Systèmes contrôlés en réseau : Evaluation de performances d'architectures Ethernet commutées

Jean-Philippe Georges

CENTRE DE RECHERCHE
EN AUTOMATIQUE DE
NANCY

FACULTE DES SCIENCES & TECHNIQUES

U.F.R. : Sciences et Techniques Mathématiques, Informatique et Automatique

Ecole Doctorale : IAEM Lorraine

Département de Formation Doctorale : Automatique

CNRS UMR 7039

Thèse

présentée pour l'obtention du titre de

Docteur de l'Université Henri Poincaré, Nancy 1

en Sciences, spécialité Automatique,
Traitement du Signal et Génie Informatique

par **Jean-Philippe Georges**

Systèmes contrôlés en réseau : Evaluation de performances d'architectures Ethernet commutées

Soutenue publiquement le 4 novembre 2005

Membres du jury :

Présidente :	Professeur Sirkka-Liisa Jämsä-Jounela	Université de Technologie d'Helsinki, Finlande
Rapporteurs :	Professeur Jean-Louis Boimond	Université d'Angers, LISA
	Professeur Jean-Marc Faure	ENS Cachan, LURPA
	Professeur Christian Fraboul	Institut National Polytechnique de Toulouse, IRIT
Directeurs de thèse :	Professeur Thierry Divoux	Université Henri Poincaré Nancy 1, CRAN
	Professeur Eric Rondeau	Université Henri Poincaré Nancy 1, CRAN

Université Henri Poincaré, Centre de Recherche en Automatique de Nancy

Faculté des Sciences, B.P. 239, 54506 Vandœuvre lès Nancy

CENTRE DE RECHERCHE
EN AUTOMATIQUE DE
NANCY

FACULTE DES SCIENCES & TECHNIQUES

U.F.R. : Sciences et Techniques Mathématiques, Informatique et Automatique

Ecole Doctorale : IAEM Lorraine

Département de Formation Doctorale : Automatique

CNRS UMR 7039

Thèse

présentée pour l'obtention du titre de

Docteur de l'Université Henri Poincaré, Nancy 1

en Sciences, spécialité Automatique,
Traitement du Signal et Génie Informatique

par **Jean-Philippe Georges**

Systèmes contrôlés en réseau : Evaluation de performances d'architectures Ethernet commutées

Soutenue publiquement le 4 novembre 2005

Membres du jury :

Présidente :	Professeur Sirkka-Liisa Jämsä-Jounela	Université de Technologie d'Helsinki, Finlande
Rapporteurs :	Professeur Jean-Louis Boimond	Université d'Angers, LISA
	Professeur Jean-Marc Faure	ENS Cachan, LURPA
	Professeur Christian Fraboul	Institut National Polytechnique de Toulouse, IRIT
Directeurs de thèse :	Professeur Thierry Divoux	Université Henri Poincaré Nancy 1, CRAN
	Professeur Eric Rondeau	Université Henri Poincaré Nancy 1, CRAN

Université Henri Poincaré, Centre de Recherche en Automatique de Nancy

Faculté des Sciences, B.P. 239, 54506 Vandœuvre lès Nancy

Remerciements

Mes remerciements les plus sincères iront, en premier lieu, à mes directeurs de thèse, les Professeurs Thierry Divoux et Eric Rondeau. Plus qu'un encadrement constructif de mon travail de thèse, ils m'ont initié aux différents aspects et à la convivialité de la recherche. Les résultats présentés dans ce document sont le fruit de leurs conseils et de leurs critiques qu'ils ont su me prodiguer en toute circonstance, que ce soit au BSR ou en déplacement.

Etudier dans un laboratoire de recherche où l'accueil est attentif et convivial, est très important à la réussite de la thèse. Je tiens ainsi à remercier le directeur du Centre de Recherche en Automatique de Nancy, le Professeur Alain Richard, des efforts du laboratoire quant à l'accompagnement et ma formation ainsi que le Professeur Francis Lepage de m'avoir permis de me lancer dans une thèse. La thèse, c'est une aventure, et pour quelqu'un de distrait de la paperasserie comme moi, cette aventure ne ce serait pas si bien conclue sans l'appui amical d'Anne et Christelle que je remercie.

Rencontrer et collaborer avec différentes personnes a été pour moi complètement instructif. Pour leur aide, leur soutien et nos longues appartés, je remercie Thierry et Eric ainsi que Nicolas Krommenacker. Je voudrais exprimer ma gratitude aux membres du groupe thématique Systèmes de Production Ambiants, et plus particulièrement aux membres de l'équipe projet Systèmes Contrôlés en Réseau. Mes travaux de recherche ont été significativement enrichi grâce à eux. Qu'ils soient jeunes ou futurs docteurs, je voudrais mentionner plus particulièrement Ahmed Aïtali, Mickael David, Thomas Holl et Fabien Michaut.

Cette thèse s'est également déroulée dans sa dernière partie dans le cadre du projet FP6 IST STREP intitulé *Networked Control Systems Tolerant to faults* financé par la comunauté européenne. Je remercie à ce titre les Professeurs Christophe Aubrun et Dominique Sauter du temps qu'ils m'ont consacré à mon appréhension d'autres systèmes et théories.

Il ne peut y avoir d'enrichissiment d'un travail que par la critique. Aussi, je remercie les Professeurs Jean-Louis Boimond, Jean-Marc Faure et Christian Fraboul pour avoir accepté de rapporter sur ma thèse et pour leurs suggestions et remarques pertinentes qui m'on permis de terminer ce travail. Je remercie également la Professeur Sirrka-Liisa Jämsä-Jounela de l'Université de Technologie d'Helsinki d'avoir accepté de participer au jury et d'avoir fait un si long déplacement depuis Helsinki en Finlande. Je la remercie d'autant plus de m'accueillir bientôt au sein de son équipe et de me permettre ainsi de continuer mes travaux sur les systèmes contrôlés en réseau dans le cadre d'un Post Doctorat.

Enfin, j'adresse à mes parents et à toute ma famille tous mes remerciements pour m'avoir toujours soutenu et encouragé depuis mes premières années d'étude. Ils m'ont toujours permis d'atteindre la valeur maximale...

This research has been conducted as a part of the Networked Control Systems Tolerant to Faults (NeCST) project IST-004303 that is partially funded by the EU. The project covers closely both main topics of NCS, control of networks, and control over networks. The aim of the project is to explore research opportunities in the direction of distributed control systems in order to enhance the performance of diagnostics and fault tolerant control systems. The systems under consideration in the framework of this project can be considered as a distributed network of nodes operating under highly decentralised control, but unified in accomplishing complex system-wide goals. One of the key factors in designing such a complex system is that both the physical subsystem and the control part have to be designed together in an integrated manner. The aim is to build up a software platform integrating the FDI/FTC strategies and methodology of NCS and embedded systems. Control strategies will first be tested with experimental simulation and pilot-scale platforms, and finally in the industrial environment of the oil refinery. The authors gratefully acknowledge the support.

http ://www.strep-necst.org

A mes parents

A mon frère & mes soeurs

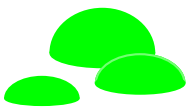
Aux belles montagnes. . .  *, les Vosges*

Table des matières

Introduction **xix**

I Positionnement 1

1 Ethernet pour les systèmes contrôlés en réseau **3**

1.1 Systèmes contrôlés en réseau (SCR) . . . 3
1.1.1 Introduction . . . 3
1.1.2 Influence du réseau sur le contrôle du système . . . 5
1.1.3 Présentation générale des travaux de recherche . . . 7
1.1.4 Contrôle & stabilité . . . 9
1.1.5 Vers une vision intégrée . . . 10
1.1.6 Simulations . . . 13
1.1.7 Derniers développements . . . 13
1.2 Des réseaux déterministes sur mesure 14
1.3 ... aux réseaux Ethernet . . . 15
1.3.1 Contexte . . . 15
1.3.2 Présentation . . . 16
1.3.3 Disparité des délais sur Ethernet . . . 17
1.3.4 Structuration actuelle . . . 19
1.4 Délimitation de notre travail . . . 20

2 Améliorer les performances d'Ethernet **23**

2.1 Ethernet à travers les âges (du linéaire à l'étoile) . . . 23
2.2 Propositions pour l'Ethernet partagé . . . 27
2.2.1 Modifications de l'accès à la voie . . . 27
2.2.2 Algorithmes déterministes pour la gestion des collisions sous CSMA . . . 28
2.2.3 Prioritisation de l'accès . . . 29
2.2.4 Contrôle d'admission . . . 29
2.3 Propositions pour l'Ethernet commuté . . . 30
2.3.1 Problème de la diffusion multicast . . . 30
2.3.2 Le service offert aux messages à temps-critique . . . 31
2.3.3 Synchronisation des horloges . . . 36

2.3.4 Optimisation de la topologie . . . 38
2.3.5 Evaluation de performances . . . 40
2.4 Analyse des propositions . . . 41
2.5 Conclusion . . . 43

II Contributions 45

3 Modélisation du réseau et du trafic 47
3.1 Le calcul réseau comme support d'analyse . . . 47
3.1.1 Introduction . . . 47
3.1.2 Concepts . . . 48
3.1.3 Définitions et propriétés établies par Cruz . . . 49
3.1.4 Fondements du calcul réseau . . . 51
3.1.5 Exploitation . . . 52
3.2 Modélisation de la commutation : identification d'une courbe de service . . . 53
3.2.1 *802.1D*, ou qu'est ce qu'un commutateur ? . . . 53
3.2.2 Technologies 54
3.2.3 Modèles de commutateurs . . . 58
3.2.4 Modélisation d'un noeud de commutation par R. Cruz . . . 59
3.2.5 Nos propositions (Georges *et al.*, 2003*b*),(Georges *et al.*, 2003*c*) . . . 60
3.2.6 Conclusion : courbe de service et majorant du délai . . . 61
3.3 Modélisation du trafic : identification d'une courbe d'arrivée basée sur le volume des rafales 62
3.3.1 Généralités . . . 62
3.3.2 Spécificités des systèmes contrôlés en réseau . . . 64
3.3.3 Notion de rafale, ou avalanche de données . . . 65
3.4 Etude des délais de composants FIFO . . . 66
3.4.1 Introduction . . . 66
3.4.2 Système à file d'attente . . . 67
3.4.3 Multiplexage . . . 69
3.4.4 Démultiplexage . . . 74
3.5 Conclusion . . . 74

4 Détermination des délais maxima de bout en bout 77
4.1 Méthodologie générale . . . 77
4.1.1 Courbe de départ $\alpha^*(t)$ . . . 77
4.1.2 Inter-dépendances des avalanches de données de bout en bout . . . 80
4.1.3 Calcul des délais de bout en bout . . . 81
4.2 Comparaison des modèles (Georges *et al.*, 2003*c*),(Georges *et al.*, 2003*b*) . . . 83
4.2.1 Scénario de communication . . . 84
4.2.2 Résultats . . . 85
4.2.3 Conclusion . . . 85
4.3 La classification de service . . . 86

4.3.1 Etude de la standardisation . . . 86
4.3.2 Modélisation du commutateur à priorité . . . 88
4.3.3 Ordonnancement à priorité stricte (SP) . . . 89
4.3.4 Ordonnancement WFQ . . . 91
4.3.5 Majorant des délais pour SP et WFQ . . . 94
4.3.6 Conclusion . . . 95
4.4 Validation expérimentale . . . 95
4.5 Conclusion . . . 100

5 Application aux systèmes contrôlés en réseau (Ethernet commuté) 103
5.1 Dimensionnement d'architectures pour les systèmes contrôlés en réseau (Georges *et al.*, 2002) 103
5.1.1 Passage à l'échelle du réseau et influence de la fragmentation . . . 104
5.1.2 Capacité d'accueil de trafic non-critique . . . 106
5.1.3 Comparaison des topologies . . . 107
5.2 Efficacité du plan de câblage d'architectures commutées pour des applications à temps-réel 108
5.2.1 Problème de partitionnement . . . 109
5.2.2 Fonction objectif . . . 110
5.2.3 Application . . . 110
5.3 Configuration de la CdS par rapport aux niveaux de QdS exigées par l'application . . . 112
5.4 Conclusion . . . 115

Conclusions 117

Majorants du Network Calculus (Le Boudec et Thiran, 2001) 121

Liste des publications 125

Notice bibliographique 127

Table des figures

1.1 Vue traditionnelle des systèmes contrôlés en réseau (Zhang *et al.*, 2001) 4
1.2 Eléments d'un système d'enroulement de bande 5
1.3 Architecture de simulation / émulation 6
1.4 Dépassement en sortie d'un SCR dû aux délais introduit par le réseau (*la grandeur en abscisse est le temps en s*). 6
1.5 Modèle d'asservissement distribué (Juanole et Blum, 1999) 8
1.6 Modèle d'asservissement distribué (Nilson, 1998) 9
1.7 Modèle d'asservissement distribué à multiples entrées et sorties (Nilson, 1998) 9
1.8 Réseau de Petri d'un système distribué temps-réel (Juanole et Blum, 1999) 11
1.9 Réseau de Petri d'un échange maître / esclave 12
1.10 Apparition d'une collision avec CSMA/CD 17
1.11 Délais sur Ethernet 18
1.12 The International Fieldbus - IEC 61158 19
1.13 Schéma d'étude du projet européen NeCST 20
1.14 Contexte d'étude 21

2.1 Comparaison des zones de collision d'architectures Ethernet 24
2.2 Evolution de la normalisation 'Ethernet' 26
2.3 Principe de CSMA/DCR 28
2.4 Isolation des équipements par la technologie VLAN 31
2.5 La couche temps-réel 32
2.6 Fomat de la trame Ethernet étiquettée 34
2.7 Instanciation d'une file d'attente pour chaque classe de service au niveau d'un port de sortie d'un commutateur 35
2.8 Transactions liées au processus de synchronisation de PTP 37
2.9 La charge des commutateurs dépend de la distribution des équipements de contrôle sur les commutateurs 39
2.10 L'architecture du réseau étudiée est de nature commutée full-duplex micro-segmentée . . . 43

3.1 Interprétation graphique de l'arriéré de traitement 51
3.2 Délais et arriéré : distances horizontales et verticales entre les courbes d'arrivée et de service 52
3.3 Relais d'une trame par un pont *802.1D* (IEEE, 1998, figure 7-4) 53
3.4 Modèle de bufferisation 55

3.5 Implémentation de plusieurs files sur chaque port . . . 56
3.6 Implémentations de commutation . . . 56
3.7 Fabriques de commutation . . . 57
3.8 Modèles de commutateur . . . 59
3.9 Modèle n°1 de commutation proposé par (Cruz, 1991*b*) . . . 60
3.10 Vers un modèle de commutateur 802.1D . . . 61
3.11 Courbe d'arrivée en escalier . . . 62
3.12 Courbes d'arrivée possibles pour un flux périodique . . . 63
3.13 Le concept de seau percé . . . 64
3.14 Composants élémentaires du réseau . . . 66
3.15 Majoration du délai dans une file d'attente . . . 68
3.16 Arriéré de traitement d'un multiplexeur FIFO à 2 entrées . . . 69
3.17 Paramètres d'entrée et de sortie du multiplexeur. . . . 70
3.18 Vue modélisée du système contrôlé en réseau . . . 76

4.1 Evolution de la contrainte d'avalanche de données tout au long d'un réseau Ethernet commuté. . . . 78
4.2 Confluences croisées des flux . . . 80
4.3 L'outil de simulation *Comnet III* . . . 83
4.4 Modélisation d'un commutateur dans *Comnet* . . . 84
4.5 Réseau Ethernet commuté d'étude . . . 84
4.6 Evaluation des modèles . . . 85
4.7 Relais d'une trame par un pont *802.1D/p* (IEEE Computer Society, 2003, figure 8-4). . . . 86
4.8 Le mécanisme de prioritisation modifie la mise en attente au niveau des ports de sortie . . . 88
4.9 Courbe de service & arriéré de traitement pour un ordonnancement statique à priorité stricte. . . . 90
4.10 Contexte de retardement du service offert à un paquet sous PGPS par rapport à GPS. . . . 91
4.11 Courbe de service *weighted round robin.* . . . 93
4.12 Première plateforme expérimentale. . . . 96
4.13 Mesures des délais de bout en bout des trames émises par garros (flux 1) en μs. . . . 98
4.14 Une seconde plateforme expérimentale. . . . 98
4.15 Seconde série de mesures des délais de bout en bout des trames émises par garros (flux 1). 99
4.16 Majoration des délais de traversée de bout en bout pour chaque flux . . . 101

5.1 Capture d'écran de l'outil Cerise . . . 104
5.2 Réseau d'étude à topologie hiérarchique . . . 104
5.3 Evolution de l'acceptation de charge du réseau de la figure 5.2 . . . 106
5.4 Influence du trafic apériodique sur le réseau de la figure 5.2 . . . 106
5.5 Réseau d'étude à topologie linéaire . . . 107
5.6 Evolution de l'acceptation de charge du réseau de la figure 5.5 . . . 107
5.7 Influence du trafic apériodique sur le réseau de la figure 5.5 . . . 108
5.8 Etapes de la conception d'une architecture Ethernet commutée . . . 109
5.9 Représentation d'un réseau Ethernet commuté par un graphe . . . 109

5.10 Cas d'étude de la méthode d'optimisation de la topologie. 111
5.11 Cas d'étude 112

Introduction

Les impératifs de réduction de coût, de simplicité, de flexibilité et d'évolutivité ont concouru ces dernières années à l'éclatement des systèmes centraux de contrôle / commande d'applications industrielles en plusieurs *sous* systèmes qu'il faut coordonner. Ces différents nœuds de contrôle, de mesure ou encore de supervision sont alors reliés par des *réseaux locaux industriels* qui sont eux-mêmes de plus en plus interconnectés au réseau d'entreprise et désormais au réseau Internet. Ces réseaux se sont naturellement développés en parallèle des produits et des solutions offertes par les constructeurs ; chacun fournissant ses propres produits (protocoles de communication et interfaces réseau pour automates, capteurs ...). Il s'ensuit que ces réseaux sont caractérisés par une forte hétérogénéité de leurs composants logiciels et matériels.

Toutefois, bien que la distribution de l'asservissement d'une application offre de nombreux avantages (Gallara et Decotignie, 1984), plusieurs précautions sont nécessaires. Ainsi, les réseaux de communication introduisent inévitablement des retards et des pertes qui sont notamment générés par les limites de la capacité du canal de communication ou de la surcharge des équipements du réseau et des nœuds de contrôle. La vérification des performances du système global (incluant le réseau) est alors la problématique essentielle de ces systèmes générant un nouvel axe de recherche appelé *systèmes contrôlés en réseau*[1]. La dynamique de ces applications impose que le contrôle soit temps-réel (c'est-à-dire que le temps de réaction soit borné en fonction des contraintes applicatives). Par conséquent, l'architecture de communication retenue doit également satisfaire cette contrainte temporelle.

L'une des caractéristiques fondamentales des délais introduits par les réseaux est qu'ils varient de manière relativement imprévisible du fait de la complexité des architectures et de situations liées au hasard (distribution spatio-temporelle imprévisible des occurrences des besoins en communication). Le comportement du réseau est également influencé par les protocoles de communication et la topologie utilisée. Le réseau auquel sont connectés les nœuds de contrôle peut également être surchargé par un trafic supplémentaire non lié à l'asservissement du système distribué. Des applications comme la télémaintenance ou la lecture de documentation à distance peuvent alors venir perturber le bon fonctionnement de l'application de contrôle / commande.

