pour résumer l'ensemble de ces expériences, on peut dire que les bornes donnent de meilleurs résultats pour des flux en nombre important, dont la charge est relativement faible, et les caractéristiques homogènes.

### 3.2 Confrontation des bornes de Vojnović à une simulation

Il semble intéressant de conforter notre confiance dans les résultats de Vojnović par une approche de type simulation. Cette simulation a été effectuée en QNAP en considérant des sources dont les temps inter trames suivent une loi normale de moyenne la valeur  $\frac{\rho_i}{S_{\max}^i}$ . Les flux passent ensuite par un dispositif (shaper) de type seau percé avant de pénétrer dans un

multiplexeur. La figure suivante compare les résultats obtenus par simulation aux valeurs théoriques, dans le cas de 10 flux homogènes.

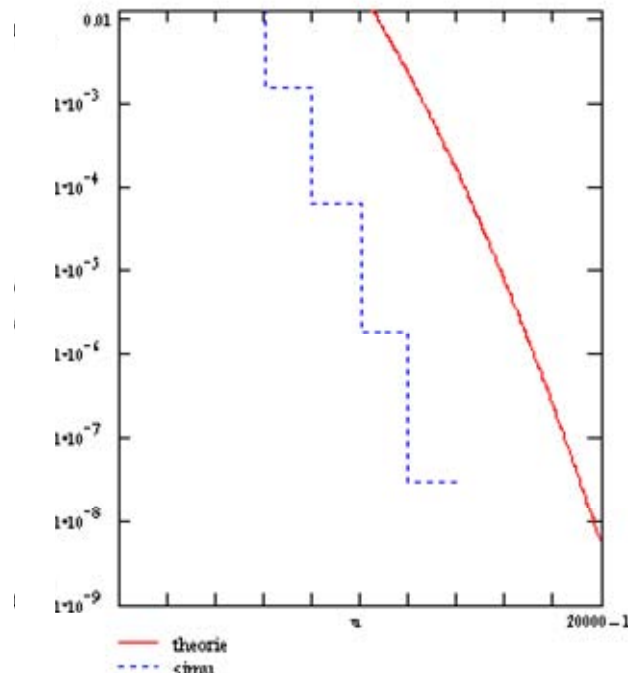


Figure 65 Comparaison résultats théoriques / simulation

Les mesures ont été effectuées avec des trames de taille constante, ce qui explique l'aspect en marche d'escalier : on peut voir ici la « probabilité » mesurée d'avoir 2, 3, 4 ou 5 trames dans la file d'attente (la « probabilité » est en fait une mesure du nombre d'occurrences, qu'on divise ensuite par le nombre d'observations total).

La première constatation est de remarquer que les valeurs théoriques sont bien des bornes sur les valeurs mesurées. On peut également constater que les deux courbes offrent un comportement relativement similaire, ce qui conforte les résultats théoriques. L'écart entre les deux courbes tend à montrer que les bornes théoriques ne sont pas très « serrées » ; cependant, il ne faut pas oublier que nos résultats obtenus par simulation ne prétendent pas à être parfaitement précis, compte tenu de la simplicité du modèle de simulation retenu.

Il semble d'après cette première comparaison que le comportement des bornes est satisfaisant, bien qu'il soit possible de mieux vérifier cette proposition en menant à bien des simulations plus poussées et précises.

Nous concluons de cette étude de type « boîte noire » que les résultats de Vojnović peuvent être intéressants dans notre contexte, mais qu'ils semblent assez sensibles à la façon de les appliquer (nombre de flux notamment) ; nous prévoyons que selon la configuration du réseau et du trafic, les résultats obtenus seront plus ou moins bons.

#### 4 Application de l'approche combinée au réseau étudié

Nous avons appliqué les résultats de Vojnović au cas d'étude présenté dans le chapitre 2 de cette thèse. Nous utilisons le modèle des commutateurs utilisant des files FIFO pour modéliser les ports de sortie des commutateurs. Nous faisons l'hypothèse que tous les flux ont même priorité. Cette hypothèse répond principalement à des raisons de clarté d'exposition,

pour ne pas avoir à présenter une courbe par priorité pour chaque étude. Bien entendu, puisqu'on sait définir une courbe de service offerte à chacune des priorités, les résultats sont applicables pour des flux de priorités différentes.

#### **4.1 Vérification des hypothèses de la méthode**

On commence par vérifier que le réseau avionique considéré répond bien aux hypothèses des travaux de Vojnović.

##### **4.1.1 Indépendance des flux**

Les résultats de Vojnović nécessitent que les flux soient statistiquement indépendants (ou au moins qu'ils le soient au moment où ils pénètrent dans le réseau, pour appliquer la méthode de « fresh traffic majorization », présentée dans le paragraphe 2.2.5.2).

Il est évident que cette hypothèse est vérifiée pour des VL issus de End System différents, puisque ceux-ci sont tous indépendants les uns des autres (aucun contrôle centralisé, par exemple). Par contre, la question se pose à deux autres niveaux : à l'intérieur d'un End system donné, est-ce que les VL issus de deux partitions différentes sont indépendants ? D'autre part, si une même partition émet des données sur deux VL, ceux-ci seront-ils indépendants ?

###### **4.1.1.1 Deux partitions**

Les partitions s'exécutant sur un même équipement partagent des ressources (temps de calcul, mémoire). Selon la façon dont est effectué ce partage, il pourrait se créer des dépendances entre les VL utilisés par deux partitions différentes. Par exemple, ce serait le cas si une partition attendait la fin du traitement d'une autre avant d'émettre une donnée sur le VL qu'elle utilise. Cependant, en avionique, il existe une exigence forte de ségrégation temporelle et spatiale des partitions. En général, la ségrégation temporelle est réalisée en utilisant un ordonnancement statique des tâches, de type « rate monotonic ». Avec ce type d'ordonnancement, il ne peut exister de dépendance entre les VL utilisés par deux partitions différentes.

###### **4.1.1.2 Pour une même partition**

Il est tout à fait possible qu'une partition donnée envoie des données sur deux VL. Selon la façon dont sont gérées ces émissions, il est possible que ces deux VL ne soient pas statistiquement indépendants. Ce serait par exemple le cas si une émission sur le deuxième VL intervenait systématiquement après chaque émission sur le premier. Pour savoir si les VL utilisés par une même partition sont indépendants, il est donc nécessaire de posséder une connaissance fine du fonctionnement de la partition en question. Ne disposant pas de telles informations, nous nous contenterons par la suite d'indiquer la manière d'obtenir des résultats, dans le cas où on peut supposer que ces flux sont bien indépendants, et dans le cas contraire. Bien entendu, les résultats obtenus dans le premier cas seront meilleurs.

##### **4.1.2 Régulation des flux**

Les flux sont bien régulés avant leur entrée dans le réseau, afin de correspondre à la définition du VL. Nous avons vu que ceci constitue même la principale raison pour laquelle nous avons recours à une approche combinée déterministe stochastique. Les résultats sont établis pour une forme quelconque de cette enveloppe, mais nous savons que dans notre cas d'étude les enveloppes initiales sont des enveloppes affines de type  $(\sigma_i, \rho_i)$ .



### 4.1.3 Débit à long terme

La troisième hypothèse signifie que l'accroissement du trafic cumulé pour un flux donné se fait en moyenne au débit  $\rho_i$ . La définition des VL, ainsi que la méthode de calcul déterministe adoptée, nous conduisent en pratique à utiliser des enveloppes affines par morceaux et concaves. Pour celles-ci,  $\rho_i = \lim_{t \rightarrow +\infty} \frac{\alpha_i(t)}{t}$  représente la pente de leur dernier segment.

L'hypothèse est donc naturellement vérifiée, car si ce n'était pas le cas l'enveloppe ne serait pas respectée à long terme.

### 4.1.4 Caractérisation des éléments

On a pu définir lors de nos premières études les courbes de service offertes par chacun des éléments du modèle, à l'ensemble des flux qui les traversent. On rappelle qu'on prend pour courbe de service pour un port de sortie la fonction  $\beta(t) = C.t$ , où  $C$  désigne la capacité du lien physique relié à ce port (généralement 10 ou 100Mbps). La dernière hypothèse des travaux est donc bien vérifiée.

## **4.2 Propositions de méthodes d'application des bornes au réseau complet**

Les résultats de Vojnović s'appliquent a priori sur un seul élément, puisqu'après la traversée d'un élément, l'hypothèse d'indépendance des flux n'est plus vérifiée. En suivant les pistes présentées dans [85], nous proposons deux méthodes permettant d'étendre ce calcul à tout le réseau. Les deux méthodes ont en commun de nécessiter une analyse déterministe préalable du réseau. Pour ce faire, nous appliquons la méthode d'analyse décrite dans le chapitre 3 de cette thèse, en utilisant notamment la notion de groupes, pour obtenir de bons résultats pour les bornes déterministes.

La première méthode proposée consiste à appliquer les résultats de Vojnović en prenant pour entrées les groupes, et non les flux individuels. Les groupes ont en effet la propriété d'être indépendants entre eux à l'intérieur du réseau, ce qui n'est pas le cas des flux individuels à cause de la sérialisation sur les liens physiques.

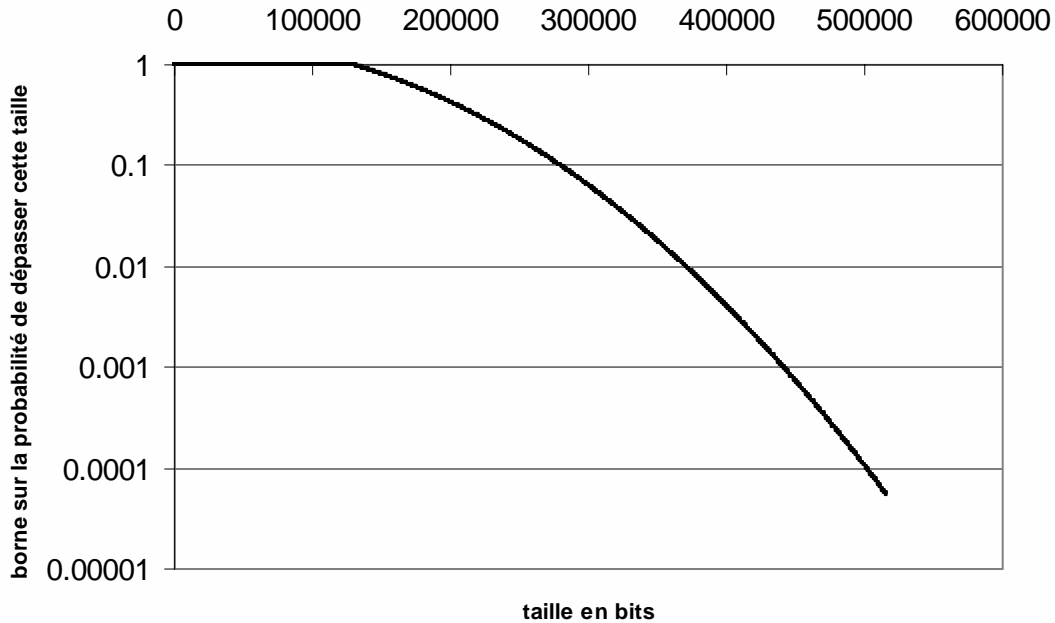
La seconde méthode consiste à utiliser la méthode de « fresh traffic majorization » présentée dans le paragraphe 2.2.5.2. Suite aux résultats obtenus dans la partie 3.1, on s'attend à ce que cette seconde méthode fournisse de bien meilleurs résultats. En effet, l'utilisation des groupes dans la première méthode diminue le nombre de flux pris en compte pour le calcul, et ces flux sont plus chargés, ce qui tend à dégrader les résultats.

Les graphiques suivants sont obtenus sur le port de sortie pour lequel le calcul déterministe donne la plus forte borne, 516600 bits en l'occurrence. On effectue les calculs avec 5 subdivisions de l'intervalle  $[0, \tau]$ .

## **4.3 Résultats obtenus**

### 4.3.1 Première méthode

Le graphique ci-dessous, comme tous ceux de cette partie, représente en abscisse la taille de la file d'attente, et en ordonnée une majoration de la probabilité de dépasser cette valeur. Par exemple, on peut lire ci-dessous que la probabilité de dépasser une taille de 300000 bits est inférieure à  $10^{-1}$ .



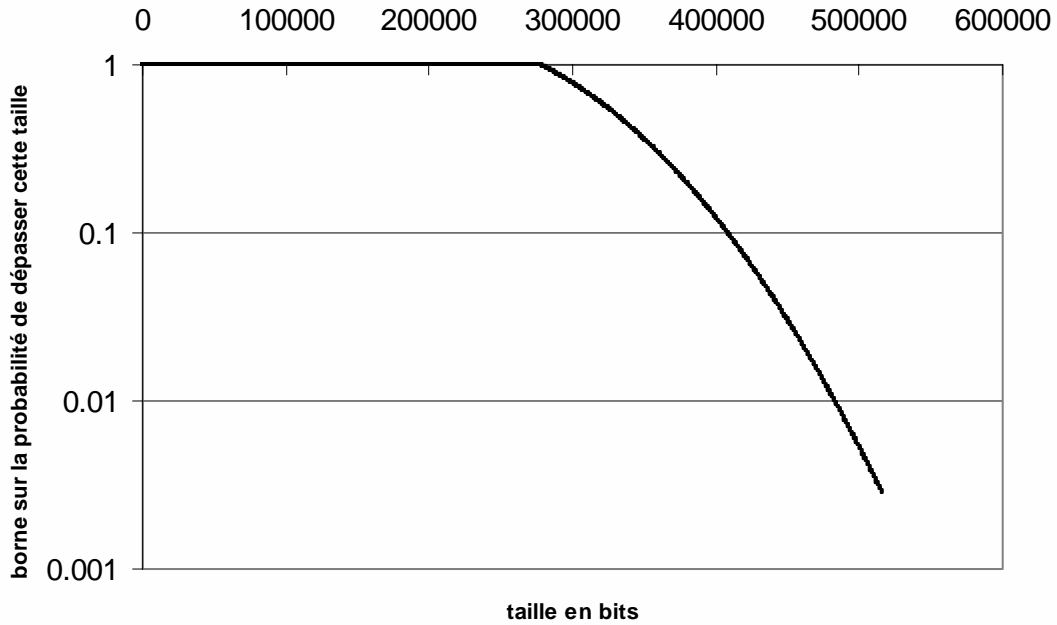
**Figure 66 Application avec les groupes**

Ces résultats sont assez décevants, dans le sens où on peut y lire que la probabilité de dépasser la borne déterministe connue est inférieure à 0.0001, alors que par ailleurs on sait que cette probabilité est nulle. Il semble donc que cette méthode d'application ne donne pas de résultats exploitables.

#### 4.3.2 Deuxième méthode

Dans un premier temps, on suppose qu'on ne sait pas si les VL issus d'un même End System sont indépendants entre eux ou non. La seule information dont on dispose alors est que les émissions de deux End System différents sont indépendantes. On applique donc la méthode de « fresh traffic majorization », mais en considérant comme flux initiaux indépendants les groupes issus de chaque End-System. Plus précisément, lors du calcul dans un port de sortie d'un commutateur donné, on regarde l'origine des flux qui le traversent. Pour chaque End System origine, on calcule l'enveloppe du groupe formé par tous les flux qui passeront ensuite dans le port considéré.

Voici les résultats obtenus, toujours pour le port possédant la borne déterministe la plus grande.



**Figure 67 Application "fresh traffic majorization" aux groupes sortant des ES**

Encore une fois, on peut voir que les résultats obtenus avec cette hypothèse ne sont pas bons, puisqu'ils sont encore moins bons qu'avec la méthode précédente. Ceci est normal dans le sens où on utilise à peu près le même nombre de flux (groupes) pour le calcul que dans cette méthode, mais qu'ils sont plus chargés, à cause de la méthode de « fresh traffic majorization ».

Dans un second temps, nous avons considéré au contraire que nous disposons de la connaissance que tous les VL sont indépendants entre eux. Les résultats doivent être bien meilleurs que précédemment, puisqu'on utilisera dans les calculs tous les flux initiaux, et non les groupes issus de chaque End System.