Dans ce contexte, les premiers réseaux développés pour les environnements industriels comme les réseaux de terrain FIP, PROFIBUS ... se sont attachés à satisfaire les contraintes temporelles d'applications maîtrisées. Toutefois, le caractère propriétaire des solutions offertes par les constructeurs n'a pas permis de pleinement répondre aux attentes de réduction des coûts et d'interopérabilité des équipements. De plus, l'environnement protecteur de ces architectures limite l'évolutivité et la flexibilité du réseau. Bien

[1] appellation notamment définie par l'IFAC Technical Committee 1.5 Networked Systems

qu'il ne soit pas dédié à ce contexte de communications temps-réel, plusieurs organisations, constructeurs et travaux de recherche se sont alors consacrées à l'utilisation du réseau Ethernet pour les systèmes contrôlés en réseau, car il présente la souplesse nécessaire à la satisfaction de ces besoins.

Dans la mesure où le réseau Ethernet s'appuie sur un mécanisme d'accès à la voie indéterministe et n'offre par conséquent pas de garantie concernant les délais d'acheminement des trames, une étude de ce type de réseau est nécessaire pour justifier l'opportunité de son utilisation dans le cadre d'applications temps-réel distribuées. L'objectif qui sera développé dans la suite de ce document sera alors axé sur la satisfaction des contraintes temporelles sur un réseau Ethernet. Le résultat majeur des travaux que nous allons présenter est une méthode d'analyse de majorants des délais de traversée de bout en bout d'une architecture Ethernet. L'utilisation de cette méthode conduira par la suite à comparer les performances du réseau aux exigences de l'application. Plus particulièrement, cela permettra d'identifier le service minimal offert par le réseau et d'intégrer cette information à la loi de commande de l'application.

La méthode de détermination de majorant du délai de traversée s'appuie sur une théorie d'évaluation des réseaux appelé calcul réseau, ou *network calculus*. L'utilisation de cette théorie s'appuie sur différentes étapes : modélisation de l'architecture de communication, identification d'une courbe d'arrivée des données (ou modélisation du trafic), méthode de formulation des délais de bout en bout. Les résultats obtenus sont également vérifiés au moyen d'outils de simulation et d'expérimentations. Enfin, les mécanismes d'accommodation du service comme la classification de service sont pris en compte dans l'analyse en vue d'évaluer l'amélioration des performances du réseau Ethernet en regard des contraintes temporelles de l'application. Toutes ces étapes obéissent à l'enchaînement suivant.

Organisation du document

Ce document est organisé en deux parties. La première partie présente le contexte et le positionnement de nos travaux par rapport à la bibliographie. Elle regroupe les deux premiers chapitres. La seconde partie couvre les trois autres chapitres qui détaillent les contributions apportées dans cette thèse.

Le premier chapitre vise à identifier le contexte de nos travaux. Le concept de systèmes contrôlés en réseau y est présenté. Cette présentation permet de mettre en évidence l'importance des délais introduits par le réseau dans la performance du contrôle du système distribué, notamment en terme de stabilité de la commande. Une étude des différents travaux menés dans le cadre des systèmes contrôlés en réseau montre que la caractérisation de la Qualité de Service offerte par le réseau n'est pas encore suffisante notamment en terme de déterminisme. Parallèlement, ce chapitre développe les motivations qui ont conduit à une utilisation de la spécification Ethernet comme support de systèmes contrôlés en réseau. Il expose alors le paradoxe qui veut qu'Ethernet s'appuie sur une méthode indéterministe.

Le deuxième chapitre présente alors un panorama des travaux menés jusqu'ici dans le but d'améliorer les performances d'Ethernet en vue de son utilisation dans les systèmes contrôlés en réseau. Cet état de l'art montre que la plupart des solutions conduit à une extension ou une modification de la standardisation d'Ethernet. Par contre, nous choisirons de ne pas modifier la norme afin de conserver les apports d'interopérabilité et d'évolutivité d'Ethernet. Notre objectif est donc de vérifier *a priori* qu'une architecture Ethernet standard satisfait les contraintes temporelles applicatives dans le pire cas. Il ne s'agit donc pas de rendre le réseau déterministe mais de garantir qu'il offre un service conforme à la contrainte

requise par les systèmes contrôlés en réseau. Dans ce cadre, ce chapitre montre que seule une topologie particulière, Ethernet commuté full-segmenté, permet d'éliminer le problème de l'accès à la voie (les collisions). Toutefois, l'élimination des collisions conduit à une étude de la congestion des files d'attentes du réseau qui nécessite d'être quantifiée.

Les chapitres suivants illustrent ensuite notre contribution qui relève de l'adaptation de la théorie du calcul réseau au contexte des systèmes contrôlés par un réseau Ethernet commuté.

Le troisième chapitre présente ainsi les différents fondements du calcul réseau, et plus particulièrement ceux introduits par René Cruz. Une étude des différentes architectures de communication nous permet alors d'identifier un modèle de commutateur basé sur le standard IEEE 802.1D. Sous l'hypothèse déterministe que l'arrivée des données est contrainte par une courbe d'arrivée dérivée de la contrainte d'avalanche maximale, ce chapitre développe les formules liées au majorant du temps de traversée de chacun des composants élémentaires utilisés dans la modélisation des commutateurs.

L'objectif du quatrième chapitre est l'expression d'une méthode d'analyse des délais maximum de bout en bout. Cette formulation s'appuie sur les expressions identifiées au troisième chapitre ainsi que sur l'idée introduite par Cruz que l'attente liée à la traversée d'un composant peut se traduire dans le pire cas par une augmentation des rafales de données. Dans ce chapitre, le modèle identifié au chapitre précédent est complété afin de supporter la classification de service et diverses séries de mesures expérimentales sont présentées. Ces mesures confirment la validité de la méthode proposée et introduisent les différents *scenarii* d'application détaillés au chapitre suivant.

Le dernier chapitre illustre enfin les différentes utilisations possibles de la méthode de calcul de majorant des délais de bout en bout proposée. Dans un premier temps, cette méthode est appliquée à un scénario de communication particulier afin d'identifier les possibilités d'ajout de trafic supplémentaire au contrôle et de comparer différentes architectures commutées. Dans un second temps, la majoration des délais est utilisée comme critère d'évaluation conjointement à une méthode d'optimisation de la topologie d'un réseau Ethernet commuté. Enfin, un dernier exemple d'application montre comment le calcul de majorants des délais de bout en bout permet d'identifier le gain obtenu lorsque la classification de service est utilisée et comment cette classification de service peut être optimisée.

Première partie

Positionnement

Chapitre 1

Ethernet pour les systèmes contrôlés en réseau

1.1 Systèmes contrôlés en réseau (SCR)

1.1.1 Introduction

Un système contrôlé en réseau (SCR) (ou *networked control systems* dans la littérature anglophone) correspond simplement à un système de contrôle/commande distribué *via* un réseau pouvant être partagé avec d'autres applications non impliquées dans la commande du système. Dans la communauté des systèmes à espace d'état continu (le cadre des systèmes contrôlés en réseau inclut également les systèmes à événements discrets), un SCR est un système de contrôle à asservissement dans lequel les boucles de régulation sont fermées au moyen d'un réseau de communication (Zhang *et al.*, 2001). Cette définition peut alors être approfondie à partir des présentations de la journée du groupe de travail StrQds (Systèmes Temps-Réel et Qualité de Service) du groupe de recherche ARP (Architecture, Réseaux et système, Parallélisme) consacrée aux *Networked Control Systems* (Richard, 2005)(Juanole, 2005).
Le domaine des systèmes contrôlés en réseau (SCR) est relativement nouveau dans la communauté académique du contrôle (nouveau Technical Committee 1.5 de l'IFAC), mais il ne l'est pas dans l'industrie. On les retrouve dans les systèmes de production, les avions, les automobiles. De même, l'implémentation des contrôles distribués n'est pas récente. On peut par exemple citer le Distributed Control System (DCS) de Honeywell dans les années 70. De plus, plusieurs compagnies comme Rockwell Automation, ABB, Siemens, GE proposent des équipements appelés adaptateurs de communication qui permettent aux systèmes de contrôle de communiquer les données du process sur un réseau. Ces interfaces de communication peuvent utiliser différents réseaux de contrôle comme DeviceNet, ControlNet, Profinet, Modbus, Firewire ou TTP : chaque type de réseau possède son propre média physique, mécanisme d'arbitrage de l'accès au bus et une suite de protocoles de couche haute. De plus, chacun de ces réseaux répond à des critères de performances différents tels la vitesse de transfert, la taille des messages, des majorants du délai et la disponibilité.

La caractéristique principale des systèmes contrôlés en réseau est donc que l'information (consigne, sortie du système, commande ...) est échangée en utilisant le réseau entre les composants du système de

contrôle (capteurs, contrôleurs, actionneurs ...). (Zhang *et al.*, 2001) illustrent ainsi un système contrôlé en réseau (figure 1.1).

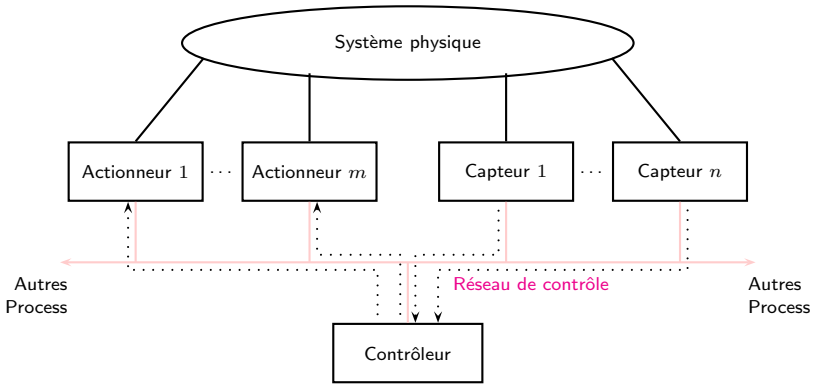


FIG. 1.1: Vue traditionnelle des systèmes contrôlés en réseau (Zhang *et al.*, 2001)

Les avantages des SCR sont la réduction des coûts de câblage (Gallara et Decotignie, 1984), l'aide au diagnostic et à la maintenance des systèmes (Iung, 2002), l'amélioration de la modularité et de la flexibilité dans la conception des systèmes.

L'insertion du réseau de communication dans la boucle de contrôle rend l'analyse et la conception des SCR relativement complexe. Les théories de contrôle conventionnelles qui comportent plusieurs hypothèses idéales telles la synchronisation de la commande et de l'observation ainsi que la capacité de réaction sans retard doivent être réévaluées avant de pouvoir être appliquées aux systèmes contrôlés en réseau. Les points critiques mis en avant par (Zhang *et al.*, 2001) sont :

- le délai induit par le réseau (du capteur vers le contrôleur et du contrôleur vers l'actionneur) lors de l'échange de données entre les équipements connectés au médium partagé. Ce délai, qu'il soit constant ou variable, peut dégrader la performance d'un système de contrôle qui ne le prendrait pas en compte, voire même rendre instable le système.
- le réseau fournit un ensemble de chemins non fiables. Des paquets peuvent être perdus, dupliqués, désordonnés ... ce qui doit être pris en compte.
- l'information peut être contenue dans plusieurs messages. Les chances que tous, une partie ou aucun des paquets arrivent doivent alors également être étudiés.

Les systèmes informatiques distribués (calculateurs interconnectés à travers un réseau de communication qu'il soit local ou réseau de terrain) sont aussi de plus en plus utilisés pour réaliser des applications de contrôle commande et en particulier de type bouclé. Ces systèmes sont des systèmes temps-réel, c'est-à-dire que la maîtrise temporelle du service fourni est essentielle pour garantir les performances des applications. Cette maîtrise temporelle dépend, non seulement, de l'exécution des tâches concernées (début, durée) mais aussi des échanges à travers le réseau (délai de transmission, perte).

Cela couvre alors aussi bien l'ordonnancement et le transfert des messages et des tâches applicatives. Compte tenu de la particularité de cette ressource (délais, gigues, pertes, protocoles), il est nécessaire que les différents équipements prennent notamment en compte les problèmes suivants :

- cohérences spatiales et temporelles de l'information
- production (conditions) et fraîcheur de l'information
- priorité de l'information

– disponibilité de la ressource
– autonomie des sous-systèmes

Toutes ces notions mettent l'accent sur la vérification formelle et l'évaluation de performances des caractéristiques temps-réel de l'application. Dans ce cadre, nous sommes convaincus que les travaux de la communauté des systèmes à événement discrets peuvent apporter des éléments de réponses.

1.1.2 Influence du réseau sur le contrôle du système

Afin d'illustrer l'influence du retard introduit par le réseau de communication dans l'asservissement de l'application, nous présentons maintenant le résultat d'une expérimentation. L'étude porte sur l'analyse d'un procédé d'enroulement - déroulement de bande présent dans les industries papetière ou sidérurgique pour le conditionnement de produits.
Un système d'entraînement de bande peut être divisé en plusieurs sous-systèmes. Parmi ces différents sous systèmes, on distingue un dérouleur et un enrouleur disposés respectivement en début et fin de ligne de traitement. Les autres éléments tels que les moteurs tracteurs et les rouleaux libres sont disposés le long de la chaîne en fonction du procédé de traitement désiré.
Le dérouleur est le point de départ d'une ligne de transport de bande. La bande est entraînée et guidée par des rouleaux qui peuvent être motorisés ou libres. Le schéma du système d'enroulement de bande est présenté à la figure 1.2. Un rouleau central motorisé commande la traction de la bande et lui assure une tension adéquate.

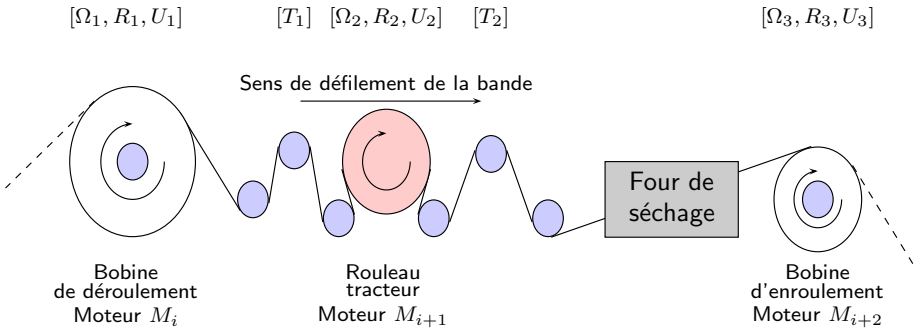


FIG. 1.2: Eléments d'un système d'enroulement de bande

L'étape d'enroulement est une étape délicate du processus de traitement qui conditionne la qualité finale du produit. Les défauts qui peuvent être provoqués par une commande (la tension appliquée aux moteurs continus U_i) mal conditionnée se caractérisent par l'apparition de phénomènes d'ondulations, de bulles d'air voire de déchirures au niveau des bobines. Les états observés correspondent aux trois mesures de la tension de la bande, T_1 la mesure entre R_1 et R_2, T_2 la mesure entre R_2 et R_3 et Ω_2 mesure la vitesse angulaire de R_2.

$$\dot{X}_t = AX_t + BU_t \qquad U = [U_1\ U_2\ U_3]^T$$
$$Y_t = CX_t + DU_t \qquad X = [T_1\ \Omega_2\ T_2]^T$$

L'expérimentation consiste en une modélisation de la commande et du système sous MATLAB/Simulink et à une émulation du réseau (utilisation d'un réseau réel). La plateforme de test est schématisée sur la figure 1.3.

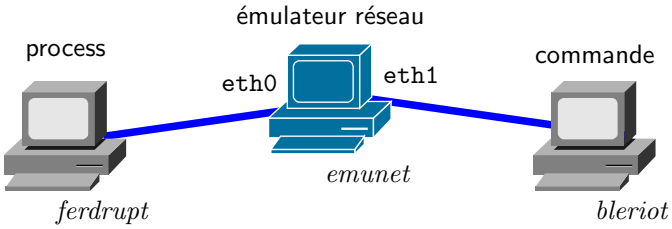
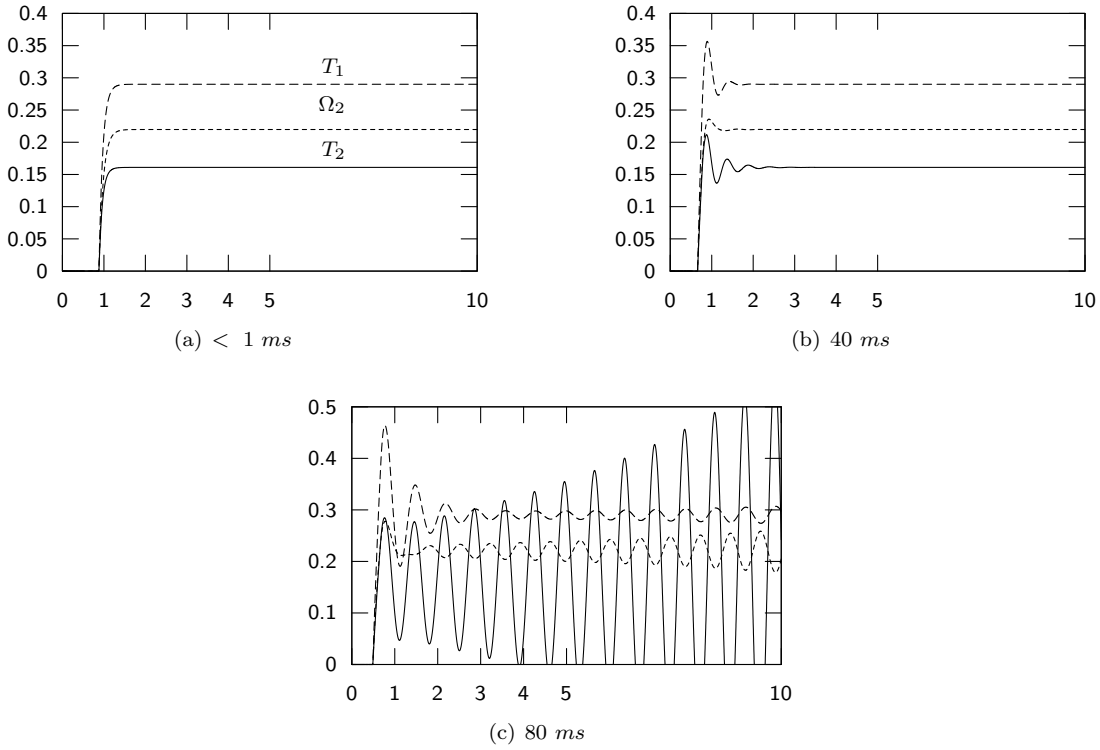


FIG. 1.3: Architecture de simulation / émulation

Deux PCs distincts sont utilisés pour exécuter en temps-réel la simulation de la commande et du système. Les deux PC *ferdrupt* et *bleriot* exécutent respectivement les modèles Simulink du process et de la commande. Ces deux PCs sont interconnectés *via* un réseau Ethernet. Le troisième PC sera utilisé pour émuler les performances du réseau à l'aide de l'émulateur *nistnet*. Les états observés correspondent aux trois mesures (T_1, Ω_2 et T_2). A ce stade, nous présentons ici directement les valeurs obtenues pour trois valeurs de délai. Pour plus d'information, le lecteur pourra se reporter au *deliverable* (Aubrun *et al.*, 2005) ainsi que sur le site du projet européen NeCST[1].