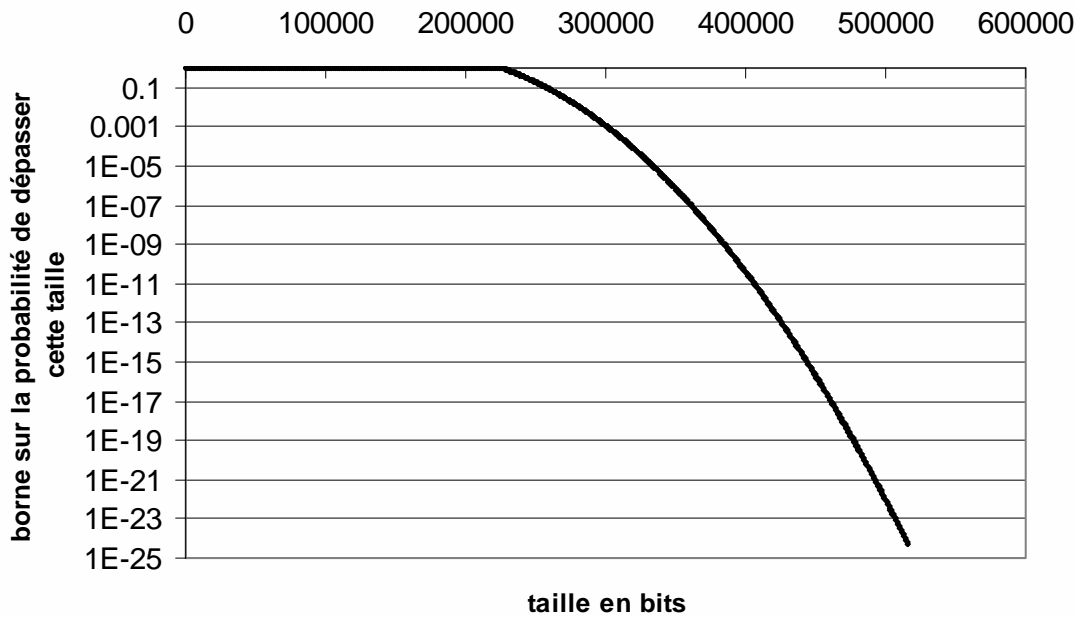


Figure 68 Application "fresh traffic majorization" avec l'hypothèse flux indépendants

Les résultats obtenus par cette méthode sont bien plus satisfaisants, et comme on pouvait s'y attendre bien meilleurs qu'avec la première méthode. On voit par exemple que la probabilité de dépasser 75% de la borne déterministe est inférieure à  $10^{-9}$ .

Selon la connaissance disponible sur l'indépendance relative des VL, il est possible d'obtenir des résultats en suivant notre méthode d'analyse, en regroupant les flux dont on ne sait pas s'ils sont indépendants. Les résultats obtenus seront alors compris entre les deux extrêmes que nous venons de présenter. Dans la suite de cette thèse, les résultats présentés feront l'hypothèse que tous les VL sont bien indépendants.

Grâce à cette approche combinée, on confirme aussi l'intuition présentée dans le paragraphe 1.11.1, à savoir que la borne déterministe est très pessimiste par rapport aux valeurs qu'on pourrait qualifier de « probables ». Plus exactement, ces résultats indiquent que les probabilités d'occurrence des cas provoquant de grandes tailles de files d'attente sont très faibles.

#### 4.4 Obtention de bornes en trames

L'ensemble des formules précédentes concerne les probabilités que la taille de la file d'attente dépasse un certain nombre de bits. Dans le cas particulier où la taille de la file d'attente est comptée en trames, et non en bits, les résultats manquent. Dans un premier temps, on pourrait penser effectuer l'ensemble de l'analyse en trames, comme indiqué dans le chapitre 3 de cette thèse. On appliquerait donc les formules précédentes avec les enveloppes des flux en trames, et pour courbe de service  $\beta(t) = \frac{Ct}{S_{\max}}$  pour chaque port de sortie des commutateurs, c'est-à-

dire supposer de manière pessimiste que le commutateur ne fait sortir que des trames de taille maximale. Malheureusement, on a pu constater qu'avec ces courbes de service, les multiplexeurs ne sont pas stables : les courbes d'arrivée et de service ne se coupent plus. On ne peut donc pas appliquer les formules de cette manière.

On adopte alors la démarche suivante : l'évènement « la file d'attente a plus de N trames » implique l'évènement « la taille de la file d'attente est supérieure à  $N \cdot S_{\min}$  bits ». On a donc la relation suivante pour les probabilités :

$$P(Q_f \geq N) \leq P(Q_b \geq N \cdot S_{\min})$$

A partir des résultats précédents, une simple division par  $S_{\min}$  permet donc d'obtenir une majoration sur la probabilité d'avoir plus de N trames dans la file d'attente :

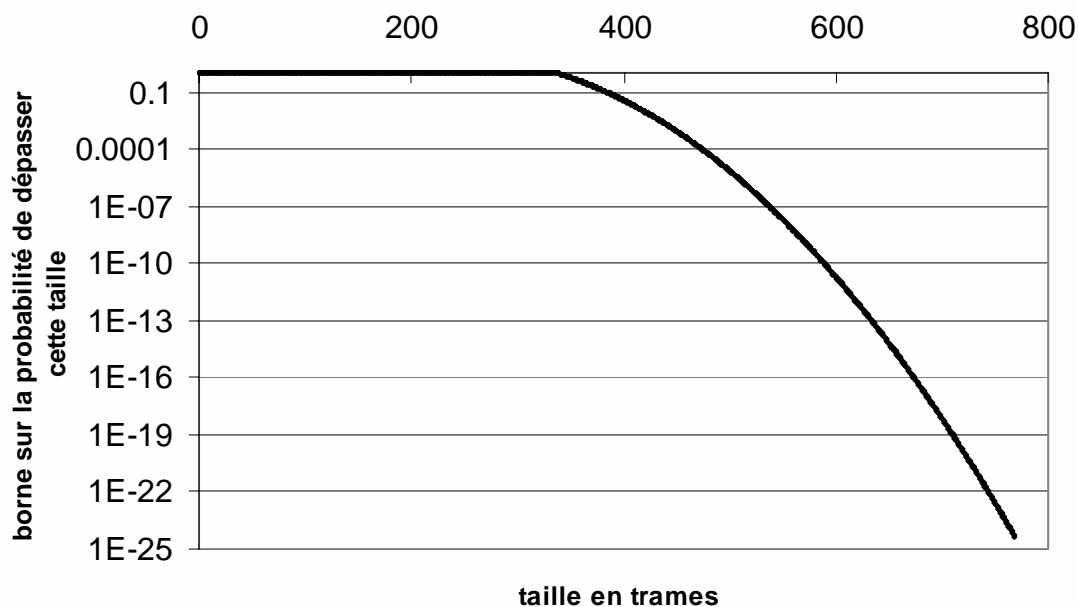


Figure 69 Cas de la taille comptée en nombre de trames

Encore une fois, l'ordre de grandeur des résultats est intéressant, puisqu'il est identique à celui obtenu pour la taille en bits. On peut noter que la majoration effectuée pour obtenir le résultat précédent est assez grossière, puisqu'il est très probable qu'une taille supérieure à  $N \cdot S_{\min}$  bits soit le résultat de l'arrivée dans la file d'attente de trames de taille différente de  $S_{\min}$ . Il doit donc être possible de découvrir une autre démarche pour aboutir à de meilleurs résultats dans le cas où la taille de la file d'attente est comptée en nombre de trames.

#### 4.5 Obtention de bornes sur le délai

Dans le cas général d'un élément seulement caractérisé par la courbe de service offerte à un flux agrégé, il n'est pas possible de trouver une borne sur le délai à partir de la borne sur la taille de la file d'attente. C'est d'ailleurs ceci qui a conduit à utiliser le modèle de PSRG dans la définition actuelle du PHB EF de DiffServ. Cependant, pour un élément FIFO à capacité de sortie C constante, on peut obtenir des résultats : le délai subi par le premier bit d'une trame est constitué par le temps nécessaire pour émettre au débit C la quantité de bits déjà présents dans la file FIFO. Cette quantité est majorée grâce au résultat présenté dans le paragraphe 2.2.5.1, à partir des probabilités de Palm. Ainsi, pour obtenir une borne sur la répartition du délai subi par le premier bit d'une trame, il suffit de prendre la courbe donnant la répartition de la taille de la file d'attente au moment de l'arrivée de la trame, et de diviser les abscisses par C. Le dernier bit attendra quant à lui au maximum  $S_{\max}/C$  de plus, où  $S_{\max}$  représente la taille maximale d'une trame pour tous les VL passant dans l'élément considéré. A partir de la courbe précédente, une simple translation de  $S_{\max}/C$  donne donc la probabilité de dépasser un certain délai. La figure ci-dessous a été obtenue de cette manière ; on rappelle que dans le cas considéré, la borne déterministe sur le délai est de 5,163ms.

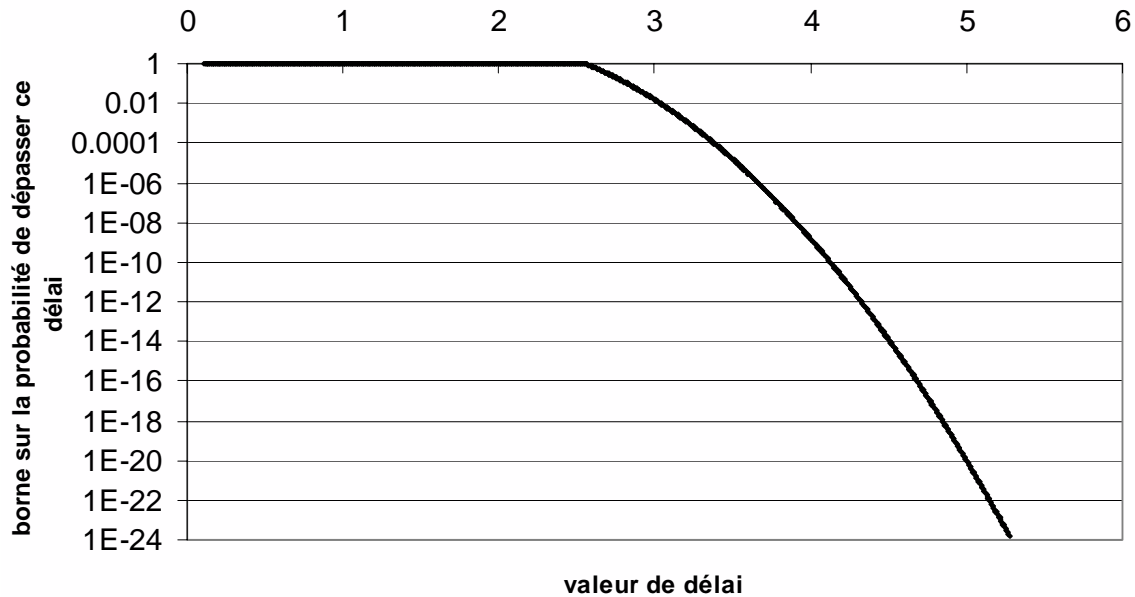


Figure 70 Répartition des délais (ms)

Ici encore, on peut lire que l'application des résultats de Vojnović donne des résultats d'un ordre de grandeur suffisamment faible pour permettre une utilisation pratique. Par exemple, la probabilité de dépasser 75% du délai maximum est inférieure à  $10^{-9}$ .

#### 4.6 Illustration de l'intérêt pratique de ces bornes

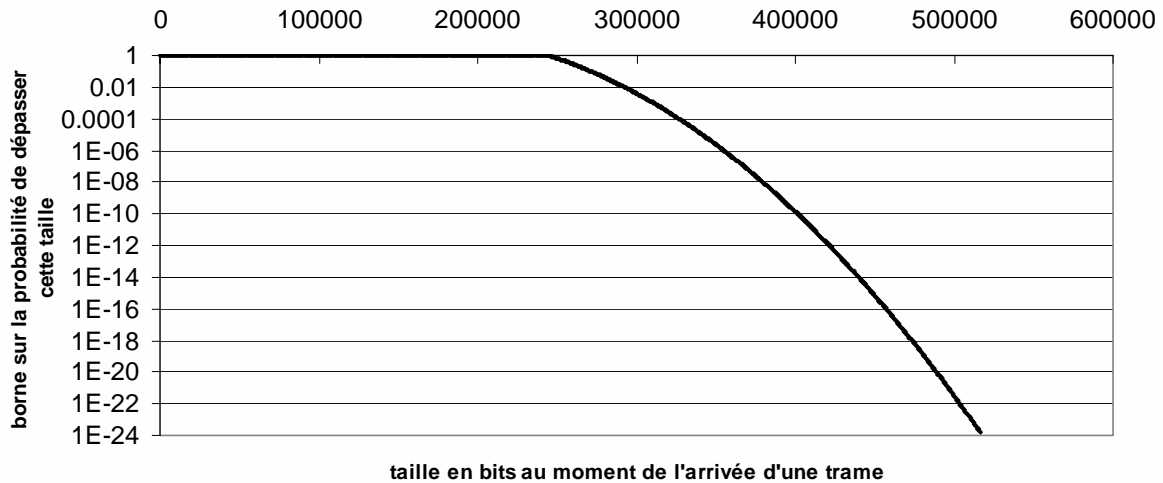
Nous souhaitons montrer dans ce paragraphe comment les résultats précédents peuvent être utilisés en pratique.

##### 4.6.1 Premier exemple

Supposons qu'on souhaite dimensionner la taille maximale (notée  $Q$ ) de la file d'attente précédente, de sorte qu'on puisse garantir sur 100h de fonctionnement une probabilité de perte de trame inférieure à  $10^{-9}$ .

On commence par calculer combien de trames peuvent arriver au maximum en 100h. On calcule donc la valeur de l'enveloppe globale (somme des enveloppes de tous les flux) pour la valeur  $t = 60 \times 60 \times 100 = 360000s$ . On trouve  $1,06 \cdot 10^{13}$  bits, ce qui correspond dans le pire des cas (avec une taille minimale de trames de 84 octets) à 15.64 milliards de trames.

Dans un deuxième temps, on cherche à évaluer la probabilité de perdre une trame. Une trame sera perdue si, au moment où elle arrive, la taille courante de la file d'attente est égale à  $Q$ . Ceci impliquerait que le nombre de bits présents dans la file, au moment de l'arrivée de la trame, serait supérieur à  $Q \times S_{\min}$ . La probabilité de perdre une trame est donc inférieure à la probabilité que la taille de file d'attente soit supérieure à  $Q \times S_{\min}$  bits. Cette probabilité est bornée par la courbe suivante, obtenue conformément au paragraphe 2.2.5.1.



La probabilité de perdre une trame en 100h est inférieure au nombre de trames pouvant passer dans le port considéré durant 100h, multiplié par la probabilité que chacune d'entre elle soit perdue au moment de son arrivée. Pour atteindre l'objectif fixé, il faut donc que  $Q$  soit tel que la probabilité de perdre chacune des trames soit inférieure à  $6.39 \cdot 10^{-20}$ . Ceci est le cas pour des tailles supérieures à 473215 bits, ce qui implique  $Q$  supérieur ou égal à 705 trames.

On vient donc de montrer que dimensionner la taille de file d'attente à 705 trames suffit pour garantir que sur 100h de fonctionnement, la probabilité de perdre une trame est inférieure à  $10^{-9}$ , ce qui représente un gain d'environ 8% par rapport à un dimensionnement fondé sur la seule borne déterministe.

#### 4.6.2 Deuxième exemple

Un autre cas pratique d'utilisation est le suivant : l'analyse déterministe du réseau fournit comme borne pour le port le plus chargé du réseau 769, or la taille physique de la file d'attente est inférieure de 20 % à cette valeur (on prend comme exemple 612). On ne peut donc pas garantir avec cette borne que la file d'attente ne débordera jamais ; on va utiliser les résultats ci-dessus pour majorer cette probabilité de perte de trame, par exemple sur une heure de fonctionnement.

Avec le même calcul que précédemment, on montre que la file d'attente reçoit au maximum 156,4 millions de trames en une heure. Il y aura perte de trame s'il se trouve plus de 612 trames ( $612 \times 84 \times 8 = 411264$  bits au minimum) dans la file au moment où en arrive une nouvelle. Donc la probabilité de perdre une trame sera inférieure à la probabilité que la taille de file d'attente dépasse 411264 bits, soit  $1 \cdot 10^{-11}$ . La probabilité de perdre une trame durant une heure de fonctionnement sera donc inférieure à  $156,4 \cdot 10^6 \times 1 \cdot 10^{-11} = 1,5 \cdot 10^{-3}$ , ce qui correspond à moins d'une trame perdue toutes les 600.

## Conclusion

Le but de ce chapitre était de compléter notre analyse du fonctionnement du réseau, en essayant d'estimer les probabilités d'occurrence des cas pires, pour lesquels nous avons majoré certaines quantités dans les chapitres précédents. Pour ce faire, nous avons utilisé des résultats présentés par Vojnović, qui propose une méthode combinée déterministe-stochastique. Comme pour le calcul déterministe, nous avons montré que cette méthode

s'applique bien au contexte considéré, notamment grâce à la notion de VL qui permet de bien caractériser les entrées dans le réseau. D'autre part, nous avons proposé une méthode permettant d'appliquer ces résultats à un réseau avionique, en fonction de la connaissance du réseau dont on dispose. Enfin, nous avons montré comment obtenir à partir de ces résultats des informations sur la répartition du nombre de trames dans une file d'attente, ou sur le délai subi par les trames.

Les résultats obtenus confirment bien l'intuition que les bornes déterministes sont largement supérieures aux valeurs atteintes dans la très grande majorité des cas. L'ordre de grandeur de cette différence est même très intéressant : on sait en effet qu'en aéronautique, on accepte dans le cadre de la méthode de conception « safe-life » des probabilités de  $10^{-9}$  ce qui est tout à fait le cas ici.

Dès lors, on peut songer utiliser cette méthode pour atteindre deux buts précis. Le premier serait naturellement de redimensionner les éléments du réseau. On pourrait en effet concevoir de relâcher la contrainte de non perte de trame, pour adopter une contrainte du type « probabilité de perdre une trame  $< \varepsilon$  », avec  $\varepsilon$  connu. De ce fait, les tailles de files d'attente pourraient être largement diminuées. Bien entendu, une telle approche ne saurait être retenue qu'en accord avec les autorités de certification.