(a) $< 1\ ms$ (b) $40\ ms$

(c) $80\ ms$

FIG. 1.4: Dépassement en sortie d'un SCR dû aux délais introduit par le réseau (*la grandeur en abscisse est le temps en s*).

La figure 1.4 montre bien qu'un retard trop élevé (c'est-à-dire une faible qualité de service du réseau) entraîne une dégradation significative de l'efficacité de la commande. On observe ainsi que si la réponse du système est appropriée lorsque le réseau n'engendre aucun de retard, l'apparition d'un retard provoque

[1] `http ://www.strep-necst.org/private/platforms/simulink`

un dépassement de la consigne (graphe 1.4(b)) et peut dans le cas extrême rendre le système instable (graphe 1.4(c)). *Les délais de transmission sur le réseau dégradent la performance de la dynamique du système et affectent la stabilité du SCR.* Les valeurs de délai sont ici propres à l'application et ne sont pas fondamentales. Par contre, cet exemple montre que l'étude de la dynamique d'un SCR doit reposer sur une analyse poussée de la qualité de service offerte par le réseau au système.

1.1.3 Présentation générale des travaux de recherche

Le domaine de recherche des systèmes contrôlés en réseau est relativement récent. Néanmoins, il existe un certain nombre de travaux sur ce sujet. Le lecteur intéressé pourra consulter les papiers déposés sur le cache Internet (Liberatore *et al.*, 2004).

Les mécanismes d'accès à la voie (Medium Access Control) sont responsables de la satisfaction des besoins en termes de réponse temps-réel/temps-critique tout au long du réseau ainsi que de la fiabilité des communications entre les équipements du réseau. Ces paramètres temporels influent en définitive sur le contrôle des applications. (Lian *et al.*, 2001) donne une analyse et une comparaison des délais des messages pour trois types de réseaux particuliers : Ethernet, DeviceNet et ControlNet. Les caractéristiques prises en compte sont la petite taille des données et la périodicité des messages. Les besoins comprennent la garantie de transmission et des délais bornés. Les conclusions de ces travaux sont les suivantes : malgré les haut débits offert par Ethernet, ce réseau est pénalisé par le non-déterminisme de sa méthode d'accès et n'est recommandé que pour le transport de messages apériodiques ou à temps non-critique. Pour des communications caractérisées par des contraintes déterministes, Lian retient que des réseaux de terrain classiques comme DeviceNet ou ControlNet pourraient plus aisément éviter les problèmes de stabilité mis en évidence au paragraphe 1.1.2.

La problématique des SCR s'inscrit dans le cadre général des *systèmes à retard* (ou *time-delay systems* en anglais). Les systèmes contrôlés en réseau sont des systèmes à retard pour lesquels le retard est introduit par le réseau. Le groupe de travail SAR (Systèmes A Retard) du groupe de recherche MACS (Modélisation, Analyse et Conduite des Systèmes dynamiques) participe ainsi à l'animation scientifique de cette problématique. Un tutoriel des travaux et des problématiques est donné dans (Richard, 2003). Les points clés des systèmes à retard sont la modélisation des délais, le contrôle optimal, et la robustesse de la stabilité. La collecte des informations relatives au comportement du délai est alors primordiale. Par exemple, si des observateurs pour les systèmes à retard ont été proposés (Darouach, 2001), ces travaux s'appuient sur des mesures supposées du retard. Le délai est supposé connu ou calculable, de sorte que cette connaissance nécessite d'être fournie par une étape d'identification ou d'analyse des limites technologiques. Ce problème est alors d'autant plus délicat que le retard est produit par un réseau dont le comportement peut être difficilement analysable.

Les systèmes contrôlés en réseau partagent également la problématique du contrôle des retards induits par le réseau avec les systèmes téléopérés comme le contrôle d'un robot à distance qui sont très sensibles à la fluctuation des délais de transmission. Des travaux montrent qu'une solution est que le retard soit constant. Des techniques informatiques peuvent alors être utilisée pour assurer cette hypothèse. Ainsi que ce soit dans le cadre de la téléopération d'un robot (Lelevé et Fraisse, 2003) ou pour les applications distribuées (Luck et Ray, 1990), la régulation des retards variables par bufferisation peut être utilisée. L'information est volontairement retenue de façon à pouvoir la restituer avec un retard le plus constant

(annulation de la gigue).

Cette prise en compte des délais liés à la traversée du réseau se matérialise alors dans les boucles de régulation par l'ajout de blocs retardateurs. Cette intégration peut alors se traduire comme dans le modèle de l'asservissement présenté à la figure 1.5.

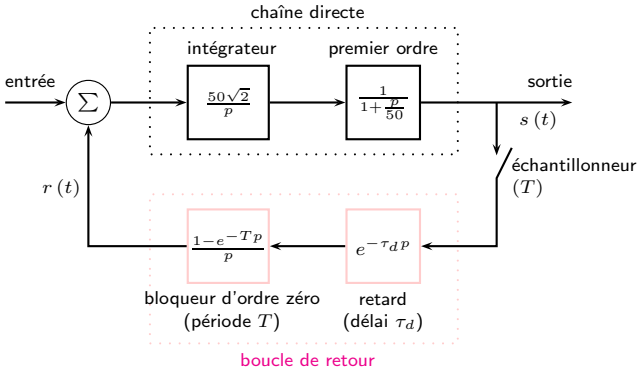


FIG. 1.5: Modèle d'asservissement distribué (Juanole et Blum, 1999)

Ce modèle est établit dans le cadre d'une architecture producteur / consommateur semblable à la figure 1.1, mais où le contrôleur est directement intégré à l'actionneur. Dans cette architecture, la boucle de retour tient compte de l'influence du réseau. Dans un premier temps, le bloc $e^{-\tau_d p}$ introduit le retard issu de la traversée du réseau qui dépend de divers paramètres tels que le protocole de transfert ou les politiques d'ordonnancement. Dans un second temps, le bloqueur d'ordre zéro matérialise une probabilité de perte non nulle des paquets. Dans ces travaux, il est admis que la dégradation de la périodicité d'émission des paquets due aux pertes peut être approximée par une période T plus large que la période initiale (ceci n'est toutefois pas juste si l'on considère des rafales de pertes). La fonction de transfert du bloqueur d'ordre zéro est ainsi approximée par celle d'un retard pur. Notons que s'il n'y a ni pertes ($T = T_0$) ni retard ($\tau_d = 0$), on retombe dans le schéma classique d'un asservissement échantillonné.

La figure 1.5 permet d'introduire différents indicateurs de performance du réseau dans le modèle d'asservissement d'une application distribuée. Dans ce modèle, le réseau est simplement utilisé pour l'échange des mesures effectuées en sortie du système. La figure 1.6 élargit le champ d'utilisation du réseau puisqu'il intègre cette fois les communications du capteur au correcteur ainsi que du correcteur à l'actionneur.

Contrairement au premier modèle, celui de la figure 1.6 n'intègre que le retard issu de la traversée du réseau. Toutefois, le système est ici totalement distribué : le contrôle se fait sur un équipement distribué. Ce modèle intègre trois délais : le temps de calcul au niveau du contrôleur ainsi que les temps de traversée du réseau du capteur au contrôleur et du contrôleur à l'actionneur. La distinction des retards suivant l'échange considéré est essentielle lorsque le système à retard est un réseau (notamment en raison des chemins multiples). Si bien que comme le montre la figure 1.7, ce modèle doit être complété lorsque l'application intègre plusieurs entrées et plusieurs sorties.

La figure 1.7 introduit un aspect supplémentaire de la difficulté d'analyse des SCRs. Outre l'influence du réseau pouvant rendre le contrôle du système instable, le réseau est un système à part entière et les caractéristiques telles que le retard dépendent de nombreux paramètres (chemins, taille des messages,

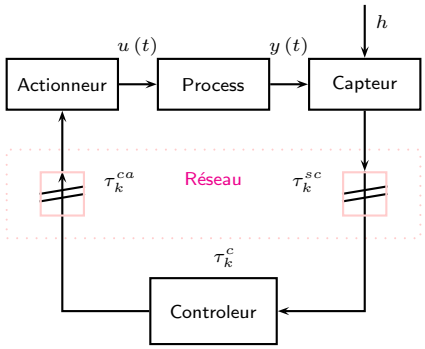


FIG. 1.6: Modèle d'asservissement distribué (Nilson, 1998)

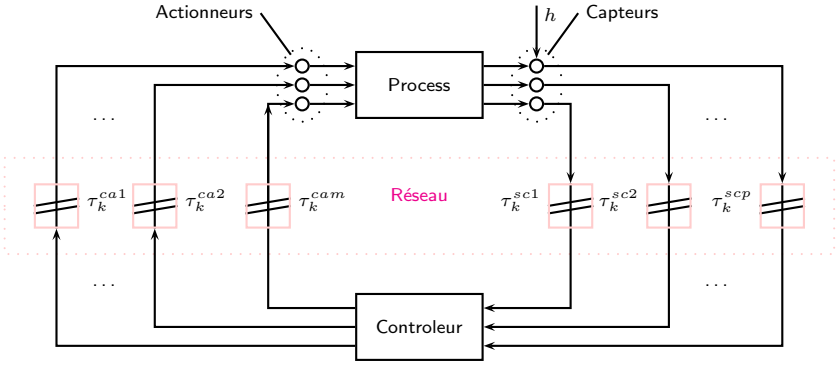


FIG. 1.7: Modèle d'asservissement distribué à multiples entrées et sorties (Nilson, 1998)

ordonnancement, protocole ...) et peuvent fortement varier. Il sera donc nécessaire de formuler une analyse des retards dédiée pour chaque application et plus particulièrement pour chaque échange.
Les paragraphes suivants présentent les différents axes d'étude des systèmes contrôlés en réseau tels la commandabilité et la stabilité du contrôle ainsi que l'intégration des niveaux de performance du réseau dans le contrôle de l'application.

1.1.4 Contrôle & stabilité

Un axe d'étude concerne la partie contrôle/commande. L'objectif est de prendre en compte l'influence du réseau dans les diverses expressions de la commande. Les études des systèmes contrôlés en réseau reposent généralement sur des inégalités linéaires ou bilinéaires de matrices ou encore sur des chaînes de Markov. Ainsi, (Nilson, 1998) introduit différents modèles de délai dans les SCR, le délai étant considéré tour à tour fixe, indépendant et stochastique ou comme un processus markovien. Un contrôle optimal stochastique des SCRs est donné dans ces deux derniers cas.

(Zhang *et al.*, 2001) notent deux grandes approches dans l'accommodation des SCR aux problèmes posés par le réseau. Une première solution est de concevoir un système de contrôle indépendant des délais et des pertes mais de concevoir un protocole de communication qui minimise l'occurrence de ces événements. C'est ainsi que des techniques de contrôle de congestion et des algorithmes à évitement de congestion ont été proposées. La seconde approche revient à traiter le protocole réseau et le trafic comme des conditions données et de concevoir des stratégies qui prennent explicitement en compte les

challenges précédemment identifiés. Comme nous l'avons montré à la figure 1.4, un point important des SCR concerne la stabilité de l'application. Ce point est notamment traité dans les travaux suivants. (Walsh *et al.*, 1999) présentent une analyse de la stabilité des SCR à partir de politiques d'ordonnancement statique et dynamique des données des capteurs. (Zhang *et al.*, 2001) étendent ces travaux et propose une solution moins conservative puisqu'elle ne nécessite pas que la fonction de Lyapunov soit strictement décroissante à tout instant. Finalement, il donne une indication sur la relation entre le pas d'échantillonnage et le délai induit par le réseau. Une étude de la stabilité et différentes méthodes de compensation des délais et des pertes sont étudiées.

1.1.5 Vers une vision intégrée

Dans les travaux précédents, les caractéristiques du réseau sont simplement intégrées comme données d'entrée pour l'analyse de la commandabilité et de la stabilité de l'application ; il n'existe pas de modélisation du réseau en vue de le caractériser. (Juanole et Blum, 1999) notent pourtant que le développement d'applications sur les systèmes distribués nécessite des études à caractère pluridisciplinaire : on ne peut plus se contenter d'évaluer le comportement du système distribué par rapport à des objectifs de Qualité de Service de la communication ; il faut impérativement relier cette Qualité de Service aux indices de performances spécifiques des applications.

(Juanole et Blum, 1999) proposent une méthodologie d'évaluation de l'asservissement du système distribué présenté sur la figure 1.5. L'évaluation de l'asservissement du système est basée sur la stabilité. Le système distribué concerné par l'application est modélisé par un réseau de Petri temporisé stochastique représenté sur la figure 1.8.

L'intérêt majeur du réseau de Petri est l'intégration sur le même modèle de la partie applicative avec la partie communication du SCR. Les transitions probabilistes permettent de faire évoluer le modèle suivant les particularités du réseau (pertes, protocole) et les transitions temporisés de modéliser les phases d'attentes. L'étude du comportement dynamique de ce modèle est ensuite utilisée afin de déterminer deux paramètres de la Qualité de Service du réseau : τ_d le délai et T, la période moyenne entre deux échantillons. Ces deux paramètres sont ensuite réutilisés dans la boucle de régulation et dans l'expression de la marge de phase de l'asservissement qui permet de définir la stabilité de l'asservissement.

Dans le cadre de nos travaux, nous avons ainsi initié cette approche à partir d'une simple modélisation d'un échange maître / esclave. Le réseau de Petri temporisé de la figure 1.9 fait ainsi émerger trois problèmes majeurs. Les deux premiers sont de niveau application et consistent à étudier les périodes de rafraîchissement des effecteurs ainsi que leur promptitude par rapport aux tâches qui s'exécutent cycliquement sur l'automate. La troisième est relatif au retard subi sur le réseau.

Le modèle de la figure 1.9 prend en compte deux entités (un automate et un capteur). L'automate exécute cycliquement un ensemble de tâches, la durée du cycle est définie par la transition t_1 et est bornée par $[a, b]$. Le rôle du capteur est de mesurer une variable (cycle de production défini par t_{11}) et de la communiquer à l'automate à chaque fois qu'il la lui demande. Le scénario de communication considéré est de type maître / esclave. L'automate (le maître) défini un cycle de *polling* (transition t_7) de requêtes auxquelles le capteur répond instantanément (place $P15$). La modélisation du réseau n'est pas détaillée. Elle est simplement représentée par les transitions t_{20} et t_{21}, qui permettent d'introduire le délai engendré par sa traversée. Il est à noter que ce délai est supposé borné.

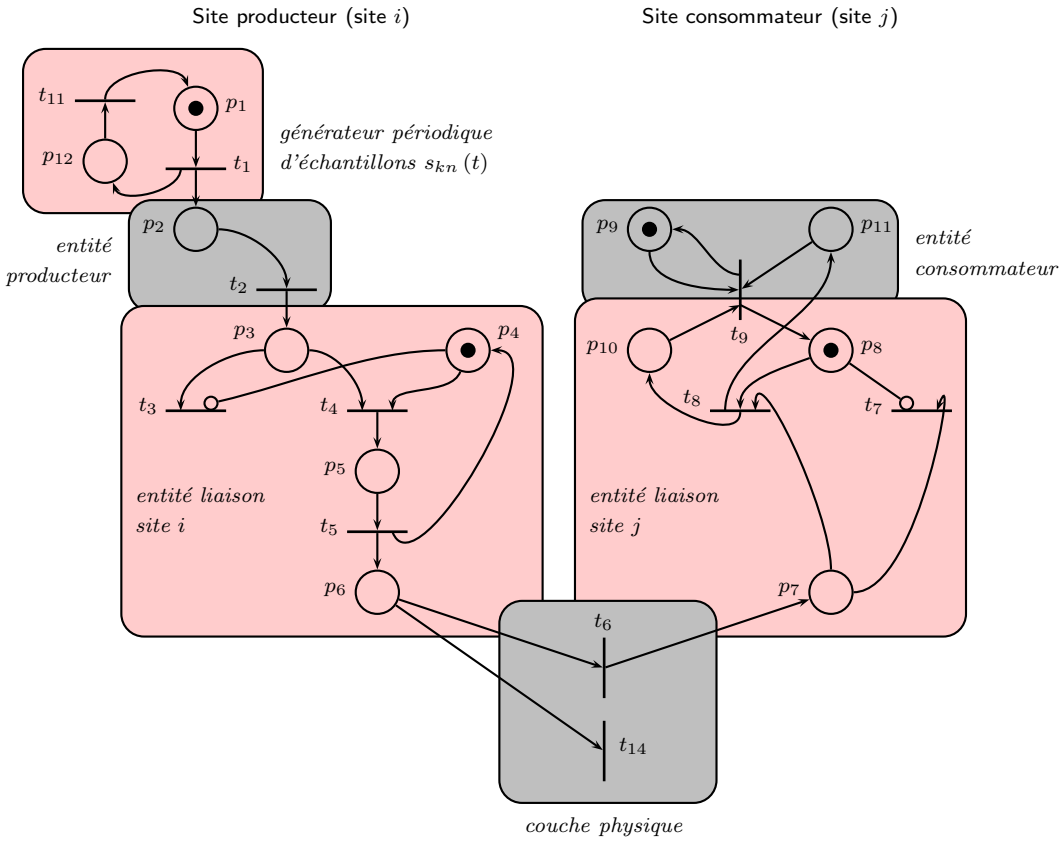


FIG. 1.8: Réseau de Petri d'un système distribué temps-réel (Juanole et Blum, 1999)

La particularité de ce RdPT est de prendre en compte l'asynchronisme au niveau de l'automate entre le cycle d'exécution des tâches (t_1) et le cycle de collecte des informations distantes (t_7). Les requêtes *via* le réseau ne sont pas directement forcées dans le cycle applicatif. Si la durée du cycle est ainsi garantie, cette technique nécessite l'utilisation d'une mémoire partagée dans laquelle l'automate viendra directement piocher les valeurs mesurées à distance ($P2$ réprésente l'opération de lecture locale par cycle et $P3$ l'état de la mémoire dédiée à cette valeur). Le but du cycle $P5$, $P6$ est donc de mettre à jour ces valeurs en émettant les requètes nécessaires. Le même principe est nécessaire sur le capteur où la fréquence de *polling* établie par l'automate peut être différente du cycle de production de la valeur mesurée stockée dans $P13$.

L'intérêt de ce RdPT est l'étude des opérations de lecture / écriture sur cette mémoire partagée qui est une zone frontière entre deux processeurs complètement autonomes. Les possibilités d'analyse offertes par ce modèle concernent notamment la vérification de la compatibilité de la durée des cycles de production et de scrutation en fonction des délais introduits par le réseau. Prenons par exemple le cas de la place $P13$ qui représente l'état de la mémoire tampon de la valeur mesurée par le capteur. La présence d'un jeton y indique que cette variable a été mise à jour depuis la dernière requête de l'automate reçue en $P13$. Ainsi, si la transition t_{16} est franchie, cela signifie que la valeur qui sera transmise à l'automate sera la même que la précédente. A l'inverse, le franchissement de la transition t_{13} correspond à une configuration où le capteur met à jour la valeur plus souvent qu'elle n'est scrutée par l'automate, ce qui peut se traduire par un manque de réactivité de l'automate.

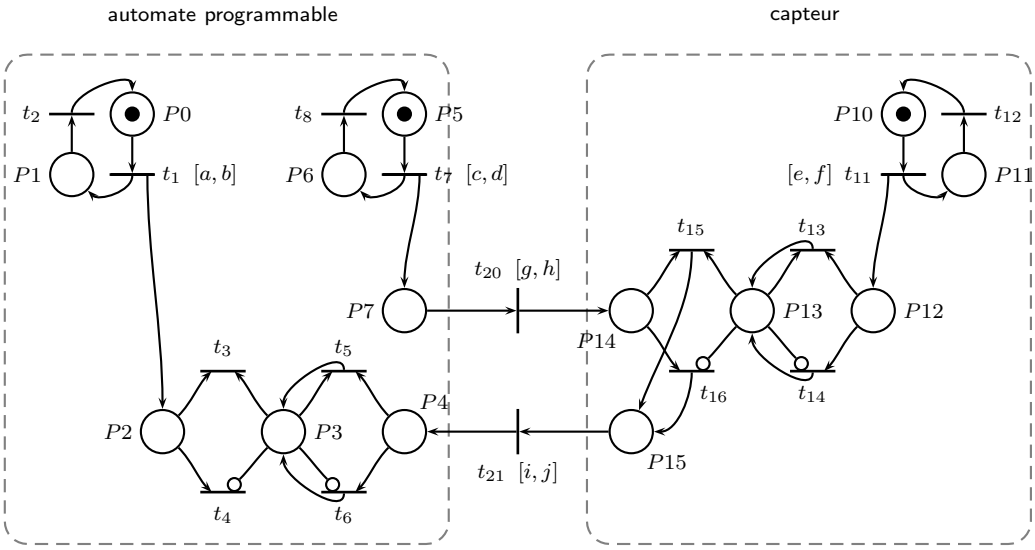


FIG. 1.9: Réseau de Petri d'un échange maître / esclave

Cette étude (qui pourrait être étendue au niveau de l'automate) est intéressante car elle illustre bien l'intérêt de ce type d'analyse pour la vérification et l'évaluation de performances d'architectures de commande comme les systèmes contrôlés en réseau. L'exemple de la figure 1.9 reste toutefois simple et à détailler afin d'obtenir un modèle complet de l'architecture (de la modélisation de la tache applicative à l'équipement d'interconnexion réseau) et vérifier ainsi les éventuels blocages ou problème d'incohérence d'une information distribuée sur deux automates à un instant donnée.

Dans (Poulard *et al.*, 2004), les réseaux de Petri colorés sont utilisés pour modéliser le comportement dynamique d'une architecture de commande basée sur Ethernet. Le modèle est détaillé pour chacun des composants de l'architecture (réseau, automate, commutateur et process).Ces travaux ont la particularité de présenter un modèle complet d'une architecture de commande, c'est-à-dire la description des tâches, l'interface de communication réseau mais aussi la modélisation du protocole de communication utilisés incluant les équipements réseaux. Si dans (Juanole et Blum, 1999), la modélisation par réseau de Petri permet d'évaluer les modifications de la commande d'un système à asservissement, (Poulard *et al.*, 2004)(Marsal *et al.*, 2005) montrent que l'évaluation d'une commande d'un système à évennements discrets distribué sur un réseau ouvert comme Ethernet est liée au déterminisme des états de la commande. En effet, la distribution des communications sur un réseau implique que les périodes de scrutations soient correctement définies conformément aux spécificités des tâches applicatives. Comme nous l'avons initié sur la figure 1.9, cette analyse permet d'ajuster la réactivité de la commande. De plus, la variance des délais engendrés sur le réseau peut amener dans le cas d'une scrutation cyclique des situations où un nouveau cycle de scrutation démarre alors que les réponses n'ont pas encore été reçues. La distribution des délais issue du comportement du protocole de communication et des équipements utilisés doit être prise en compte dans le comportement du système.
Tout ceci montre que les spécificités du réseau influent aussi bien sur les systèmes à événements discrets que sur les systèmes à espace d'état continu et que les influences sur la commande doivent être étudiées dans les deux cas, sachant que les outils d'évaluation des systèmes à événements discrets font apparaître

d'autres points cruciaux.

Outre l'étude de l'accès au réseau, l'accès concurrent de plusieurs tâches à un même processeur doit également être pris en compte. Les travaux présentés dans (Krákora et Hanzalek, 2004)(Krákora *et al.*, 2004) visent l'étude des propriétés temporelles (temps de réponse, ordonnançabilité des processus périodiques) et des propriétés logiques (deadlock, exclusion mutuelle, accès à base de priorité) des applications comprenant deux types de ressources partagées : le processeur et le bus. La modélisation du protocole de communication (ici, le réseau de terrain CAN) est réalisée au moyen d'automates temporisés. L'intérêt est une modélisation complète de l'application et l'objectif est une aide à la vérification complète de ses propriétés. D'autres travaux similaires se portent également sur le réseau Ethernet (Dolejš *et al.*, 2004).

1.1.6 Simulations

L'un des points clés des systèmes contrôlés en réseau est la vérification des performances du contrôle pour divers niveaux de performance du réseau. L'approche classique consiste alors à utiliser des outils de simulation.

Deux modélisations du réseau ont été récemment proposées pour l'environnement MATLAB/Simulink (dédié à la simulation de la partie contrôle) : NCsimulator (Lian, 2001) et TrueTime (Cervin *et al.*, 2003)[2]. Elles permettent alors d'inclure dans la simulation du contrôle des effets caractéristiques du réseau.

(Branicky *et al.*, 2003) introduisent un nouvel ensemble de travaux dont l'idée originale est la co-simulation. Dans cette approche, l'outil de simulation réseau `ns-2` est étendu pour prendre en compte les nœuds des SCR (contrôleurs, capteurs, actionneurs ...) qui interagissent sur un réseau Ethernet. Cette extension (Agent/Plant pour `ns-2`) est utilisée afin de raffiner la conception du réseau et du système de contrôle. Cette idée de co-simulation est reprise par (Hartman, 2004) qui réutilise les résultats de la simulation sous `ns-2` dans une simulation sous MATLAB/Simulink du système de contrôle global. L'avantage est une modélisation plus fine de la Qualité de Service du réseau.

1.1.7 Derniers développements

L'étude des systèmes contrôlés en réseau donne lieu depuis ces dernières années à diverses manifestations et conférences. Ainsi, on peut citer le projet européen STREP **NeCST** (Networked Control System Tolerant to faults)[3] dont nos travaux sont partie intégrante. Citons également le numéro spécial consacré au réseau et au contrôle de l'*IEEE Control Systems magazine*. Plus récemment, une session invité et une session poster ont été organisées dans le cadre de la 16$^{\text{ème}}$ conférence *IFAC World Congress*. Parmi les travaux présentés, (Yang *et al.*, 2005) considèrent que les délais sont à temps variant et stochastique. Ainsi, (Yang *et al.*, 2005) proposent une méthode de compensation des délais de transmission stochastique supposés conformes à une chaîne de Markov et entier multiple de la période d'échantillonnage. (Colandairaj *et al.*, 2005) notent alors que la principale difficulté des SCR est l'introduction de délais dans la boucle de contrôle. L'analyse de ces délais est par conséquent très importante pour mieux comprendre les effets sur le système de contrôle. Ainsi, la nature des délais doit être elle même correctement modélisée. Pour

[2] `http ://www.control.lth.se/~dan/truetime/`

[3] `http ://www.strep-necst.org`

1er Workshop on Networked Control System Tolerant to faults, `http ://www/strep-necst.org/necst05/`

(Colandairaj *et al.*, 2005), les délais sur un réseau de communication ne sont pas aléatoires mais dépendent de nombreux facteurs, comme par exemple la charge de trafic ou le type de protocole. De mauvais modèles de délais peuvent conduire à des suppositions non précises produisant finalement de mauvaises conclusions lors de l'analyse de stabilité. (Colandairaj *et al.*, 2005) mentionnent alors que la modélisation des délais en utilisant des distributions statistiques n'est pas suffisante et qu'une modélisation du réseau qui reproduirait les effets des protocoles réseaux est nécessaire. Il se place dans le même contexte de simulation du réseau sous Matlab mais le réseau ciblé diffère dans la mesure où il s'agit d'une étude des réseaux sans-fil. (Colandairaj *et al.*, 2005) visent un nouveau simulateur réseau dédié au réseau sans fil IEEE 802.1b. Ce simulateur est implémenté comme une C MEX S-FUNCTION dans MATLAB/Simulink. Deux cas sont étudiés afin de mettre en avant les performances de ce type de réseau comme support de système contrôlé en réseau. L'utilisation de la simulation comme source d'analyse de la performance du réseau devra être progressivement accompagnée de mesures expérimentales de la qualité de service comme le délai, les pertes ou la bande passante. Pour cela, les futurs travaux s'intéresseront aux développements de la thématique métrologie réseau. Ainsi, (Michaut, 2003) illustre comment des mesures de délai peuvent être utilisé dans le cadre de la téléopération d'un robot pour l'adaptation de la loi de commande. Il démontre également la difficulté à obtenir ces mesures du retard introduit par le réseau.

Une conclusion de la session invitée de la conférence IFAC World Congress 2005 s'est portée sur le caractère premier des résultats actuels concernant les systèmes contrôlés en réseau. Il n'existe pas encore de solution globale prenant en compte une interaction entre modélisation du réseau, de la commande, et du système. De plus, il est à noter que les résultats obtenus devront être ajustés, modifiés ou abandonnés à chaque fois que le type de réseau sera différent. Examinons alors les réseaux utilisés actuellement.

1.2 Des réseaux déterministes sur mesure ...

Durant les dernières décennies, différents protocoles réseaux ont été développés et implémentés pour supporter le besoin de communications temps réel dans les couches basses du processus industriel. Les plus populaires sont PROFIBUS (Process Field Bus (Nutzer Organisation e.v., 1992)), FIP (Factory Automation Protocol (AFNOR, 1990)) ou CAN (Controller Area Network (ISO, 1993)) qui utilisent chacun un protocole d'accès au médium déterministe. Leur principale caractéristique est d'assurer que les délais de bout en bout supportés par les messages soient bornés et restent limités comparés aux contraintes temporelles des applications. Conçus dès le départ pour prendre en compte ces problèmes de déterminisme de transmission, ces protocoles réseaux ont été fortement implémentés dans les entreprises et font toujours l'objet de nouveaux travaux (notamment avec l'utilisation de la technologie sans-fil).

Néanmoins, ces réseaux ne sont pas sans inconvénients. (Song, 2001) remarque ainsi que *EN50170* (CENELEC, 1996), considéré comme un standard européen pour les réseaux de terrain, n'est rien de plus qu'une collection de trois protocoles indépendants : *Profibus*, *P-Net*, *WorldFip*. Comme chacun a sa propre norme, les solutions proposées sont très "constructeur" et ne sont pas inter-opérables. De plus le coût de leur implémentation reste très élevé. Ces réseaux industriels traditionnels sont pénalisés par la faiblesse du débit qu'ils proposent (1 à 5 Mb/s). Ces taux sont en effet relativement inadaptés à la transmission de vidéos dans le cadre d'applications de téléopération ou de télésurveillance. Enfin, un certain manque de flexibilité du réseau peut être mis en avant face à des événements de type réorganisation topologique

ou coupure de lien.

Les industriels attendent clairement un réseau standardisé, moins cher et homogène, c'est-à-dire, le même dans les niveaux du CIM (Computer Integrated Manufacturing). Sachant que les réseaux informatiques sont majoritairement de type Ethernet/IP, l'une des propositions les plus soutenues est l'implémentation d'Ethernet dans les réseaux industriels. Cette évolution est confirmée dans les secteurs automobile (Jaguar), pharmaceutique (Boehringer Ingelheim), aéronautique (EADS A380).

1.3 . . . aux réseaux Ethernet

1.3.1 Contexte

Comme le note (Decotignie, 2001), Ethernet est un réseau attractif de plus en plus utilisé comme solution d'interconnexion des équipements de terrain dans l'automatique industrielle. Bien que l'idée ne soit pas nouvelle, la disponibilité des technologies bon marché font d'Ethernet une solution attrayante comme réseau de terrain. Pour Decotignie, aucune réponse définitive ne peut tout à fait être apportée. Récemment, une grande attention est portée sur Ethernet dans le domaine des communications industrielles. Un nombre important de manufacturiers offre des produits de communications industrielles basés sur Ethernet et TCP/IP comme solution d'interconnexion d'équipements industriels au premier niveau de l'automation. D'autres réduisent le champ d'application aux communications entre des équipements d'automation (automates . . .) et complètent l'intégration avec des réseaux de terrain existant. Les fabricants d'équipements Ethernet commencent aussi à proposer de nouveaux produits, comme Cisco avec son commutateur *Catalyst 2955*, spécialement dédiés au contexte industriel.

L'idée n'est pas nouvelle, et des solutions utilisant Ethernet ont été proposées depuis les 20 dernières années. Les premiers produits utilisant Ethernet ou autres dérivés dans le contexte industriel apparaissent dans les années 80. On peut citer par exemple le réseau MAP (Manufacturing Automation Protocol) initialement prévu pour le standard IEEE 802.4 à jeton sur bus qui a été adapté pour Ethernet (ESPRIT consortium CCE-CNMA, 1995). La communauté scientifique reconnaît alors la supériorité de CSMA/CD par rapport aux autres protocoles d'accès à la voie en terme de flexibilité, robustesse et simplicité. C'est pourquoi les mécanismes de priorité pour réseau à échéances ne furent pas ajoutés au standard.
L'acceptation d'Ethernet sur le marché industriel a été relativement longue comparée à celle des environnements informatiques, d'où peu de produits au début de cette idée et un retour en grâce avec l'utilisation d'Ethernet comme réseau de backbone de l'entreprise. Ce retour s'explique également par la chute des coûts du matériel (avec l'utilisation intensive dans les réseaux informatiques, apparition de nouveaux câbles RJ45 de bon marché) et l'augmentation continue des débits (IEEE 802.3u 100baseT, 1000baseT, . . .).

Decotignie présente alors sept bonnes raisons d'utiliser Ethernet comme réseau de terrain : la disponibilité d'un grand nombre de circuits électroniques de bon marché, la simplicité d'intégration avec Internet (même technologie), le bénéfice des applications et protocoles amenés par TCP (FTP ou encore le format XML pour le typage / définition des données (Wollschlaeger, 2000)), l'évolution constante de la bande passante sur Ethernet, l'universalité et la compatibilité (une solution unique plutôt que plusieurs réseaux de terrain), la saturation et la pauvreté de l'évolution des réseaux traditionnels et la demande du marché d'Ethernet (de plus en plus de produits).

L'intérêt est de fournir des services d'interopérabilité entre :

- les constructeurs d'équipements,
- les mondes de la production et de la gestion
- les sites distribués des compagnies et de leurs fournisseurs, pour développer sur Ethernet et au travers de l'Internet, des applications de téléopération, de télémaintenance ...

Ce dernier point conduit à l'échange d'images, sons, vidéo, qui nécessitent de la part du réseau de plus en plus de débit. A l'inverse des réseaux de terrain, Ethernet offre des débits allant jusqu'au gigabit/s, ce qui a été considéré dans un premier temps comme suffisant pour satisfaire les contraintes temps-réel des applications.
Enfin, l'un des derniers avantages d'utiliser Ethernet dans les systèmes distribués est qu'il permet une intégration verticale des communications de terrain avec le Manufacturing Execution Systems (MES).

1.3.2 Présentation

Ethernet est la technologie LAN la plus populaire. Elle est bon marché et robuste, du fait de sa facilité de déploiement et de son évolutivité, et cherche désormais à élargir son champ d'application aux réseaux industriels où les besoins temps-réel doivent être satisfaits. La conception d'un réseau de contrôle temps-réel sur Ethernet reste néanmoins difficile dans la mesure où son protocole MAC, et plus particulièrement le protocole CSMA/CD avec l'algorithme de résolution des collisions BEB (Binary Exponential Back-Off) présente des délais non prédictibles. Le rapport (Alves *et al.*, 2000) vise alors à présenter le non déterminisme inhérent à Ethernet et les nouvelles technologies apportant de réelles avancées dans la recherche d'un contrôle temps-réel.

Le Xerox Palo Alto Research Centre a commencé à développer Ethernet dans les années 70 comme technologie des LAN pour les environnements informatiques. Fin des années 70, Xerox est rejoint par Digital Equipment Corporation et Intel formant un consortium qui publiera en 1980 la première spécification d'Ethernet. La spécification sera reprise par l'IEEE (Institute of Electrical and Electronic Engineers) qui la publie comme standard en 1983 (IEEE Computer Society, 2002). Ce standard sera par la suite repris par l'ISO (International Standards Organisation) et par l'IEC (International Electrotechnical Commission) ISO/IEC 8802-3.
Aujourd'hui Ethernet est la technologie LAN la plus diffusée. Malgré la disponibilité offerte par d'autres réseaux haut-débit (ATM (Asynchronous Transfer Mode (The ATM forum, 1996)) et FDDI (Fibre Distributed Data Interface (ISO, 1989)), Ethernet reste très intéressant du fait de son faible coût, de sa maturité et de sa stabilité. Néanmoins, Ethernet présente un sérieux désavantage pour supporter des messages de contrôle temps-réel. Puisqu'un message peut rentrer en collision avec un autre, il est difficile de prédire les délais de transmission entre deux nœuds du réseau.

Ethernet est une spécification des couches 1 et 2 du modèle OSI définie dans le standard IEEE 802.3 (IEEE Computer Society, 2002). En fait, c'est une spécification d'un profil du protocole CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Mais c'est aussi bien plus. En effet, CSMA/CD ne définit pas l'algorithme de résolution d'une collision alors qu'Ethernet définit l'algorithme BEB ainsi que les temps slot, la longueur des données, le médium physique ...

Ethernet implémente une version 1-persitent de CSMA/CD puisqu'une station prête à émettre des données les transmet dès que le canal est libre avec une probabilité de 1. Comme le montre la figure

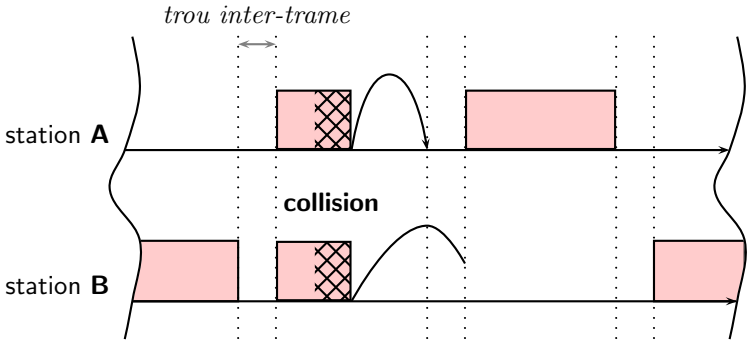


FIG. 1.10: Apparition d'une collision avec CSMA/CD

1.10, si au moins deux stations décident d'émettre "en même temps", c'est-à-dire qu'elles ont vérifié simultanément que le médium partagé est libre (non utilisé) et émettent en un temps plus court que le temps de propagation du signal sur le médium, une collision apparaît. Chaque station impliquée dans la collision se place en attente conformément à l'algorithme BEB. En fait chaque station a un compteur de collision n initialement fixé à 0. A chaque nouvelle collision d'une de ses trames, chaque station incrémente son compteur de 1. Chaque station impliquée dans la collision choisit alors un temps d'attente avant tentative de retransmission selon la distribution uniforme compris entre 0 et $(2^n - 1)$ temps slot[4].
Le standard Ethernet définit également plusieurs implémentations de la couche physique. Par exemple, 10Mbits, 100Mb/s, 1000 Mb/s. FastEthernet spécifié en 1995 IEEE 802.3u, GigaEthernet depuis 1998 IEEE 802.3z Terabit est actuellement en phase de recherche.