Une autre application de la méthode combinée peut concerner les procédures de test, par exemple lors de la recette des commutateurs. En effet, compte tenu du débit du réseau, il est très difficile d'équiper les appareils pour mesurer des valeurs en temps réel. Un des rares paramètres que l'on peut mesurer est le nombre de trames perdues, puisque ces événements doivent être enregistrés par les commutateurs selon la norme ARINC 664. Dès lors, en faisant varier le trafic entrant dans un commutateur, on peut vérifier qu'il suit bien les probabilités de perte prévues par la méthode.

Les travaux futurs devraient chercher à établir de meilleures bornes sur la probabilité de perdre des trames, alors que la méthode actuelle est entièrement dérivée du domaine des bits. D'autre part, il faudrait également établir des formules concernant la probabilité de perte de trames à long terme, qui seraient plus faciles à interpréter dans le contexte aéronautique ; on rappelle que pour l'instant Vojnović n'a démontré des formules de ce genre que dans le cas temps discret.

Un axe de recherche que nous avons envisagé est celui d'une méthode d'analyse fondée exclusivement sur ces résultats de type statistique, sans recours à une analyse déterministe préalable. L'idée fondatrice de cette étude est qu'à partir de la fonction de répartition du délai obtenue, on peut obtenir un encadrement de l'enveloppe des flux en sortie : on sait majorer la probabilité que l'accroissement d'un flux durant un intervalle donné soit comprise entre deux valeurs. De cette façon, la propagation des enveloppes n'utiliserait plus les résultats déterministes.



# CONCLUSION

## 1 Résumé des travaux

La première partie de cette thèse a permis de remettre dans son contexte le type de réseau étudié, à savoir les réseaux déterministes au sens du projet de norme ARINC 664. Compte tenu de la criticité des informations circulant sur le réseau, il convient de prouver que le réseau satisfera aux contraintes de qualité de service exigées. Pour ce faire, nous avons montré dans une première étude (chapitre 2) que les résultats de Cruz sur le Network Calculus s'appliquent dans ce contexte, principalement grâce au fait que toute la gestion du réseau est entièrement entre les mains de l'avionneur : celui-ci possède une connaissance précise de chaque élément du réseau, de tous les flux qui y circulent, et peut choisir la manière dont ceux-ci seront routés sur le réseau. Les théorèmes du Network Calculus permettent de vérifier si les contraintes de sécurité sont respectées par le réseau ainsi utilisé, en établissant des bornes sur les latences locales, de bout en bout, et sur les tailles des files d'attente. Nous avons vérifié que ces bornes possèdent un ordre de grandeur satisfaisant, qui permet d'envisager leur utilisation pour dimensionner le réseau. Cette première étude nous a également permis de montrer que ces bornes dépendent principalement de la régularité des flux circulant dans le réseau.

Par la suite (chapitre 3), nous avons apporté plusieurs modifications à cette méthode d'analyse fondée sur le Network Calculus. Certaines de ces modifications visent à améliorer les bornes obtenues par la méthode : utilisation d'une caractérisation plus fine du fonctionnement des éléments (courbe de service), prise en compte de l'effet de sérialisation entre flux induit par les liens physiques (notion de groupes). Les autres modifications permettent d'affiner l'analyse, par exemple en permettant de l'effectuer en termes de trames ou de bits, ou bien de lui ajouter de nouvelles fonctionnalités, comme la prise en compte de l'effet de policing ou de flux de priorités différentes.

Munis de cette méthode d'analyse du comportement du réseau, nous avons voulu présenter dans le chapitre 4 des méthodes permettant d'améliorer le réseau vis-à-vis d'elle, c'est-à-dire que ces méthodes vont permettre de diminuer les bornes calculées au cours de l'analyse. Les premières de ces méthodes visent à choisir une bonne affectation des priorités pour les différents VL. Nous avons notamment présenté en ce sens une heuristique, reposant sur la taille minimale des trames des VL, ainsi qu'un outil plus complexe utilisant des algorithmes d'évolution multicritères. Les autres méthodes proposées s'intéressaient à l'architecture matérielle du réseau. Nous avons ainsi montré pourquoi et comment utiliser plusieurs liens physiques pour relier les commutateurs entre eux, et nous avons également montré les avantages et les inconvénients d'une régulation des flux dans les commutateurs.

Enfin, nous avons voulu mieux comprendre le lien entre les bornes calculées par notre méthode d'analyse, et le comportement en moyenne du réseau. Nous nous sommes particulièrement intéressés dans ce sens aux récents résultats du calcul réseau probabiliste, qui s'appuie sur le Network Calculus déterministe que nous avons utilisé dans notre méthode. Nous avons montré comment développer une méthode d'analyse avec ces résultats dans notre contexte. Les résultats de cette méthode sont constitués de bornes, dont on sait qu'elles n'auront qu'une très faible probabilité d'être dépassées, ce qui pourrait constituer la base d'une nouvelle méthode de certification pour de futurs programmes.

## 2 Synthèse de la démarche

Les recherches effectuées dans le cadre de cette thèse peuvent être regroupées en trois grandes thématiques :

- déterminer une méthode d'analyse du réseau
- mieux comprendre le fonctionnement du réseau
- donner des méthodes permettant d'optimiser le réseau

Nous tenons à souligner les rapports très fréquents entre ces trois thématiques.

### Méthode d'analyse / Méthodes d'optimisation

Les méthodes d'optimisation que nous avons proposées reposent sur la méthode d'analyse que nous avons présentée. En fait, l'optimisation recherchée porte toujours sur des critères, qui sont évalués par notre méthode d'analyse.

### Méthode d'analyse / Compréhension du réseau

La méthode d'analyse du réseau permet de mieux comprendre le fonctionnement du réseau, par exemple en mettant en évidence l'importance de la régularité des flux. En fait, l'étude de type Network Calculus permet d'établir le contour du domaine d'utilisation de ce réseau, en précisant ce qu'il advient lors de cas pires. De plus, l'approche combinée déterministe stochastique ouvre de nouvelles perspectives d'études des régions proches de ce contour. Réciproquement, notre méthode d'analyse a tiré parti de la connaissance du fonctionnement du réseau, notamment grâce à l'utilisation de la notion de groupes.

### Méthodes d'optimisation / Compréhension du réseau

Les méthodes d'optimisation que nous avons présentées utilisent énormément notre connaissance du réseau : les heuristiques proposées pour affecter les priorités utilisent l'intuition qu'il vaut mieux mettre en priorité haute les trames les plus petites. La manière de placer les flux sur les nouveaux liens prend en compte l'importance que l'on sait grande de l'effet de groupe. A l'inverse, ces méthodes nous ont permis de mieux comprendre certains détails du fonctionnement du réseau : par exemple, l'algorithme génétique multicritère utilisé nous a permis de mieux comprendre les termes de l'échange taille maximale – délai moyen. Cet équilibre a également été étudié lors de notre étude sur une méthode de régulation des flux. Enfin, l'étude sur la duplication des liens physiques entre commutateurs a montré l'influence globale de modifications mineures locales.

On voit ainsi à quel point les recherches dans un secteur donné ont influencé les autres, ce qui a permis simultanément de mieux comprendre le fonctionnement du réseau, de proposer une méthode d'analyse complète de ce type de réseau, et finalement de présenter plusieurs méthodes permettant d'optimiser ces réseaux, tout en aidant leur utilisateur à mieux comprendre l'impact de ces modifications.

D'autre part, la problématique que nous avons dégagée était la suivante : l'évolution de la complexité des systèmes avioniques a conduit à l'utilisation d'une nouvelle technologie de réseaux embarqués, l'Ethernet commuté Full-Duplex. Cette nouvelle technologie pose de nouveaux problèmes, parmi lesquels un des plus importants est qu'il est nécessaire de prouver que ce réseau offre des garanties de qualité de service suffisantes : pas de perte de trames par congestion des commutateurs, temps de traversée du réseau borné. Tout le problème était donc de trouver une méthode permettant d'apporter cette preuve. La première contribution de cette thèse est justement de proposer une telle méthode, en utilisant des résultats du Network Calculus.

Dans un deuxième temps, une nouvelle problématique s'est dégagée : sachant que le réseau sera certifié suivant cette méthode, comment régler les quelques degrés de liberté du réseau étudié pour améliorer les résultats de l'analyse ? Nous avons proposé en ce sens des méthodes permettant sinon de faire le choix, du moins de mieux comprendre les conséquences de ces choix.

Enfin, nous avons pensé que ces réflexions nous menaient à la question plus générale du dimensionnement du réseau. La dernière contribution de cette thèse est une méthode, fondée sur des résultats récents combinant les côtés stochastique et déterministe, qui donne de réelles indications sur comment dimensionner les files d'attente.

### **3 Perspectives de recherche**

Nos travaux ont permis de mettre en lumière plusieurs nouvelles pistes de recherche, que nous avons regroupées au sein des trois domaines évoqués précédemment.