Ethernet, basé sur CSMA/CD, ne peut pas être directement utilisé pour les applications à temps-critique. En effet, sa méthode de contrôle d'accès au médium peut générer des collisions durant la transmission d'une trame (figure 1.10). De plus, le mécanisme de résolution des collisions est de nature stochastique. Du fait de son protocole non-déterministe (Alves *et al.*, 2000), Ethernet a simplement été utilisé pour des applications non sensibles au temps. Néanmoins, les performances d'Ethernet (débit, interopérabilité, flexibilité, standardisation) confirment l'intérêt d'Ethernet pour les applications à temps critique comme dans les environnements industriels.

1.3.3 Disparité des délais sur Ethernet

Afin d'illustrer la forte disparité des délais sur Ethernet, nous présentons ici le résultat d'une expérimentation que nous décrirons plus en détail dans la suite du document (voir paragraphe 4.4). La figure 1.11 donne les valeurs du délai d'une série de communications sur un réseau Ethernet. Le réseau est relativement peu chargé.

La figure 1.11 montre que si les délais sont généralement faibles et varient très peu, certaines trames subissent des retards beaucoup plus grands. C'est là la grande problématique des réseaux basés sur le protocole CSMA/CD et à zones de congestion. Malgré les grands débits offerts par ces réseaux, il n'y a aucune garantie sur les délais et des cas rares (pires cas) se caractérisent par de forts retards pouvant condamner l'utilisation d'Ethernet dans les systèmes contrôlés en réseau. L'une des critiques les plus fortes faites à l'encontre d'Ethernet de type partagé (configuration native d'Ethernet) dans le cadre d'une

[4]un temps slot est 512 temps bit, soit $51,2\ \mu s$ pour réseau 10 $Mbits/s$

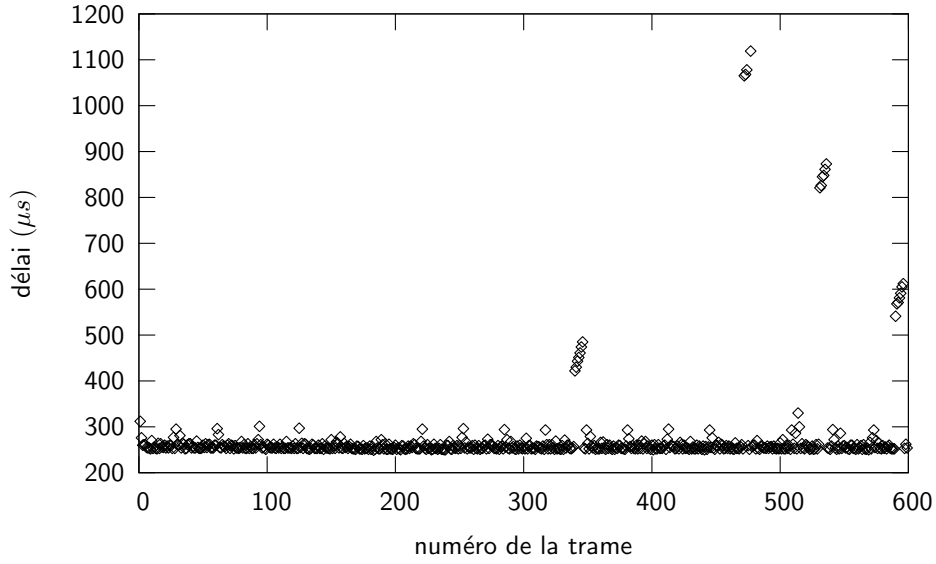


FIG. 1.11: Délais sur Ethernet

utilisation pour des applications temps-réel est donc l'introduction d'un délai d'accès non déterministe conduisant à un comportement d'ensemble du réseau non prévisible.

Le terme d'Ethernet partagé (Shared Ethernet) correspond à une utilisation d'un médium de communication physique tel qu'un bus ou un hub, c'est-à-dire quand tout le trafic est diffusé à toutes les stations du réseau. De sorte qu'en augmentant le domaine de diffusion, les probabilités d'apparition des collisions sont également augmentées. Le caractère non déterministe d'Ethernet provient majoritairement de l'algorithme BEB de résolution des collisions qui présente un délai non prévisible. Cette caractéristique est renforcée par le fait qu'une station peut monopoliser l'accès au médium, c'est-à-dire émettre consécutivement plusieurs trames sans que d'autres stations puissent transmettre leurs données, quelque soit leur niveau de criticité. C'est l'effet de "capture". Cet effet peut conduire également à ce qu'une trame se trouve confrontée à 16 collisions successives et ne puisse finalement pas être transmise (valeur maximale de tentatives définies dans la norme). C'est l'effet famine. Avec Ethernet, on n'est même pas sûr que la trame sera émise ! Ce protocole peut donc être non équitable alors qu'il présente à la base un mécanisme d'accès à la voie démocratique. Une certaine relation entre le délai de transmission et la charge du réseau peut ainsi être mise en avant : pour un réseau peu chargé, la fréquence d'apparition des collisions est relativement faible, si bien que le délai sera relativement court (de l'ordre de la ms). Si la charge augmente, les délais vont alors radicalement augmenter pour aller jusqu'à l'écroulement d'Ethernet autour de 60% de charge. Dans ce cas, les famines apparaissent. Ceci étant dit, il est clair que pour une application à très fortes contraintes temps-réel, l'algorithme du BEB pose problème et rend l'utilisation d'Ethernet définitivement trop indéterministe. Néanmoins, le large panel de produits (matériels et logiciels) d'Ethernet et l'aspect bon marché conduisent à une utilisation de plus en plus intensive.

Le temps d'accès au medium n'est théoriquement pas majorable sur CSMA/CD. En pratique, la plupart des opérateurs supposent que les réseaux opèrent bien au-dessous de leur capacité (un pourcentage faible), et que par conséquent, les collisions sont rares et la latence est basse.

De même, la périodicité ne peut pas être strictement garantie sur Ethernet. Il y toujours de la gigue (variation des retards). L'important est simplement de recevoir la donnée avant la fin de l'échéance, mais encore faut-il en avoir la garantie *a priori* ! La gestion de l'indication de la cohérence temporelle d'une donnée passe généralement par l'estampillage de trames. Cela suppose la synchronisation des horloges. La synchronisation des horloges est un point important pour gérer l'ordre des événements et vérifier la périodicité. De nouveaux standards sont apparus améliorant la robustesse en environnement industriel d'Ethernet (protection IP 67, nouveaux connecteurs autres que RJ45).

Rappelons toutefois qu'Ethernet ne décrit que les niveaux 1 et 2 du modèle OSI et suppose l'utilisation de protocoles de couches hautes appropriés à la différence de réseaux comme Profibus ou FIP.

1.3.4 Structuration actuelle

L'attrait de la technologie Ethernet se concrétise par la prise de positions sur le marché de constructeurs et l'apparition d'organisations institutionnelles ayant pour but la promotion et la standardisation d'Ethernet pour les applications industrielles. On peut citer ainsi le constructeur Rockwell Automation (Rockwell International Corporation, 1998) qui vise à définir l'intérêt pour ses clients à utiliser Ethernet. D'autres manufacturiers comme Richard Hirschmann (Hirschmann GmBH, n.d.) supposent le déploiement d'Ethernet commuté comme infrastructure des applications industrielles à temps-réel. Les grands constructeurs développent actuellement des solutions réseaux basées sur Ethernet : *Rockwell* avec *EtherNet/IP* (Brooks, 2001), *Siemens* avec Profinet ou encore *Schneider* avec *Modbus TCP/IP*. C'est ainsi que l'IEC 61158 qui constitue la référence des standards des réseaux de terrain intègre désormais ces réseaux basés sur Ethernet comme HSE (High Speed Ethernet) de Fieldbus Foundation. C'est ce que montre la figure 1.12

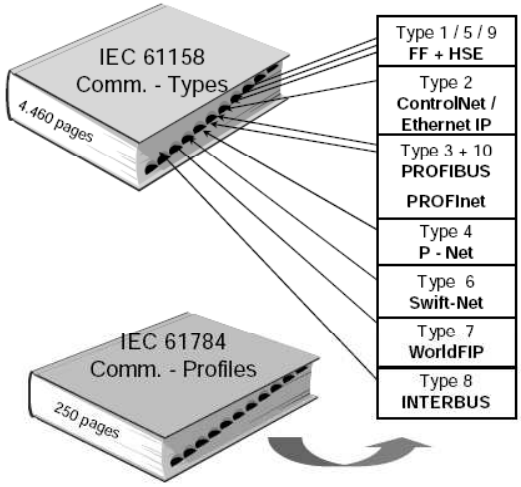


FIG. 1.12: The International Fieldbus - IEC 61158

Peu de produits se sont maintenus durant la première phase. Mais aujourd'hui, ils sont de plus en plus nombreux et la revue *The Industrial Ethernet Book*[5] maintient à jour une liste de 680 produits industriels.

[5]http ://ethernet.industrial-networking.com/

De plus, des organisations comme l'IAONA[6] (Industrial Automation Networking Alliance) ou l'IEA[7] (Industrial Ethernet Association) promeuvent Ethernet comme le nouveau « standard en environnement industriel ».

L'Industrial Networking Alliance (IAONA) a été fondée aux Etats Unis avec le but d'établir Ethernet comme le standard dans les environnements industriels. Depuis novembre 1999, la même organisation joue un rôle similaire en Europe (IAONA Europe[8]) de promotion d'Ethernet pour les applications industrielles et embarquées. L'Industrial Ethernet Association (IEA) est une autre association fondée en 1999 qui vise à établir les standards de déploiement des produits Ethernet sur le marché industriel.

1.4 Délimitation de notre travail

Dans cette partie, nous avons montré que le domaine d'étude des systèmes contrôlés en réseau est relativement vaste et peut donner lieu à différentes recherches. Un seul type de réseau utilisé pour interconnecter les équipements de l'application est déjà un facteur discriminant. L'utilisation d'Ethernet comme support de communications d'applications à temps critique reste néanmoins problématique du fait du non déterminisme de son protocole d'accès à la voie. Avec Ethernet, rien ne garantit l'arrivée du message dans un temps donné. Compte tenu de l'évolution de l'utilisation du réseau Ethernet dans les réseaux industriels, notre étude se focalisera sur les problèmes posés par cette utilisation.

Le positionnement de ces travaux peut alors être résumé au travers du schéma d'étude du projet européen NeCST.

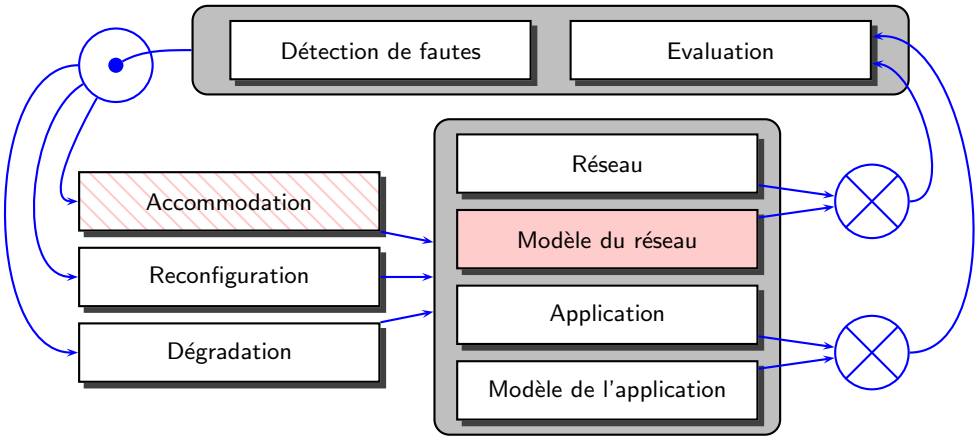


FIG. 1.13: Schéma d'étude du projet européen NeCST

Le schéma de la figure 1.13 intègre les différentes étapes de l'analyse d'un système contrôlé en réseau en prenant en compte les fautes survenant à la fois sur le process et sur le réseau. Différentes études ont déjà été menées au CRAN autour de cette problématique. On peut par exemple citer les travaux de (Michaut et Lepage, 2003) concernant l'adaptation de l'application à la qualité de service du réseau. D'autres

[6]http ://www.iaona.com
[7]http ://www.industrialethernet.com
[8]http ://www.iaona-eu.com

travaux (Krommenacker, 2002), (Rondeau *et al.*, 2001) se sont portés sur les aspects reconfiguration, et plus particulièrement sur l'optimisation de la topologie des réseaux Ethernet. La situation de nos travaux sur ce schéma est concentrée sur les aspects réseau et modèle de réseau.

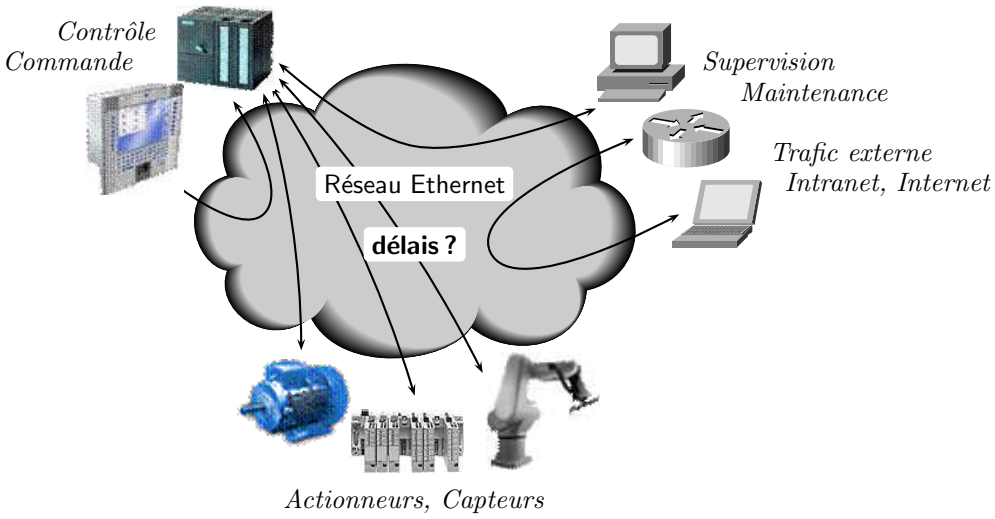


FIG. 1.14: Contexte d'étude

Comme le montre la figure 1.14, notre étude porte sur une évaluation de performances des réseaux basés sur Ethernet dans le contexte de systèmes contrôlés en réseau. Les hypothèses qui seront prises au niveau des entrées du réseau seront donc définies dans le cadre applicatif des systèmes contrôlés en réseau. L'évaluation de performances visera à répondre à un critère de satisfaction des besoins en temps-réel. Pour cela, un modèle fonctionnel et analytique du réseau Ethernet sera présenté en vue de majorer les temps de traversée du réseau. Le périmètre des travaux sera par conséquent restreint aux couches 1 et 2, les seules implémentées par Ethernet.
Ce chapitre nous a permis de mettre en évidence le problème de l'indéterminisme de l'accès au medium d'Ethernet. Le chapitre suivant présente alors un état de l'art des travaux visant justement à améliorer ses performances, voire à le rendre déterministe.

Chapitre 2

Améliorer les performances d'Ethernet

Ce chapitre présente un état de l'art des travaux visant à améliorer les performances, voire à rendre déterministe[1] les réseaux basés sur Ethernet pour les applications temps-réel. Il détaille dans un premier temps les différentes évolutions technologiques apparues dans les dernières décennies. Ensuite, les travaux sont classés suivant la nature de la topologie utilisée : partagée ou commutée. Enfin, les différentes propositions sont comparées ; nous permettant de positionner notre travail relativement à la contrainte de déterminisme.

2.1 Ethernet à travers les âges (du linéaire à l'étoile)

Un des grands avantages d'Ethernet repose sur son évolutivité qui est due à sa standardisation. En effet, plusieurs groupes de travail ont été crées dans le cadre de l'IEEE Computer Society. Ces groupes de travail ont donné lieu à la publication de différents standards améliorant continuellement les possibilités d'Ethernet.

Le taux de transfert originel d'Ethernet (IEEE Computer Society, 2002) de 10 Mb/s a évolué selon deux ordres de grandeurs. Tout d'abord en 1995 avec FastEthernet à 100 Mb/s puis en 1998 avec GigaEthernet à 1 Gb/s. Toutefois, la plupart des applications n'ont pas profité de l'augmentation substantielle de performance due à la seule amélioration du taux de transfert. En particulier, les réseaux de bas niveau (composés de racks de modules d'entrées sorties à petits microprocesseurs, de capteurs et actionneurs et autres interfaces) consomment et produisent de petites quantité de données qui sont encapsulées dans des trames Ethernet de 72 octets (la plus petite longueur de trames). En fait, la performance de ces équipements est davantage limitée par la vitesse des microprocesseurs et firmwares embarqués que par la vitesse de communication sur le réseau. Néanmoins, la tendance à supporter un trafic hétérogène dans les systèmes de communication industriels se confirmant, les applications gourmandes en bande passante telles la vidéo ou la voix sont très attachées à cette augmentation de la bande passante. Cette augmenta-

[1] En épistémologie, le principe du déterminisme renvoie à un ordre des faits dans lequel l'apparition de chaque phénomène est nécessaire lorsque les conditions sont satisfaites et devient donc **prévisible**. Cela suppose donc de la capacité de prédire un résultat

tion de la performance du réseau est également intéressante par rapport aux collisions puisqu'elle conduit à diminuer le temps d'accès au réseau et repousse donc le seuil d'écroulement de celui-ci. Néanmoins, cette augmentation de la bande passante ne permet pas de résoudre définitivement la question du non déterminisme d'Ethernet concernant le temps d'accès au réseau. La seule issue serait soit de résoudre les collisions de façon déterministe ou alors de les éliminer complètement.

L'un des développements technologiques concernant Ethernet est l'apparition de la technologie de commutation. L'intérêt majeur des commutateurs est de casser les domaines de collision en petits ensembles d'équipements voire à un équipement seul. Cela permet ainsi de réduire ou même d'éliminer les collisions. En effet, si la segmentation au sein d'un commutateur est poussée à l'extrême, chaque équipement sera isolé sur son propre segment (sur un port unique du commutateur) et disposera ainsi de la totalité de la bande passante du port. Cette technique est appelée micro-segmentation. Elle permet de résoudre l'une des premières causes de collisions sur un réseau, à savoir l'agrégation de trafic de plusieurs stations. Dans ces conditions, la gestion de cette agrégation est renvoyée au commutateur qui doit être capable de traiter la congestion. Il doit supporter la charge totale offerte aux stations (caractéristique non bloquante), car dans le cas inverse des trames pourraient être perdues. C'est particulièrement vrai dans le cas où plusieurs stations envoient des informations vers une même station. Si les collisions sont limitées, elles peuvent toujours apparaître dans le cas d'émission de trafic simultané par le port du commutateur et la station connectée à ce même port. Ethernet est traditionnellement (notamment dans le cas de l'utilisation de câble coaxial) un système de communication half-duplex : une station est à un instant soit en émission ou soit en réception. Avec la paire torsadée et 10baseT, les paires d'émission et de réception ont été séparées. Avec la micro-segmentation où seulement un équipement est connecté à un port, il n'y a qu'une station qui souhaite accéder à la paire. La station émettra donc ces données *via* la paire de transmission et le port du commutateur *via* la paire de réception. C'est le mode full-duplex. Il n'y a alors plus de concurrence d'accès au medium entre les stations et il n'y a théoriquement plus besoin de l'algorithme de résolution des collisions. Le non-déterministe du à l'accès au medium est ainsi résolu par l'utilisation de commutateurs et la combinaison des technologies micro-segmentation et full-duplex qui est également appelé full-segmentation.

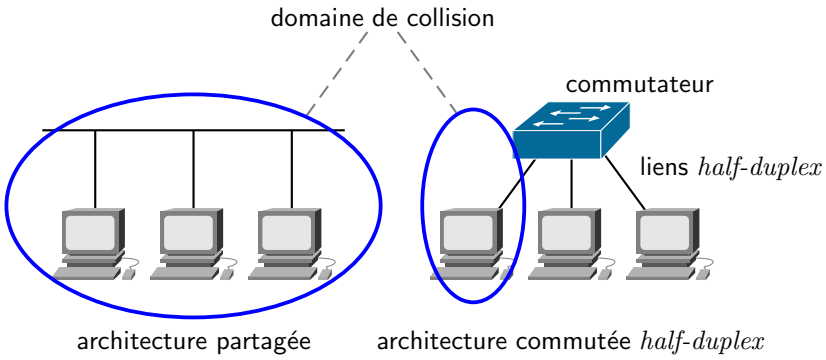


FIG. 2.1: Comparaison des zones de collision d'architectures Ethernet

Le développement des commutateurs a permis d'apporter au réseau Ethernet deux qualités supplémentaires.
Tout d'abord, la notion de différenciation de trafic en classes en utilisant des priorités est apparue avec

le groupe de travail IEEE 802.1p. Ce groupe de travail a donné lieu a une extension de la spécification permettant aux commutateurs de prioritiser le trafic et de filtrer le trafic multicast. Ces travaux sont aujourd'hui directement référencés dans (IEEE, 1998).
Le second point est le contrôle de flux qui permet de limiter la charge à un certain niveau. Ce mécanisme s'apparente au lissage de trafic. Pour de plus amples informations, le lecteur pourra consulter (Alves *et al.*, 2000).

A partir des travaux précédents, les propositions qui vont suivre se concentrent sur l'étude d'un comportement déterministe inhérent aux commutateurs et la technologie full-segmentée. Les travaux suivants s'attardent alors tour à tour à l'analyse des garanties de temps d'accès dans les réseaux Ethernet commutés et à l'utilisation des réseaux Ethernet commutés comme infrastructure de communication pour les systèmes temps-réel, tolérants aux fautes et évolutifs. Une première partie de ces besoins est prise en compte par plusieurs mécanismes implémentés de manière native par les commutateurs. Ainsi, en ce qui concerne les aspects tolérance aux fautes, on peut citer le protocole Spanning Tree (IEEE, 1998) qui permet de concevoir des chemins redondants tout en se protégeant des boucles. De la même façon, l'agrégation de ports permet à un commutateur de regrouper plusieurs liens parallèles et de les traiter comme un seul port. Cela permet aussi de gérer la tolérance aux fautes (si un agrégat est endommagé au niveau d'un lien, le commutateur basculera automatiquement sur les autres liens constitutifs de l'agrégat). Plusieurs travaux visent également à améliorer le temps de réactivité de l'algorithme du Spanning Tree de gestion des chemins de niveaux 2. De la même manière, des réponses technologiques sont disponibles sur les commutateurs permettant de prendre en compte une évolutivité permanente de la topologie et des communications. Par exemple, l'auto-négociation (IEEE Computer Society, 2002) permet à une station et au port du commutateur de s'accorder sur le taux de transfert à utiliser. La notion de pont intelligent (Seifert, 2000) impose au commutateur de maintenir dynamiquement une adresse physique MAC à un port du commutateur ce qui lui permet de prendre en compte une certaine mobilité des stations et l'ajout / suppression de stations. Un point essentiel couvert par l'évolutivité du réseau et de sa dynamique est la définition des Virtual LAN (VLANs) spécifiée dans (IEEE Computer Society, 2003). Avec les VLANs, la flexibilité veut qu'un groupe de stations (groupe de travail, de coopération) ne soit plus défini en fonction du cloisonnement géographique des stations mais par leur coopération. Ceci permet de créer des domaines de diffusion en fonction de l'applicatif et non plus physique.

La figure 2.2 résume les différents développements technologiques d'Ethernet. Elle fait apparaître une évolution entre l'architecture initiale d'Ethernet dans les années 1980 basée sur des hubs aux nouvelles topologies construites à partir des commutateurs définis dans le standard IEEE 802.1D. L'avènement des commutateurs permet de développer des architectures *sans* collisions, mais introduit également des systèmes beaucoup plus complexes ainsi que des temps de latence à chaque traversée d'un commutateur. Il a également donné lieu à de nombreux débats[2] quant à savoir quel type d'Ethernet devait être utilisé pour les applications distribuées temps-réel : Ethernet partagé (ou traditionnel) ou Ethernet commuté. Ces réflexions se sont alors soldées par différentes propositions que nous allons maintenant détailler.

[2]session *Why Moving to Switched-Ethernet ?* lors du 1st Intl. Workshop on Real-Time LANs in the Internet Age

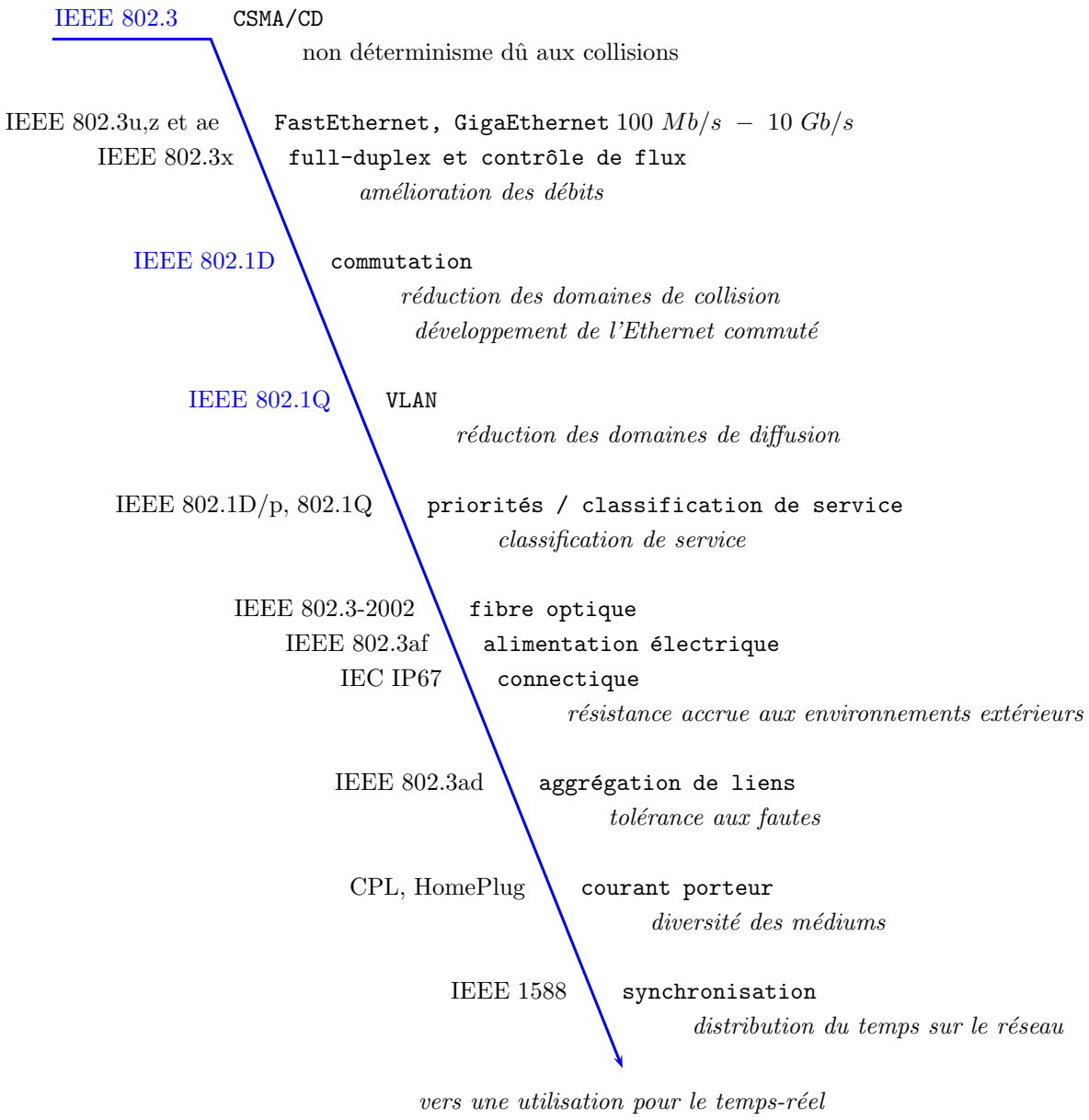


FIG. 2.2: Evolution de la normalisation 'Ethernet'

2.2 Propositions pour l'Ethernet partagé

2.2.1 Modifications de l'accès à la voie

La principale solution mise en œuvre pour garantir un temps d'accès au medium borné sur un réseau Ethernet partagé est l'utilisation de la stratégie Time Division Multiple Access (TDMA), pour laquelle chaque station bénéficie d'un temps d'accès pré alloué. Une seule station dispose du medium à un instant donné, si bien qu'il n'y a plus de collision. Même si cela permet d'obtenir un temps d'accès borné, cela a pour conséquence de rendre le réseau non flexible : une station muette occupe de la bande passante inutilement et l'ajout ou le retrait de nouvelles stations nécessitent l'arrêt provisoire des communications. Contrairement à TDMA qui réserve le medium par rapport aux stations, (Yavatkar *et al.*, 1992) modifient le protocole Ethernet en utilisant un schéma à base de réservations pour le trafic temps-réel. Cet algorithme appelé Predictable Carrier Sense Multiple Access (PCSMA) suppose que tout le trafic temps-réel est périodique. La structure des trames périodiques est alors modifiée de sorte que lorsqu'une collision se produit entre une trame périodique et une trame non temps-réel, l'émission de cette dernière est stoppée. La source périodique continue son émission et finit ainsi par obtenir le medium. Afin de gérer les collisions entre les trames périodiques, chaque source diffuse des annonces précisant les émissions périodiques. Des tables locales permettent alors aux différentes stations sources de messages périodiques de s'accorder sur des réservations d'intervalle de temps pour la transmission de leurs messages. Le trafic restant suit le protocole CSMA/CD.

D'autres stratégies de modification de la méthode d'accès à la voie peuvent également être utilisées. Par exemple, TEMPRA (TimEd Mechanism Packet Release Access (Pritty *et al.*, 1995)) et RETHER (Realtime ETHERnet (Chiueh et Venkatramani C., 1994)) proposent l'utilisation de techniques de passage de jeton (seule la station détenant le jeton peut émettre). Dans RETHER, la station a deux modes de fonctionnement. Tant qu'il n'a pas de besoin temps-réel, le mode d'accès CSMA/CD est préservé. Si une station veut désormais émettre un message temps-réel, elle demande à passer en mode temps-réel et attend un acquittement de toutes les autres stations du réseau. TEMPRA reprend les deux modes de fonctionnement mais l'accès au mode temps-réel est géré par une station moniteur ce qui nécessite un matériel dédié et du logiciel non compatible avec le standard Ethernet.

Le protocole FTT-Ethernet (Pedreiras et Almeida, 2002) se base sur le concept Flexible Time-Triggered (FTT)[3] (FTT a également été implémenté dans le cadre du réseau de terrain Controller Area Network, FTT-CAN (Almeida *et al.*, 2002)). Dans le réseau FTT-Ethernet, le trafic est alloué durant des intervalles de temps à durée fixe appelés cycles élémentaires. Chaque cycle commence avec un message de lancement envoyé par le maître. Le protocole Flexible Time Triggered (FTT) Ethernet (Pedreiras, 2003) utilise la configuration Maître/Esclave afin de supporter les communications fortement temps-réel au vu de la flexibilité et d'une certaine efficacité en terme de bande passante. La principale notion du protocole FTT est l'utilisation de cycle élémentaire qui correspond à un intervalle de temps fixe. L'accès au bus est organisé selon une succession infinie de cycles élémentaires lancés par le maître. Le cycle élémentaire est découpé en deux fenêtres : la fenêtre Synchronous et la fenêtre Asynchronous. La fenêtre synchrone est sujette à un contrôle d'admission et est utilisée pour le trafic temps-réel. La fenêtre asynchrone est utilisée pour les communications événementielles. Le nœud maître joue le rôle d'un coordinateur système. Il main-

[3] http ://www.ieeta.pt/lse/ftt/

tient une base de données comprenant les configurations système, les besoins de chaque communication pour lesquels il établit les cycles élémentaires. Les autres nœuds maintiennent une table d'identification des messages synchrones à produire. A la réception d'un message de début de cycle élémentaire, le nœud esclave décode le message afin d'identifier le message synchrone qu'il doit transmettre. Le message est alors placé en attente dans la fenêtre synchrone. Bien que ce protocole puisse maintenir une synchronisation précise, il induit une surcharge protocolaire considérable. Le modèle maître/esclave suppose une commande centralisée, qui implique un point de rupture unique. Ce protocole doit aborder les questions telles que la tolérance de fautes quand le nœud principal est défaillant. Il peut avoir besoin d'un maître de secours ou d'une méthode d'élection d'un nouveau maître quand le nœud principal tombe en panne.

2.2.2 Algorithmes déterministes pour la gestion des collisions sous CSMA

Jusqu'ici les méthodes étudiées visent à supprimer les collisions. Un autre axe d'étude est de modifier l'algorithme de résolution des collisions afin de rendre son comportement déterministe. Le plus connu est l'algorithme proposé par Le Lann, CSMA/DCR (CSMA avec Deterministic Collision Resolution) (Le Lann et Rivierre, 1993). Comme le montre la figure 2.3, cet algorithme établit un ordre d'accès au medium aux différentes stations suite à une collision. L'ordre est attribué relativement aux adresses des stations (principe dichotomique donnant la priorité aux adresses basses).

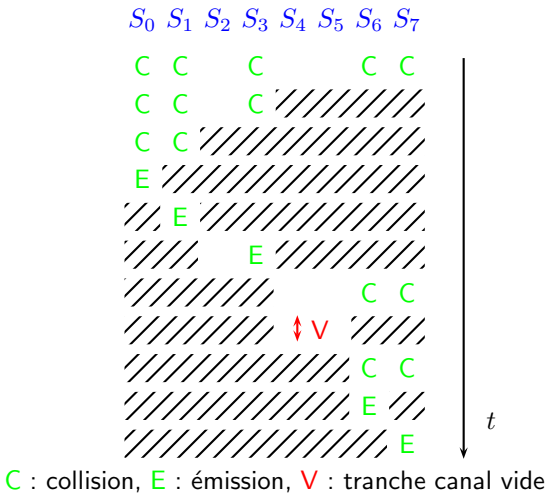


FIG. 2.3: Principe de CSMA/DCR

La figure 2.3 explicite le mécanisme de résolution déterministe des collisions de CSMA/DCR lorsque 8 stations se disputent l'accès au medium. Chaque ligne correspond à un intervalle de temps. Les colonnes permettent de représenter l'activité des stations. Dans le premier intervalle de temps, cinq stations cherchent à émettre une trame, provoquant ainsi l'apparition d'une collision. Pour l'intervalle de temps suivant, le mécanisme CSMA/DCR interdit aux stations $S4$ à $S7$ d'émettre une trame (zone hachurée). Toutefois encore trois stations émettent simultanément une trame et génèrent de nouveau une collision. Pour l'intervalle suivant, CSMA/DCR réduit encore de moitié le nombre de stations autorisées à émettre et ainsi de suite, jusqu'à ce qu'aucune collision n'apparaisse et qu'une trame puisse être émise (E). Si aucune des stations autorisées ne souhaite émettre une trame, le medium reste silencieux. Dans ce cas, une temporisation V permet à l'algorithme de détecter cette inutilisation du réseau et passer ainsi aux

stations suivantes.

Dans ce cadre, l'algorithme stochastique du BEB est remplacé par un algorithme déterministe par arbre de recherche binaire. Quand une collision apparaît, l'arbre de recherche binaire est exécuté. Cet arbre ordonnance les messages à partir de leur adresse MAC. Bien que déterministe, le critère d'ordonnancement peut être insatisfaisant pour les applications temps-réel. Aussi, la variante DOD-CSMA/CD (Deadline Oriented Deterministic CSMA/CD) (Le Lann et Rivierre, 1993) a été proposée. La différence est l'utilisation de classes d'échéance et d'une politique d'ordonnancement EDF (Earliest Deadline First). Une autre variante, CSMA/PDCR (CSMA avec Priority DCR) (Turiel *et al.*, 1996), combine quant à elle PCSMA et CSMA/DCR. A la différence de PCSMA, l'algorithme n'entre en jeu que suite à une collision et utilise comme DCR un arbre de recherche binaire.

2.2.3 Prioritisation de l'accès

Les mécanismes précédents visent à rendre le temps d'accès au medium déterministe. D'autres travaux à portée plus réduite (sans garantie d'un temps d'accès borné) cherchent quant à eux à éviter l'aspect famine, c'est-à-dire à assurer aux trames fortement contraintes d'être privilégiées par rapport à des trames sans contraintes. Cet aspect de l'algorithme BEB étant fortement lié au compteur de collisions, CABEB (Ramakrishnan et Yang, 1994) et BLAM (Molle, 1994) proposent deux nouveaux mécanismes d'accès à la voie. CABEB utilise un compteur de collisions distinct pour chaque station, alors que BLAM utilise un compteur global et définit également une limite du temps d'utilisation du medium pour chaque station.

P-CSMA (Prioritised CSMA) (Tobagi, 1982) s'apparente à une méthode TDMA basée sur la priorité des messages. L'accès au medium est divisé en n slots, n étant le nombre de niveaux de priorité. Chaque message est émis dans l'intervalle de temps correspondant à sa priorité. Cela évite simplement qu'un paquet non prioritaire entre en collision avec un paquet prioritaire.

2.2.4 Contrôle d'admission

Un autre ensemble de travaux vise à contraindre la génération du trafic de manière équitable. Fournir certaines garanties sur la bande passante peut également être atteint en contraignant la génération de trafic par les stations. Les différents travaux abondant en ce sens peuvent être classés en trois familles. L'approche lissage de trafic (traffic smoothing) - (Kweon *et al.*, 1999)(Kweon *et al.*, 2000) - utilise une méthode statistique pour limiter le temps d'accès au réseau en limitant le temps d'inter-arrivées des trames au niveau de la couche MAC. Ainsi, les rafales de trafic non temps-réel sont lissées et étalées dans le temps afin de permettre au trafic temps-réel d'accéder au réseau. Les protocoles à temps virtuel et à fenêtre constituent les deux autres familles. Ces méthodes exécutent une fonction d'ordonnancement à partir d'un état de connaissance globale du réseau pour améliorer l'équité de l'accès au réseau et de réduire le nombre de collisions à partir de critères à priorité.