En ce qui concerne la méthode d'analyse, nous pensons qu'il serait intéressant d'utiliser une nouvelle définition des groupes, qui permette des groupes « mixtes », à savoir formés de flux appartenant à des VL de priorités différentes. Nous pensons aussi qu'il serait intéressant de regarder les courbes de service de bout en bout offertes à certains flux, par opposition aux courbes de service locales étudiées ici. Enfin, nous avons vu qu'une contrainte très forte introduite par la méthode d'analyse est qu'il ne peut y avoir de boucles de trafic dans le réseau. Découvrir un moyen de propager le calcul même dans ce cas de figure serait d'un intérêt industriel certain.

Pour les différentes méthodes d'optimisation du réseau, nous pensons que la question de l'affectation des priorités peut encore susciter de nombreux travaux, peut être encore une fois lorsque la connaissance du réseau aura augmenté. De plus, il pourrait être intéressant de s'intéresser à la manière de router les flux sur le réseau. On sait ce problème complexe (nombreuses contraintes de sécurité), mais très important pour la méthode d'analyse, puisque actuellement il faut éviter la formation de boucles de trafic. Enfin, nous avons pris comme hypothèse dans notre étude que les caractéristiques des différents VL ne pouvaient être modifiées. Nous pensons qu'une source importante d'amélioration des bornes réside dans une redéfinition de ces VL. Peut-être est-il intéressant de faire plusieurs VL au lieu d'un seul très chargé, ou bien de définir un VL pour transmettre des trames de grande taille, et un autre pour les trames plus petites ?

Enfin, il reste vraisemblablement encore beaucoup d'études à accomplir pour mieux comprendre le comportement temporel du réseau. En premier lieu, nous pensons qu'une démarche de simulation compléterait notre analyse, puisqu'elle permettrait de visualiser le comportement « en moyenne » du réseau, au lieu de visualiser seulement les cas pires de fonctionnement. Les bornes de type probabiliste que nous avons établies doivent être cohérentes avec des résultats obtenus par simulation. En effet, on peut dimensionner les files d'attentes de sorte que la probabilité de perdre une trame soit plus importante, et sortir ainsi du cadre des événements rares qui sont difficiles à observer par simulation. On pourrait ainsi valider le modèle de simulation s'il permettait de visualiser ces événements plus probables. Une fois ce modèle validé, il pourrait à son tour servir à valider des éléments matériels, par comparaison avec les résultats de tests sur des bancs d'essai. Il faudrait également poursuivre l'étude combinée déterministe – stochastique qui a été amorcée ici, et qui est certainement porteuse de résultats intéressants, tant du point de vue recherche que du point de vue industriel. Enfin, nous pensons que notre étude sur l'affectation des priorités a soulevé un problème intéressant : nous avons mis en avant le fait que nos méthodes pourraient être

améliorées si on savait quels VL étaient très dépendants les uns des autres. Pour reprendre l'analogie avec le génome humain, il s'agirait d'identifier les gènes, c'est-à-dire regrouper les peptides (priorité d'un VL donné) qui doivent être modifiés en même temps.

L'ampleur du travail suggéré dans ces perspectives montre bien à quel point les travaux en ce domaine posent des questions complexes. L'évolution constatée dans le contexte aéronautique en faveur de l'utilisation de réseaux embarqués de type Ethernet commuté a également introduit de nouveaux problèmes. Ceux-ci sont d'autant plus épineux que les ressources disponibles à bord d'un avion sont limitées : on ne peut pas simplement contourner le problème en augmentant la bande passante, ce qui représente souvent la solution la plus simple dans le cadre des réseaux classiques. L'utilisation de cette nouvelle technologie, qui présente au demeurant un grand nombre d'avantages, va donc nécessiter de nouvelles techniques pour la certification, pour le dimensionnement et pour l'analyse de performances des réseaux. Il faudra désormais aux avionneurs développer un vrai savoir-faire réseau : calcul déterministe, simulations, tests sur des bancs d'essai font partie des outils indispensables aux futurs ingénieurs qui travailleront sur ces réseaux.

## BIBLIOGRAPHIE

- [1] Aeronautical Radio Inc., ARINC project paper 636 : Onboard Local Area Network
- [2] Aeronautical Radio Inc., ARINC project paper 646 : ELAN Ethernet Local Area Network
- [3] Spitzer C. R., Digital Avionics Systems, McGraw-Hill Inc., 1993.
- [4] Chanet P., Cassigneul V., « How to control the increase in the complexity of civil aircraft on-board systems », *AGARD Meeting on Aerospace Software Engineering for Advanced Systems Architectures*, mai 1993, Paris.
- [5] Martin F., Modélisation et évaluation de performances prévisionnelles d'architectures modulaires intégrées, Thèse de doctorat, ENSAE, 1999.
- [6] Requirements and Technical Concepts for Aviation, DO-178B: Software Considerations in Airborne Systems and Equipment Certification, 1992.
- [7] B. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, Luglio 1994.
- [8] J.P. Georges, E. Rondeau, T. Divoux, Evaluation of switched Ethernet in an industrial context by using the network calculus, 4<sup>th</sup> IEEE International workshop on Factory Communication Systems, WFCS'2002, Västerås, Suède, pages 19-26, 28-30 août 2002.
- [9] SAE ARP4754, Certification Considerations for Highly-Integrated or Complex Aircraft Systems, 1996.
- [10] Aeronautical Radio Inc., ARINC specification 429-ALL: Mark 33 Digital Information Transfer System (DITS) Parts 1, 2, 3, 2001.
- [11] Aeronautical Radio Inc., ARINC specification 629: Multi-transmitter Data Bus Part 1-Technical Description, 1999.
- [12] Aeronautical Radio Inc., ARINC specification 651: Design Guidance for Integrated Modular Avionics, 1991.
- [13] Aeronautical Radio Inc., ARINC project paper 653: Avionics Application Software Standard Interface, 1997.
- [14] Aeronautical Radio Inc., ARINC project paper 659: Backplane Data Bus, 1993.
- [15] Projet européen de recherche VICTORIA, "Validation platform for Integration of standardised Components, Technologies and tools in an Open, modular and Improved Aircraft electronic System", [www.euproject-victoria.org](http://www.euproject-victoria.org).
- [16] J. Huysseune, P. Palmer, "NEVADA – PAMELA – VICTORIA, "Toward the definition of new aircraft electronics", Invited paper Aeronautics days, Hamburg 2001.
- [17] IETF, RFC 1350: The TFTP Protocol (Revision 2).
- [18] MIL-STD-1553B Specification: Aircraft Internal Time Division Command/Response Multiplex Data Bus, U.S. Department of Defense, Aeronautical Systems Division, 1978.
- [19] Delorme M., Dussaussois C., Gabrilot P., « Dossier de synthèse des architecture étagère: IDEE Tâche 1.4, Synthèse des architectures étagère », Document Airbus, ref. ID3-DOC-ET14-1-AB, 1995.
- [20] IEEE 802.1D, 1998 Edition: Local and Metropolitan Area Networks: Media Access Control Level Bridging
- [21] IEEE 802.1p: LAN Layer 2 QoS/CoS Protocol for Traffic Prioritization
- [22] IEEE Std 802.3, 1998 Edition: Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier senses multiple access with collision detection (CSMA/CD) access method and physical layer specifications
- [23] Internet Engineering Task Force, RFC 1122: Requirements for Internet Hosts : Communication Layers - Required Internet Standard.
- [24] "Comparaison de bus avioniques", C. Fraboul, F. Frances, Rapport technique.
- [25] J. Ermont, « Une algèbre de processus pour la modélisation et la vérification des systèmes temps-réel avec préemption », Thèse de doctorat, ENSAE, 2002.
- [26] R. Cruz, "A calculus for network delay, Part I: Network element in isolation," IEEE Trans. Inform. Theory, vol. 37, no. 1, pp. 114-131, Jan. 1991.
- [27] R. Cruz, "A calculus for network delay, Part II: Network analysis," IEEE Trans. Inform. Theory, vol. 37, no. 1, pp. 132-141, Jan. 1991.
- [28] J. R. Perkins and P. R. Kumar, "Stable distributed real-time scheduling of flexible manufacturing/assembly/disassembly systems," Proceedings of the 27th IEEE Conference on Decision and Control, Austin, Texas, pp. 1863--65, December 1988.
- [29] Kumar, P.R. and T.I. Seidman. (1990). Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. IEEE Transactions on Automatic Control, 35, 289-298.