Ces travaux se focalisent sur le problème du contrôle d'admission dans le but de limiter le temps d'accès au medium d'Ethernet. Plus particulièrement, ils visent à garantir à une trame d'avoir un temps d'accès au medium plus court que par d'autres techniques probabilistes si le taux d'arrivée instantané des trames est limité à un certain niveau plafond, appelé limité d'entrée du réseau (network-wide input limit). Afin de valider cette valeur seuil, (Kweon *et al.*, 1999) utilisent un régulateur de trafic (traffic smoother) placé entre les couches transport et liaison de données d'Ethernet. La régulation se fait à partir d'une

analyse probabiliste de la transmission réussie d'un paquet à chaque tentative. Une modélisation semi-markovienne du protocole 1-persistent CSMA/CD et de l'algorithme BEB d'Ethernet sert à déterminer les valeurs limites de trafic.
Les travaux de Kweon (Kweon *et al.*, 1999)(Kweon *et al.*, 2000) ont donné lieu à d'autres propositions. Dans (Caponetto *et al.*, 2002), l'idée consiste à écarter les paquets par bufferisation de manière à répartir au mieux la charge des commutateurs et à limiter la génération de trafic. Les paramètres pris en compte pour la régulation dépendent de la bande passante totale et du nombre de collisions observé durant un intervalle de temps. La loi de commande de lissage utilise un *fuzzy controller*. Le seuil d'acceptation de bande passante tolérable par le réseau est partagé en seuils de bande passante pour chaque station. Cette limite est calculée à partir de l'échéance des trames et du niveau acceptable de perte de messages. Le lissage de trafic n'intervient que sur les paquets non temps-réel afin de lisser les rafales (bursts) de ce trafic. L'analyse repose sur le fait que quand les messages arrivent en rafale, ils ont plus de chance de rentrer en collision. L'implantation du régulateur de trafic retenu utilise un algorithme basé sur le seau percé (Cruz, 1991*a*). Compte tenu du comportement complexe et non linéaire du réseau Ethernet, (Caponetto *et al.*, 2002) proposent d'utiliser un contrôleur à apprentissage dans lequel différentes règles d'observation de l'état du réseau commandent différents niveaux de régulation.

Virtual Time Protocol (Zhao et Ramamritham, 1987) implémentent un mécanisme de rétention des trames (Packet Release Delay) qui est fonction d'un paramètre tel que le relâchement (laxity) ou à priorité. Le temps de rétention est proportionnel à ce paramètre, originellement la date d'arrivée de la trame ou le relâchement comme proposé par (Zhao et Ramamritham, 1986). Il y a encore beaucoup d'autres travaux dans ce domaine. Le lecteur intéressé pourra notamment consulter (Hanssen et Jansen, 2003). Dans les protocoles à fenêtres, l'émission d'un message n'est autorisée que s'il arrive dans une fenêtre de temps ou de priorité. La taille de la fenêtre est dynamiquement actualisée en fonction de l'état du réseau (libre, occupé, collision). Différentes politiques d'arbitrage de l'accès peuvent alors être utilisées tel que l'ordre d'arrivé FIFO (Gallager, 1978), la contrainte de rétention (laxity) (Minimum Laxity First par exemple), l'échéance du message, ou encore la priorité du message (voir (Hanssen et Jansen, 2003)).

2.3 Propositions pour l'Ethernet commuté

Nous présentons dans ce nouveau paragraphe un panorama des travaux développés pour les architectures Ethernet commuté visant à améliorer leurs performances, voire à les rendre déterministes.

2.3.1 Problème de la diffusion multicast

Tout d'abord, dans les environnements commutés, les communications de type un à plusieurs posent le problème de l'augmentation de la charge. En effet, un commutateur doit relayer toute trame de diffusion (broadcast) sur tous ses ports. Une solution mise en avant dans (Modlovansky, 1998) est l'utilisation des VLANs (IEEE Computer Society, 2003). L'utilisation des VLANs permet de segmenter virtuellement le réseau et contenir une trame en diffusion émise par une station appartement à un VLAN donné aux seules autres stations de ce même VLAN et non plus à la totalité des stations présentes sur le réseau comme le montre la figure 2.4.

De plus, il est proposé de réduire la charge des commutateurs en utilisant le filtrage du trafic multi-

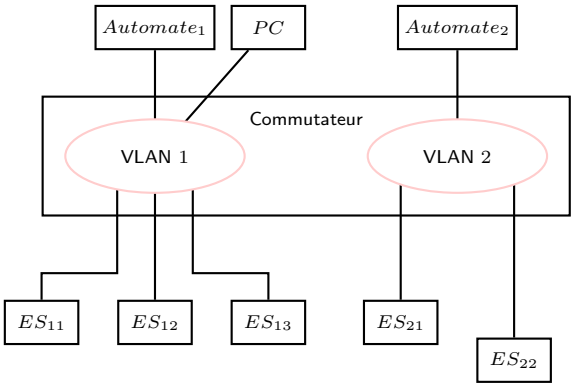


FIG. 2.4: Isolation des équipements par la technologie VLAN

cast (IGMP snooping). Seuls les équipements concernés seront abonnés et recevront le trafic multicast. L'inondation du trafic multicast dépend ainsi d'une configuration dynamique des ports du commutateur si bien que seuls les ports abonnés au groupe multicast recevront le message.

2.3.2 Le service offert aux messages à temps-critique

Un autre axe d'étude concerne l'amélioration du service offert par les architectures Ethernet commuté aux données à temps-critique. Le standard Ethernet n'offre pas de mécanismes de Qualité de Service et donc aucune garantie à ce type de trafic. Un certain nombre de travaux propose alors l'implémentation de protocoles de couche haute offrant la possibilité de réserver des ressources dans les commutateurs.

Commutateurs de niveau 3 à Qualité de Service

(Varadarajan et Chiueh, 1998) décrivent le développement d'un commutateur Fast Ethernet temps-réel nommé EtherReal offrant des garanties de bande passante au trafic temps-réel à partir du moment où il respecte une contrainte d'arrivée de type CBR (Constant Bit Rate). Le point clé mis en avant est de ne pas engendrer de modification matérielle des nœuds terminaux. Néanmoins, ce projet reste fortement orienté sur la bande passante. Ceci n'est pas complètement suffisant compte tenu des besoins des communications temps-réel fortement liées au délai par exemple.

L'approche d'Hoang (Hoang, 2004) (Hoang et Jonsson, 2003) (Hoang *et al.*, 2002) vise également à étendre la notion de réservation de bande passante d'un commutateur. Plus particulièrement, Ethernet commuté est étendu de façon à autoriser le trafic temps-réel périodique en utilisant l'ordonnancement Earliest Deadline First (EDF (Liu et Layland, 1973)) (Hoang *et al.*, 2002). Le mécanisme retenu fait intervenir une couche supplémentaire dans le modèle OSI intercalée entre les couches 2 et 3 comme le montre la figure 2.5 .

Les communications de type Internet sont supportées tant qu'elles ne remettent pas en cause les besoins temps-réel des autres communications. La couche temps-réel met en place des canaux temps-réel, chacun étant une connexion virtuelle entre deux nœuds du système. Des trames de synchronisation sont émises par le commutateur afin d'assurer la cohérence temporelle. Une phase de négociation établie *via* le canal temps-réel (figure 2.5) permet au commutateur de définir la pile à utiliser pour le trafic (à échéance pour le temps réel, FIFO sinon). Notons enfin la modification de la signification de l'entête IP. Les 8

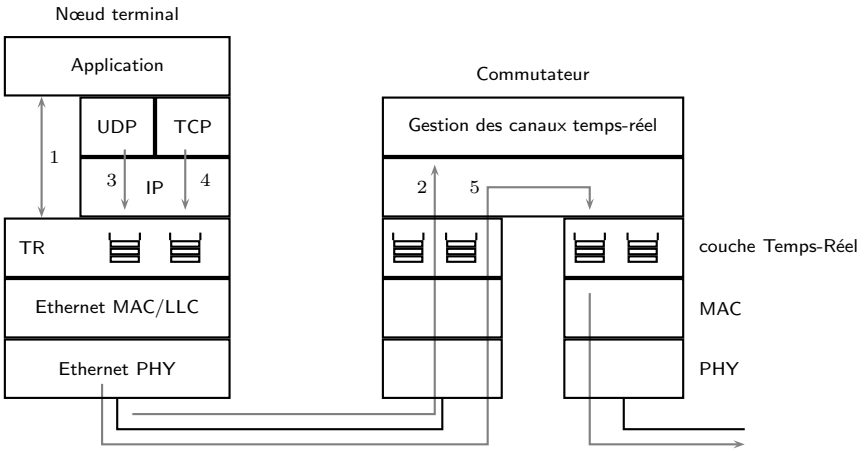


FIG. 2.5: La couche temps-réel

octets servant initialement à coder les adresses IP source et destination, sont utilisés ici pour l'échéance (48 bits) et l'identifiant du canal temps-réel (16 bits). La répartition du service entre les différents canaux se fait à partir d'un schéma de partitionnement asymétrique des échéances (Hoang et Jonsson, 2003).

Ces travaux s'inscrivent plus généralement dans le contexte de la Qualité de Service offerte dans les systèmes temps-réel distribués et peuvent ainsi être agrémenté d'autres politiques d'ordonnancement. Dans ce cadre, les travaux de (Song, 2004) proposent d'utiliser la théorie $(m, k) - firm$ qui permet de garantir un seuil minimal de ressources. Ces travaux sont destinés à être appliqué au transfert de vidéos pour lesquels l'écartement de certains types d'images peut être acceptable. L'idée sous-jacente est que cette politique pourrait utiliser par d'autres applications temps-réel où l'écartement de données non critiques est tolérable.

Dans la mesure où la Qualité de Service s'attache à garantir différents niveaux de ressources au trafic entrant sur le réseau, elle nécessite au préalable que ce trafic respecte également différentes règles. En fait, il convient que l'arrivée des données soit limitée au moyen de régulateur de trafic. Le contrôle d'admission a par conséquent été étudié pour les environnements commutés comme pour l'Ethernet partagé, et ce que ce soit dans le cadre de la réservation de ressources ou afin de limiter différents phénomènes comme les rafales par ailleurs. Ainsi, les travaux précédents (Varadarajan et Chiueh, 1998), (Hoang, 2004) supposent déjà que l'arrivée des données est régulée par différentes politiques. Cette limitation du trafic entrant se retrouve dans d'autres domaines, comme les réseaux avioniques embarqués. Dans (Grieu *et al.*, 2003), les stations doivent au besoin ralentir leur émission, par un dispositif de type seau percé. Les commutateurs ont également pour charge de faire respecter ce contrat en supprimant les trames qui le violeraient par l'intermédiaire de dispositifs de contrôle de flux dans chaque commutateur. Ces contrôles sont effectués sur une base individuelle (pour chaque flux).

La Classification de Service

Au niveau d'Ethernet, il est à noter que l'on ne parle pas de qualité de service mais simplement de classes de service. Cette différence suggère déjà le fait que l'on ne va pas chercher à satisfaire des besoins prédéfinis mais plutôt à privilégier le traitement de certaines trames que d'autres sans pour autant garantir

un niveau de service. Dans ce cadre, cela sous entend également une première étape qu'est la prioritisation du trafic. Phase qui par nature sera arbitraire.

Comme il a déjà été indiqué, il y a une différence fondamentale entre les concepts de Classe de Service (CdS) et de Qualité de Service (QdS). La QdS se définit par la garantie que donne le réseau à une application en termes d'un contrat de services au travers de la session de l'application. L'application requiert explicitement ou implicitement (*via* un gestionnaire de politique) un niveau de service d'utilisation du réseau. Le réseau réserve alors les ressources (mémoires, canaux ...) appropriées pour la durée de la session applicative. La QdS implique une connexion et une réservation des ressources réseau de façon à fournir une garantie d'un niveau de service minimum. La réservation est effectuée au moyen de protocoles tels que RSVP (IETF Intserv working group, 1997).

La CdS est plus simple. Le réseau fournit un niveau de service supérieur pour les applications prioritaires, mais ne garantit explicitement rien d'autre. Il n'y a aucune garantie d'un service minimum. Ni la mémoire ni les canaux n'ont besoin d'être réservés. Il n'y a pas besoin d'autres protocoles. Cette approche suppose que la capacité du réseau est en moyenne suffisante pour l'ensemble des applications, mais que temporairement des phénomènes de congestion peuvent apparaître. Une justification possible est qu'il est plus simple et par conséquent moins cher de surestimer quelque peu les capacités du réseau et d'utiliser la CdS pour améliorer les surcharges spécifiques que d'invoquer les mécanismes complexes de QdS.

Nativement, Ethernet n'implémente pas de mécanisme de priorité. Les priorités vont intervenir lors de la gestion de la congestion d'un commutateur, gestion qui est réalisée au travers de l'implémentation de buffers. Pour les flux les plus importants (plus sensible aux délais), un traitement préférentiel sera accordé au vu du niveau d'importance codé à travers le niveau de priorité. Deux méthodes de prioritisation pourront être considérées :

- d'accès, dans ce cas le niveau de priorité est fixé au niveau de l'accès au réseau pour un équipement ;
- d'utilisateur, la priorité est définie par l'application pour un ensemble de trames ou de façon spécifique, pour une trame.

Mécanismes de priorité pour Ethernet Dans son standard (IEEE Computer Society, 2002), Ethernet ne présente pas de mécanismes natifs de support de priorités de trafic. Ceci parce qu'il a été développé dans une recherche constante de simplicité et d'égalité (ou en anglais, fairness (Digital Equipment *et al.*, 1980)). Néanmoins, on trouve des solutions propriétaires non standard d'insertion de priorités dans Ethernet. Ainsi, au niveau des priorités d'accès, les trois principales méthodes sont :

- adapter le trou inter-trame (plus petit pour les trames prioritaires),
- modifier l'algorithme du BEB (temps d'attente non calculé aléatoirement, mais défini en fonction de la priorité, voir le paragraphe 2.2.2),
- jouer sur la longueur du préambule pour que l'équipement le plus important émette un préambule plus long avant d'émettre ses informations de sorte que les autres stations détectent une occupation du medium et se taisent.

Pour ce type de prioritisation, si le format de la trame Ethernet reste inchangé, il est néanmoins nécessaire de modifier l'algorithme d'accès à la voie. Puisque la trame Ethernet ne permet pas d'indiquer qu'il s'agit d'une trame critique, un constructeur a proposé de modifier l'usage du bit d'administration des adresses (global ou local) dans la mesure où il jugeait que ce bit était la plupart du temps positionné

à 0 (adresses globales). Dans cette technique, (Wang, 1996) propose qu'un '0' code un niveau de priorité classique et un '1' une trame importante. Toutefois, cette technique n'est pas utilisée puisqu'elle requiert la non utilisation des plans d'adressage local et la compréhension réciproque du nouveau format de trames sans parler des besoins d'homogénéité du parc et des problèmes lors de la résolution d'adresse.

Compte tenu du manque de standardisation de ces solutions, un projet nommé P802.1p fut lancé en 1995 afin de traiter initialement les problèmes de prioritisation et d'utilisation du multicast pour l'acheminement de la voix. Parallèlement, un second projet nommé P802.1Q s'est attachée aux problèmes de VLAN à l'intérieur des commutateurs. Les intérêts communs de ces deux groupes et le besoin plus global de mécanismes de prioritisation pour Ethernet amenèrent alors à intégrer le projet P802.1 au standard existant IEEE 802.1D (IEEE, 1998) et au développement parallèle de IEEE 802.1Q. La standardisation de la classification de service couvre actuellement ces deux standards. 802.1D/p précise les questions de priorité, à savoir la détermination, la regénération et l'association des ressources mémoires entre les différentes classes de service. 802.1Q inclut quant à lui l'étiquetage de la priorité d'une trame dans l'étiquette VLAN comme le montre la figure 2.6.

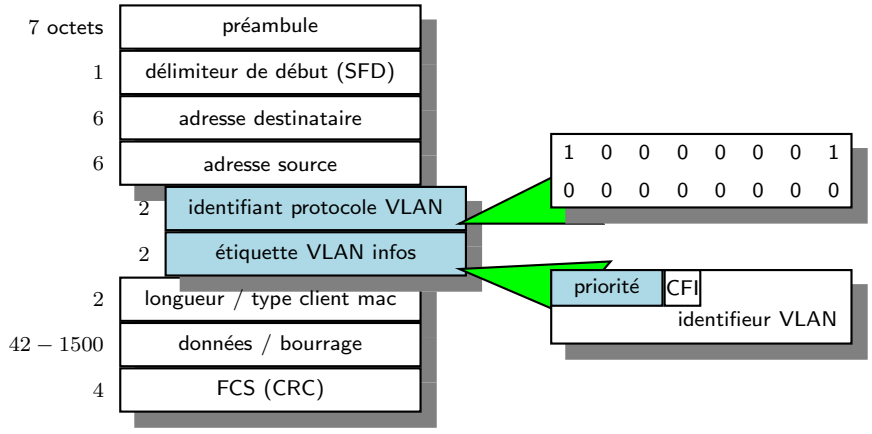


FIG. 2.6: Fomat de la trame Ethernet étiquettée

Priorités & classes de service Le nombre de niveaux de priorité et de classe de service supportée a été historiquement défini par les standards, même si en pratique, ils sont rarement tous utilisés. Le standard IEEE 802.1p définit huit niveaux de priorité et leur équivalence applicative tel que reporté dans la table 2.1.

La plupart des implémentations utilisent deux niveaux ou trois niveaux. La difficulté d'implémenter l'ensemble des huit niveaux provient de la complexité des structures de mémoires, particulièrement lorsque la commutation est réalisée *via* le matériel. Le trafic à charge contrôlée se réfère aux applications présentant un minimum de besoin en bande passante sans pour autant présenter une réelle sensibilité par rapport aux délais et à la gigue. Le trafic par défaut est considéré plus important que les tâches de fond de type sauvegardes initiées par les serveurs de fichiers ; il correspond néanmoins à la plus petite valeur, à savoir 0.

Chaque classe de service identifie un ensemble de trafic pour lequel le commutateur fournira un niveau de performance particulier. Typiquement, chaque classe de service est associée à une file d'attente distincte

TAB. 2.1: Recommandations IEEE 802.1p pour l'affectation des priorités

priorité	type
7	gestion de réseau
6	voix
5	vidéo
4	à charge contrôlée
3	excellent effort
0 (par défaut)	best-effort
2	épar
1 (le plus bas)	en tâche de fond

pour chaque port de sortie comme le montre la figure 2.7. Le standard 802.1p prévoit enfin que le nombre de classes de service puisse être différent du nombre de niveaux de priorité.

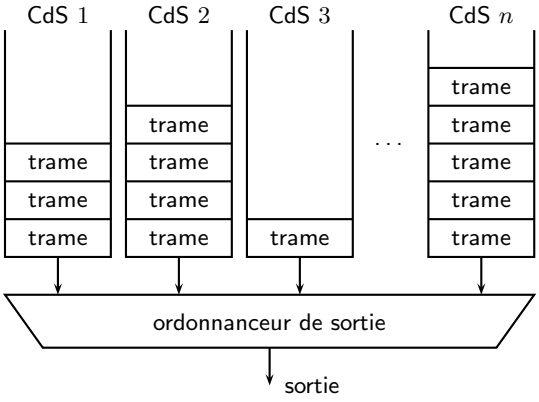


FIG. 2.7: Instanciation d'une file d'attente pour chaque classe de service au niveau d'un port de sortie d'un commutateur

L'instanciation de plusieurs files en sortie d'un commutateur nécessite d'adopter une stratégie d'ordonnancement. Plusieurs politiques sont possibles, nous les détaillerons au paragraphe 4.3.