- [30] S.H.Lu, P.R.Kumar, "Distributed Scheduling Based on Due Dates and Buffer Priorities", IEEE Transactions on Automatic Control, Vol. 36, n. 12, December 1991, pp. 1406-1416.
- [31] Parekh, A. K., & Gallager, R. G., "A Generalized Processor Sharing Approach to flow control in Integrated Services Networks - The Single Node Case," IEEE/ACM Transactions on Networking, Vol 1 #3, pp 344-357, June 1993.
- [32] Parekh, A. K., & Gallager, R. G., "A Generalized Processor Sharing Approach to flow control in Integrated Services Networks - The Multiple Node Case," IEEE/ACM Transactions on Networking, Vol 2 #1, pp 137-150, April 1994.
- [33] J.-Y. Le Boudec and P. Thiran, Network Calculus, Springer Verlag Lecture Notes in Computer Science volume 2050.
- [34] K.Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, C. Diot. A Pragmatic Definition of Elephants in Internet Backbone Traffic. In ACM Sigcomm Internet Measurement Workshop, Marseille, France, November 2002.
- [35] Parekh, A. K., & Gallager, R. G., "A Generalized Processor Sharing Approach to flow control in Integrated Services Networks - The Single Node Case," IEEE/ACM Transactions on Networking, Vol 1 #3, pp 344-357, June 1993.
- [36] Parekh, A. K., & Gallager, R. G., "A Generalized Processor Sharing Approach to flow control in Integrated Services Networks - The Multiple Node Case," IEEE/ACM Transactions on Networking, Vol 2 #1, pp 137-150, April 1994.
- [37] C. S. Chang, "Stability queue length, and delay of deterministic and stochastic queuing networks", IEEE Trans. Automat. Contr., vol. 39, no.5, pp. 913-931, May 1994.
- [38] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *IEEE Journal of Selected Areas in Communication*, special issue on "Advances in the Fundamentals of Networking" (vol. 13 no. 6), August, 1995
- [39] H. Sariowan, R. L. Cruz, and G. C. Polyzos, "Scheduling for Quality of Service Guarantees via Service Curves," Proceedings of the International Conference on Computer Communications and Networks (ICCCN) 1995, Las Vegas, September 20-23, 1995, pp. 512-520.
- [40] C.S. Chang, "On deterministic traffic regulation and service guarantee: A systematic approach by filtering", IEEE TIT vol 44, May 98, pp 913--931.
- [41] R. Agrawal, R. L. Cruz, C. Okino and R. Rajan, " Performance Bounds for Flow Control Protocols", IEEE ToN vol 7, No3, June 99, pp 310--323.
- [42] J.-Y. Le Boudec, "Application of network calculus to guaranteed service networks," IEEE Transactions on Information Theory, vol. 44, pp. 1087--1096, May 1998.
- [43] J.-Y. Le Boudec and P. Thiran, Network Calculus, Springer Verlag Lecture Notes in Computer Science volume 2050.
- [44] Cheng-Shang Chang, Performance guarantees in communication networks, SpringerVerlag, 2000.
- [45] Thomas Bäck, Ulrich Hammel, Hans-Paul Schwefel: Evolutionary computation: comments on the history and current state. IEEE Trans. Evolutionary Computation 1(1): 3-17 (1997)
- [46] H. J. Bremermann, "Optimization through evolution and recombination," in *Self-Organizing Systems*, M. C. Yovits *et al.*, Eds. Washington, DC: Spartan, 1962.
- [47] R. M. Friedberg, "A learning machine: Part I," *IBM J.*, vol. 2, no. 1, pp. 2-13, Jan. 1958.
- [48] R. M. Friedberg, B. Dunham, and J. H. North, "A learning machine: Part II," *IBM J.*, vol. 3, no. 7, pp. 282-287, July 1959.
- [49] G. E. P. Box, "Evolutionary operation: A method for increasing industrial productivity," *Appl. Statistics*, vol. VI, no. 2, pp. 81-101, 1957.
- [50] J. H. Holland, "Outline for a logical theory of adaptive systems," *J. Assoc. Comput. Mach.*, vol. 3, pp. 297-314, 1962.
- [51] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.
- [52] J. H. Holland and J. S. Reitman, "Cognitive systems based on adaptive algorithms," in *Pattern-Directed Inference Systems*, D. A. Waterman and F. Hayes-Roth, Eds. New York: Academic, 1978.
- [53] L. J. Fogel, "Autonomous automata," *Ind. Res.*, vol. 4, pp. 14-19, 1962.
- [54] L. J. Fogel, "On the organization of intellect," Ph.D. dissertation, University of California, Los Angeles, 1964.
- [55] G. H. Burgin, "On playing two-person zero-sum games against nonminimax players," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-5, no. 4, pp. 369-370, Oct. 1969.
- [56] G. H. Burgin, "Systems identification by quasilinearization and evolutionary programming," *J. Cybern.*, vol. 3, no. 2, pp. 56-75, 1973.
- [57] J. W. Atmar, "Speculation on the evolution of intelligence and its possible realization in machine form," Ph.D. dissertation, New Mexico State Univ., Las Cruces, 1976.

- [58] I. Rechenberg, *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart, Germany: Frommann-Holzboog, 1973.
- [59] I. Rechenberg, *Evolutionstrategie '94*, in *Werkstatt Bionik und Evolutionstechnik*. Stuttgart, Germany: Frommann-Holzboog, 1994, vol. 1.
- [60] H.-P. Schwefel, *Evolutionstrategie und numerische Optimierung* Dissertation, Technische Universität Berlin, Germany, May 1975.
- [61] H.-P. Schwefel, *Evolution and Optimum Seeking*. New York: Wiley, 1995 (Sixth-Generation Computer Technology Series).
- [62] T. Back, "Optimal mutation rates in genetic search," in *Proc. 5th Int. Conf. on Genetic Algorithms*, S. Forrest, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 2–8.
- [63] T. C. Fogarty, "Varying the probability of mutation in the genetic algorithm," in *Proc. 3rd Int. Conf. on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann, 1989, pp. 104–109.
- [64] H.-P. Schwefel, *Numerical Optimization of Computer Models*. Chichester: Wiley, 1981.
- [65] H.-G. Beyer, "Toward a theory of evolution strategies: Self-adaptation," *Evolutionary Computation*, vol. 3, no. 3, pp. 311–348, 1995.
- [66] A. E. Eiben, P.-E. Rauée, and Zs. Ruttkay, "Genetic algorithms with multi-parent recombination," in *Parallel Problem Solving from Nature—PPSN III, Int. Conf. on Evolutionary Computation*. Berlin: Springer, 1994, vol. 866 of *Lecture Notes in Computer Science*, pp.78–87.
- [67] [A. E. Eiben, C. H. M. van Kemenade, and J. N. Kok, "Orgy in the computer: Multi-parent reproduction in genetic algorithms," in *Advances in Artificial Life. 3rd Int. Conf. on Artificial Life*, F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, Eds. Berlin: Springer, 1995, vol. 929 of *Lecture Notes in Artificial Intelligence*, pp. 934–945.
- [68] H.-G. Beyer, "Toward a theory of evolution strategies: On the benefits of sex—the (=<sub>;</sub> <sub>)</sub>-theory," *Evolutionary Computation*, vol. 3, no. 1, pp. 81–111, 1995.
- [69] T. Back, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. New York: Oxford Univ. Press and Institute of Physics, 1997.
- [70] E. Zitzler, M. Laumanns, S. Bleuler: A Tutorial on Evolutionary Multiobjective Optimization. Workshop on Multiple Objective Metaheuristics (MOMH 2002), Springer-Verlag, Berlin, Germany, 2003.
- [71] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, vol.3, no. 4, pp.257-271, 1999.
- [72] E. Zitzler, K. Deb, and L. Thiele. *Comparison of multiobjective evolutionary algorithms: Empirical results*. *Evolutionary Computation*, 8(2):173--195, 2000.
- [73] S. Bleuler, M. Laumanns, L. Thiele, E. Zitzler: PISA - A Platform and Programming Language Independent Interface for Search Algorithms. Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), 4/2003. pages 494-508.
- [74] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design, Optimisation, and Control*, pages 19--26, Barcelona, Spain, 2002. CIMNE.
- [75] H. Zhang, "Providing End-to-End Performance Guarantees Using Non-Work-Conserving Disciplines", *Computer Communications: Special Issue on System Support for Multimedia Computing*, volume 18, 10 October 1995.
- [76] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks", *Proc. IEEE*, vol. 84, pp. 1374-1396, Oct. 1996
- [77] H. Zhang and D. Ferrari, "Rate-Controlled-Static-Priority Queuing", In *Proceedings of IEEE*, 1993.
- [78] H. Zhang and E. W. Knightly, "RCSP and Stop-and-Go : A Comparison of Two Non-Work-Conserving Disciplines For Supporting Multimedia Communication", *ACM/Springer-Verlag Multimedia Systems Journal*
- [79] J.Y. Le Boudec and P. Thiran, "Network Calculus, a Theory of Deterministic Queuing Systems for the Internet".
- [80] Christian Fraboul, Fabrice Frances, Jérôme Grieu, « Réseaux Embarqués Aéronautiques », rapport technique, nov. 2001.
- [81] J.L. Faure, X. Favory, O. Gallot, « Influence de l'architecture physique sur la latence de bout en bout d'un réseau Ethernet commuté embarqué. », rapport de projet long, ENSEEIHT, 2004.
- [82] L. Gillet, L. Oukaci, « Utilisation de politiques de services non conservatives dans un réseau embarqué avionique. »
- [83] C.S. Chang, W. Song, and Y. Chiu, "On the performance of multiplexing independent regulated inputs", in *Proc. of Sigmetrics 2001*, Massachusetts.
- [84] M. Vojnović and J.Y. Le Boudec, "bounds for independent regulated inputs multiplexed in a service curve element", in *Proc. of Globecom 2001*, San Antonio, Texas.
- [85] M. Vojnović and J.Y. Le Boudec, "Stochastic Analysis of some Expedited Forwarding Networks", Tech. report DSC200139, EPFL-DSC, [http://dscwww.epfl.ch/EN/publications/documents/tr01\\_39.pdf](http://dscwww.epfl.ch/EN/publications/documents/tr01_39.pdf), July 2001.

- [86] V. Firoiu, J.-Y. Le Boudec, D. Towsley, and Z.-L. Zhang. Theories and models for internet quality of service. Proceedings of the IEEE, 2002.
- [87] F. Baccelli and P. Brémaud, "Elements of queuing theory", vol 26, Applications of Mathematics, Springer-Verlag, 1991.
- [88] W. Hoeffding, "Probability inequalities for sum of bounded random variables", American statistical Association journal, pp. 13-130, March 1963.
- [89] Kesidis and Konstantopoulos, "Worst-case performance of a buffer with independent shaped arrival process", IEEE communication letters, 2000.
- [90] Kesidis and Konstantopoulos, "Extremal traffic and worst-case performance for queues with shaped arrivals," Fields Institute Communications/AMS, ISBN 0-8218-1991-7, 2000.
- [91] Elwalid, Mitra and Wentworth, "A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an atm node", IEEE Journal on selected areas in communications, vol. 13, n°6, pp. 1115-1127, 1995.
- [92] Lo Presti, Zhang, Towsley, and Kurose, "Source time scale optimal buffer/bandwidth trade off for regulated traffic in an atm node", IEEE Trans. on Networking, vol. 7, no 4, pp.490-501, August 1999.
- [93] Massoulié and Busson, "Stochastic Majorization of aggregates of leaky bucket-constrained traffic streams", preprint, <http://www.research.microsoft.com/users/lmassoul/2000>.
- [94] Rajagopal, Reisslein, and Ross, "Packet multiplexers with adversarial regulated traffic", in Proc. of IEEE INFOCOM 1998, pp. 347-355, 1998.
- [95] Kumaran and Mandjes, "Multiplexing regulated traffic streams: design and performance", in Proc. of IEEE INFOCOM 2001, March 2001.
- [96] T. Bonald, A. Proutière, J. Roberts, "Statistical performance guarantees for streaming flows using expedited forwarding" in Proc. of IEEE INFOCOM'2001, March 2001.
- [97] Boorstyn, Burchard, Liebeherr and Ootamakorn, "Statistical service assurances for traffic scheduling algorithms", in IEEE Journal on selected areas in communications, vol. 18, no 12, pp. 2651-2664, December 2000.
- [98] J. Liebeherr, S. D. Patek, and A. Burchard. A calculus for end-to-end statistical service guarantees. Technical Report CS-01-19, University of Virginia, Computer Science Department, August 2001.
- [99] A. Charny, F. Baker, J. Bennett, K. Benson, J.-Y. Le Boudec, A. Chiu, W. Courtney, B. Davie, S. Davari, V. Firoiu, C. Kalmanek, K. K. Ramakrishnan, and D. Stiliadis, "Ef phb redefined," November 2000.
- [100] Davie, B., Charny, A., Baker, F., Bennett, J.C.R., Benson, K., Le Boudec, J., Chiu, A., Courtney, W., Davari, S., Firoiu, V., Kalmanek, C., Ramakrishnam, K. and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC [3246<sup>prop</sup>](#), March 2002.
- [101] J. P. Schmidt, A. Siegel, A. Srinivasan. Chernoff-Hoeffding Bounds for Limited Independence. SIAM J. on Discrete Mathematics 8 (2), 1995, 223-250.
- [102] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. Internet Draft, IETF Diffserv Working Group, August 1998.
- [103] C.-M. Chuang and J.-F. Chang, "Deterministic Loss Ratio Quality of Service Guarantees for High Speed Networks", IEEE Com. Letters, vol. 4, pp. 236--238, July 2000.
- [104] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks," Proceedings of the IEEE, vol. 83, no. 10, Oct. 1995, pp. 1374--1396.
- [105] D. Anick, D. Mitra, and M. Sondhi. Stochastic theory of a data-handling system and multiple sources, Bell Systems Technical Journal, 1982, 61, pp. 1871-1894. 32
- [106] A. Lombardo, G. Morabito, G. Schembra, "An Accurate and Treatable Markov Model of MPEG-Video Traffic", IEEE Proc. Infocom '98, San Francisco, USA, April 1998.
- [107] I. Norros, "On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks," IEEE Journal on Selected Areas in Communications, vol. 13, pp.953-962, 1994.
- [108] J. Roberts, U. Mocchi and J. Virtamo (Eds.), Broadband Network Teletra#c, Final Report of Action COST 242, Lecture Notes in Computer Science, 1155, Springer, 1996.
- [109] O. Yaron and M. Sidi, "Calculating Performance Bounds in Communication Networks," In Proceedings of IEEE INFOCOM '93, Vol. 2, pp. 539-545, March 1993.
- [110] Z.-L. Zhang, D. Towsley, and J.F. Kurose, "Statistical network performance guarantees with generalized processor sharing scheduling," IEEE Journal on Selected Areas in Communications (JSAC), August 1995.
- [111] D. Starobinski and M. Sidi, Stochastically bounded burstiness for communication networks, IEEE Trans. on Information Theory 46 (2000), no. 1, 206--212.



---

## Résumé

Les nouvelles générations d'aéronefs embarquent de plus en plus de systèmes avioniques, pour augmenter à la fois la sécurité et le confort des passagers, tout en maintenant une haute rentabilité pour les compagnies aériennes. Ces nouvelles fonctions entraînent une forte hausse des échanges de données, ce qui nécessite plus de débit et de possibilités d'interconnexion. Les bus classiques de communications avioniques ne peuvent répondre à cette nouvelle demande, ce qui a poussé les constructeurs Airbus et Boeing à installer à bord un réseau de communication utilisant la technologie Ethernet commuté.

Le principal apport de cette thèse est le développement d'une méthode de preuve du déterminisme de ce type de réseaux, démonstration fondamentale pour permettre leur certification aéronautique. Cette méthode, utilisant la théorie du Network Calculus, a été retenue par Airbus pour la certification du réseau AFDX de l'A380.

Nous proposons également des optimisations de ce réseau (politiques de service des commutateurs, interconnexion des éléments réseau) en vue d'améliorer les résultats de son analyse de déterminisme par la méthode précédente. Enfin, nous proposons une extension de nature stochastique à notre méthode déterministe, qui pourrait dans le cadre d'un futur programme permettre de dimensionner au plus juste les ressources physiques du réseau.

---

## Title

STUDY AND EVALUATION OF THE SWITCHED ETHERNET TECHNOLOGY FOR AVIONIC SYSTEMS INTERCONNECTION

---

## Abstract

The recent aircrafts need to accommodate more passengers or freight, with increasing safety and comfort conditions. The new embedded systems imply a large burst in the number and the volume of exchanged data. The avionic data buses cannot cope anymore with these new communications needs. Both Airbus and Boeing made the choice to replace these buses with a network using the Switched Ethernet technology. One of the key issues of this technology is that it must be demonstrated to the certification authorities that it offers a deterministic behaviour, i.e. bounded latencies and jitter and no frames loss due to switches congestions.

The main contribution of this thesis is a method, based on the Network Calculus theory, that makes it possible to prove this determinism.

Since the network resources will be dimensioned according to the results of the determinism analysis, we also introduce some network optimisations according to this criterion. Finally, we propose a stochastic extension to the determinism analysis that gives a better understanding of the previously determined deterministic bounds, which could lead to a new certification process for future programs.

---

## Discipline

RESEAUX ET TELECOMMUNICATIONS

---

## Mots-clés

Réseau embarqué, Ethernet commuté, preuve de déterminisme, Network Calculus, bornes de latence, analyse de performances, dimensionnement de réseau.

---

## Intitulé et adresse du laboratoire:

TéSA, 2 rue Charles Camichel F-31071 Toulouse, Cédex 7, BP 7122, FRANCE.