Apports L'implémentation de cette technologie se justifiera donc dans les cas suivants :

- on cherche à minimiser des bornes temporelles délivrées par le réseau qui ne satisfont pas les besoins applicatifs,
- on veut privilégier des trafics lorsque temporairement il existe des points de congestion du réseau pour lesquels la bande passante disponible est insuffisante.

Néanmoins, pour pouvoir utiliser ce mécanisme de priorité, cela suppose notamment :

- que les APIs des protocoles soient modifiées pour supporter les niveaux de priorités,
- que de multiples files soient crées lors de l'implémentation des protocoles,
- que le logiciel des interfaces réseaux soit modifié pour gérer la priorité et le tag (nouveau format de trame),
- que des algorithmes d'ordonnancement des files soient implémentés.

On note donc un nombre important de modifications à apporter, et cela indépendamment du fait que ce seront les seuls commutateurs qui seront chargés de différencier le traitement à apporter aux trames.

En résumé, si la Classification de Service ne permet pas de rendre Ethernet déterministe, elle contribue à la réduction des délais subis par les informations critiques. Cet apport a été ainsi illustré dans différents travaux comme (Jasperneite *et al.*, 2002), (Georges *et al.*, 2004*c*) ou (Grieu, 2004) qui présente une optimisation des différents paramètres de la CdS.

2.3.3 Synchronisation des horloges

Une question importante soulevée par les systèmes contrôlés en réseau se porte sur la synchronisation des horloges des différents équipements distribués sur le réseau. Ethernet ne propose pas nativement un moyen d'assurer la synchronisation des horloges des équipements et des entités interconnectées au réseau. Toutefois, différents travaux se sont portés sur cette problématique. Le premier protocole proposé, NTP (Network Time Protocol) (Mills, 1991), définit ainsi un cadre de synchronisation d'horloges. Les fortes contraintes des applications distribuées temps-réel ont toutefois conduit au développement d'un nouveau protocole. Le standard IEEE 1588 (IEEE, 2002) présente *un protocole pour synchroniser des horloges indépendantes de nœuds d'un système de contrôle / mesure distribué avec une grande précision.* Dans ce protocole, tous les nœuds (niveau instrumentation compris) possèdent une horloge IEEE 1588 qui est synchronisée avec tous les autres acteurs *via* le protocole PTP (Precision Time Protocol). Les équipements du niveau instrumentation tels les capteurs peuvent ainsi estampiller leurs données et les actionneurs peuvent fonctionner en suivant un référentiel horaire identique. L'homogénéité de l'horloge dépend alors de la synchronisation des horloges temps réel locales.
Deux types d'horloges sont utilisées : commune (ordinary) et frontière (boundary). Les horloges de frontière sont utilisées par les équipements tels les commutateurs tandis que les horloges communes sont employées dans les équipements à un seul port tels les capteurs. Chaque interface (port) implémentant une horloge de frontière peut soit agir comme maître ou horloge ordinaire sur son segment. PTP nécessite le support du multicast, mais aussi le confinement des communications multicast sur un même sous-réseau où chaque horloge locale satisfait exactement aux recommandations. Une horloge de référence (grandmaster clock GMC) est choisie suivant sa stabilité et sa précision. Cette horloge est unique pour un système et il n'y a qu'un maître par sous-réseau. Tout comme le protocole NTP, PTP se base sur modèle de transaction $T_1 - T_4$ comme le montre la figure 2.8. En fait, le processus de synchronisation nécessite deux étapes : les mesures de l'offset et du temps de propagation.

La mesure de l'offset correspond à la différence entre les horloges du maître et de l'esclave. Pour cette correction, le maître transmet périodiquement un message *Sync_Msg* qui contient une valeur estimée (a_0) de la date réelle (T_1) d'émission du message. Dans le message *Follow_up*, le maître retourne la valeur mesurée plus précise de la date d'émission du message précédent.
Dans la seconde étape, la mesure du délai permet de déterminer le temps de propagation entre le maître et l'esclave. Pour cela, l'esclave émet un message *Delay_Req* auquel le maître répond par un message *Delay_Resp*. Comme le montre la figure 2.8, cet échange permet de mesurer précisément le temps de propagation (une fois l'offset corrigé). L'esclave peut alors calculer les informations nécessaires à l'ajustement de son horloge. Les messages de synchronisation sont confinés au sous-réseau. La résolution de l'heure système incombe au GMC qui peut être également synchronisée par une source externe (par GPS par

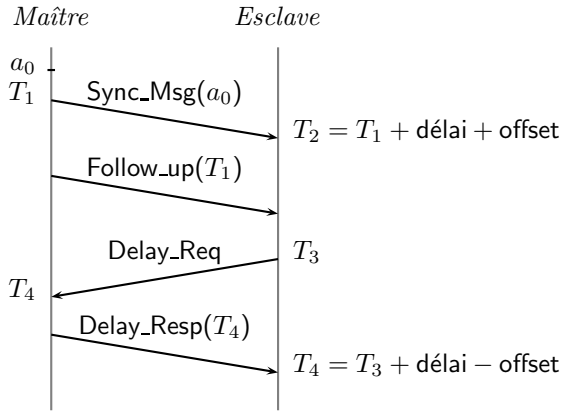


FIG. 2.8: Transactions liées au processus de synchronisation de PTP

exemple).

L'annexe D du standard (IEEE, 2002) présente une implémentation de PTP pour Ethernet. Ainsi, même si IEEE 1588 ne modifie pas Ethernet et ne le rend pas plus déterministe ou sûr, il fournit une méthode aux autres protocoles pour y arriver. Un système de synchronisation performant pour des nœuds distribués pourrait se coupler avec une application de contrôle de trafic afin de délivrer une implémentation déterministe d'Ethernet (précision sous la microseconde). En plus des besoins en temps-réel, il est alors important de noter les aspects de flexibilité et d'interopérabilité que nécessite l'utilisation d'un tel protocole. Cette fonctionnalité est par ailleurs de plus en plus implémentée dans les produits proposés par les constructeurs comme le commutateur RS2-16M de Hirschmann Electronics Ltd.

Différentes adaptations[4] du standard IEEE 1588 ont été proposées pour son utilisation dans les réseaux Ethernet commutés. Par exemple, (Jasperneite *et al.*, 2004) notent que le mécanisme d'horloge de frontière qui est implémenté dans les commutateurs et qui nécessite une boucle de contrôle de l'horloge locale, peut conduire à une instabilité et une déviation des horloges distribuées dans les environnements commutés linéaires. Dans ce type d'architecture où il peut y avoir plusieurs dizaines de commutateurs entre l'horloge de référence et l'horloge client, autant de niveaux de synchronisation (boucles de régulation) interviennent. Les travaux développés par Jasperneite visent à synchroniser le client directement par l'horloge de référence. Pour cela, les commutateurs intermédiaires laissent passer le message tout en compensant le délai. Ils ajoutent en fait le délai de traversée du commutateur et le délai de propagation sur le lien amont au niveau du message de synchronisation (bypass clock).
(Höller *et al.*, 2003) présentent une méthode pour l'utilisation de la synchronisation d'horloges basées sur la technologie SynUTC et le standard IEEE 1588 dans les réseaux Ethernet industriels. Cet article présente ainsi des systèmes prototypes d'interface réseau et de commutateur implémentant la synchronisation d'horloges.
Les travaux de (Gaderer *et al.*, 2005) visent quant à eux à améliorer la tolérance aux fautes du protocole IEEE 1588. En effet, le point le plus crucial de ce protocole s'articule autour du principe maître / esclave. Si le maître s'arrête (ne répond plus ou ne transmet plus les messages de synchronisation), le standard prévoit l'élection d'un nouveau maître. Néanmoins, le standard impose que cette nouvelle élection ne démarre qu'après 10 périodes de synchronisation manquantes (période assez longue pouvant

[4]http ://ieee1588.nist.gov/

conduire à une désynchronisation des horloges locales). De plus dans les réseaux Ethernet commutés, il est nécessaire de considérer la rupture d'un commutateur. Pour cela, une solution hybride comprenant un ensemble d'horloges de référence démocratiquement synchronisé est utilisé au lieu d'un seul maître. Dans le cas d'une application du standard IEEE 1588, (Gaderer *et al.*, 2005) proposent une architecture de commutateur modifiée prenant en compte des éléments de redondance (double alimentation) et l'utilisation d'une topologie en double anneaux afin de compenser une rupture de liens.

2.3.4 Optimisation de la topologie

Le point d'étude soulevé par (Höller *et al.*, 2003) concernant la topologie du réseau Ethernet commuté la plus adaptée a donné lieu à un ensemble de travaux en particulier au CRAN. Nous avons ainsi précédemment défini la nécessité d'utiliser les commutateurs lorsque les besoins en temps-réel apparaissent sur un réseau Ethernet. Le tout est de savoir quelle architecture d'interconnexion des équipements terminaux et commutateurs doit être mise en œuvre. Il existe trois grandes topologies : la structure linéaire, la structure en anneau et la structure dite hiérarchique.

Le premier axe des travaux (Jasperneite et Neumann, 2001) consiste à simuler plusieurs *scenarii* de communications industrielles. Dans (Jasperneite et Neumann, 2001), l'outil de simulation réseau OPNET 7.0 est utilisé pour simuler le comportement des topologies linéaire et hiérarchique (ou en étoile). Les simulations prennent en compte différents paramètres (type de service, priorité, volume des données ...) et confrontent différentes qualités du réseau. (Jasperneite et Neumann, 2001) concluent que la topologie hiérarchique fournit de meilleures garanties que la topologie linéaire, mais que cette dernière reste toutefois suffisante pour un certain nombre d'applications industrielles. Outre la simulation, d'autres travaux visent une évaluation théorique de la performance du réseau. (Rüping *et al.*, 1999) présentent une étude comparative des topologies utilisées dans les réseaux industriels Ethernet commutés. L'étude se focalise autour d'une architecture type comprenant un nœud maître et n équipements de terrain. Les communications sont de type cyclique. Les connexions sont de type point à point (réseau micro-segmenté, liens full-duplex) et (Rüping *et al.*, 1999) établissent une analyse des trois types de topologies en comparant la durée minimale du cycle. Le temps cycle est obtenu en prenant en compte les temps de transmission et un temps de latence de commutateur de 2 μs. Les résultats présentés dans (Rüping *et al.*, 1999) conduisent à privilégier la topologie hiérarchique, voire en anneau par rapport à la topologie linéaire. Toutefois, cette étude ne porte que sur un scénario de communication bien particulier et l'analyse des temps de communication reste relativement simpliste. D'autres travaux sont venus étendre cette évaluation de performance d'une architecture donnée dans le but d'optimiser la topologie qui serait par la suite utilisée. Dans un premier temps, (Torab, 2000) propose une solution systématique utilisant une représentation sous forme de graphe qui permet d'estimer la charge des commutateurs présents dans les architectures hiérarchiques et linéaires. Etant donné le volume et les chemins du trafic des nœuds terminaux, cela revient à calculer la quantité de trafic (la charge) sur les équipements internes du réseau (les commutateurs). Le détail de la technique de détermination de la charge des commutateurs est donné dans (Krommenacker, 2002). Cette méthode montre que l'analyse et la conception de réseau commuté pour les systèmes de contrôle est alors proche du problème d'analyse de charge (Torab et Kamen, 1999). A partir de là, (Kamen *et al.*, 1999) proposent alors d'utiliser un système à file d'attente $M/D/1$ pour calculer les temps d'acheminement des messages de bout en bout. L'arrivée de trafic sur les équipements terminaux est supposée de type Poisson,

la taille des messages est supposée fixe et la politique de bufférisation des commutateurs est le 'store & forward'. Le délai moyen d'attente dans la file d'un commutateur est alors équivalent à $W = \frac{\lambda T^2}{2(1-\lambda T)}$ où λ est le taux d'arrivée moyen et T le temps de service tel que $\lambda < \lambda_{max} = \frac{1}{T}$. Le temps d'acheminement entre deux nœuds de communication i et j est obtenu en sommant les temps d'attente dans les files de chaque commutateur traversé et les temps de transmission. (Kamen *et al.*, 1999) proposent ainsi :

$$D_{ij} = \sum_{k \text{ commutateurs traversées}} W_k + \sum_{k \text{ transmissions}} T_k$$

(Krommenacker, 2002) note alors que les travaux présentés précédemment posent deux problèmes. Dans un premier temps, ils permettent simplement de se faire une idée sur la topologie la plus adaptée, mais ne présentent pas de méthode de conception élaborée indiquant comment interconnecter les équipements terminaux aux commutateurs. Et dans un second temps, les architectures étudiées ne présentent pas de mécanismes de fiabilisation ou de tolérance aux fautes tels que la redondance de chemins.

Concernant ce dernier point, une analyse de la criticité des architectures est présentée dans (Krommenacker, 2002). Krommenacker propose alors une topologie hybride mêlant topologie hiérarchique et linéaire. Cette nouvelle topologie présente alors l'intérêt de fiabiliser l'architecture tout en limitant le coût de la redondance comparé à une architecture à fiabilisation totale. Le problème de l'optimisation de la conception d'une architecture Ethernet commuté a fait l'objet de plusieurs travaux au CRAN (Rondeau *et al.*, 2001)(Krommenacker, 2002). L'objectif de ces travaux est de minimiser le délai d'acheminement des messages dès la phase de conception de la topologie.

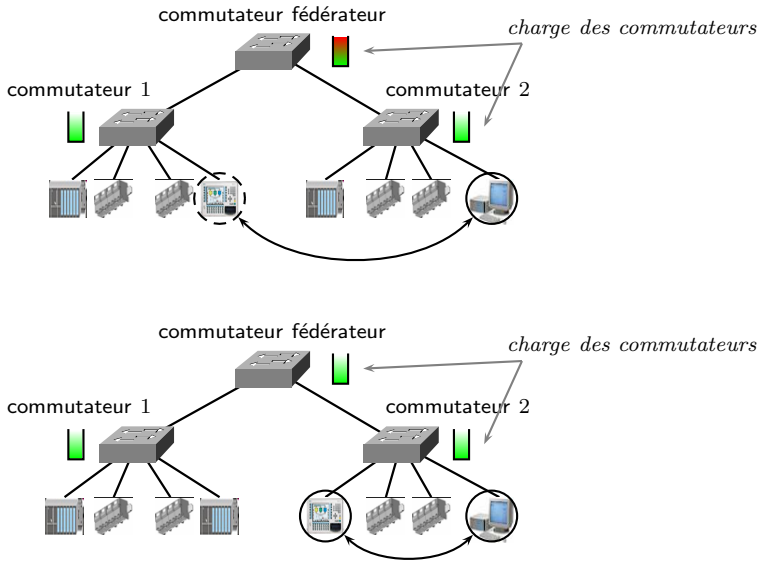


FIG. 2.9: La charge des commutateurs dépend de la distribution des équipements de contrôle sur les commutateurs

L'analyse retenue montre alors que ce délai dépend notamment des deux critères mis en évidence sur la figure 2.9 : le nombre de commutateurs traversés ainsi que l'équilibre de la charge dans chaque commutateur. Comme le montre la figure 2.9, le problème posé correspond à la distribution des équipements sur les commutateurs adéquats. L'originalité des travaux est de présenter la conception d'un réseau commuté

comme un problème de partitionnement de graphes. La résolution de ce problème d'optimisation a été traité en utilisant les algorithmes spectraux (Rondeau, 2001)(Rondeau *et al.*, 2001)(Adoud, 2002) et les algorithmes génétiques (Krommenacker *et al.*, 2001)(Krommenacker *et al.*, 2002*a*). Ces travaux montrent qu'en minimisant les échanges inter-commutateurs (c'est-à-dire en regroupant les équipements terminaux selon leur niveau de connexité), les délais sont minimisés et les échéances temps-réel d'autant plus satisfaites. Nous verrons par la suite que ces travaux ont été complétés en spécifiant la satisfaction des bornes temporelles comme critère d'optimisation (Georges *et al.*, 2004*b*).

2.3.5 Evaluation de performances

L'analyse de performances de l'utilisation d'un réseau Ethernet commuté pour les applications temps-réel est relativement récente. (Rüping *et al.*, 1999) présentent une méthode d'analyse des temps pour différentes topologies. Parmi les hypothèses utilisées, on notera l'utilisation d'un nœud de contrôle (sorte de maître) et le fonctionnement en mode cyclique. L'article explique l'intérêt de la topologie hiérarchique. (Torab et Kamen, 1999) développent une analyse de la charge des équipements du réseau (en particulier les commutateurs). Dans ces travaux, la modélisation des communications s'attache à un modèle du réseau par graphes et les communications sont représentées par une matrice de trafic. Une fois la charge connue, une estimation simpliste des temps de traversée du réseau est donnée.

Outre (Rüping *et al.*, 1999) et (Kamen *et al.*, 1999), (Song, 2001) propose une autre étude des délais de traversée du réseau. Le calcul présenté vise à établir le temps de bufferisation dans un commutateur pour des entrées périodiques. Ces travaux sont établis pour un ensemble de sources de trames périodiques $\{C_i, T_i\}$ avec $i = 1, 2, \ldots, N$ représentant la priorité dans l'ordre décroissant (1 est plus prioritaire que 2), C_i le temps de transmission (longueur de la trame divisé par le débit du lien) et T_i le temps d'inter-arrivée (période entre deux trames). Le commutateur considéré prend en compte un ordonnancement à priorité fixe et le calcul vise à établir le pire temps d'attente. Selon (Song, 2001), ce temps correspond pour une trame de priorité i au temps nécessaire de traitement de toutes les trames de priorité supérieure ou égale à celle de la trame considérée arrivée au même instant, plus le temps d'émission B_i de la trame la plus longue de priorité inférieure ($B_i = \max_{j>i} C_j$). Ainsi, le pire temps de réponse d'une trame avec la priorité i est donné par

$$R_i = C_i + I_i$$

avec I_i le temps d'interférence tel que

$$I_i^{n+1} = B_i + \sum_{j \leq i} \left\lceil \frac{I_i^n}{T_j} \right\rceil C_j$$

qui peut être calculé en utilisant la méthode classique du point fixe ($I_i^0 = 0$ jusqu'à la convergence $I_i^n = I_i^{n+1}$).
(Song, 2001) présente également un calcul du délai de bufferisation pour des entrées apériodiques. Toutefois, le calcul est soumis aux hypothèses suivantes : toutes les trames sont de longueur identique, le flux d'arrivée de chaque port d'entrée suit un process de Bernoulli indépendant et identique de p trames par intervalle et chaque trame a une probabilité de $1/N$ d'être destinée à un port de sortie donné.

2.4 Analyse des propositions

L'impératif d'améliorer les performances d'Ethernet, voire de le rendre déterministe pour les applications distribuées à fortes contraintes temporelles justifie alors de comparer les différentes approches présentées dans ce chapitre. Cette quête d'un comportement temps-réel conduit à modifier la normalisation vers un certain déterminisme ou de façon plus limitée à une amélioration du service offert par ce type de réseau. Le tableau 2.2 compare ainsi les différentes propositions en fonction de différents indicateurs de qualité mais aussi de facilité d'utilisation dans le contexte industriel.

Le premier élément de comparaison de ces différents travaux est le respect de la normalisation (nous considérons ici aussi bien les standards IEEE 802.3, 802.1D, 802.1Q). On note alors que la majorité des travaux s'appuie alors sur une modification ou une extension de la normalisation. Ces travaux requièrent alors tour à tour de profondes modifications (l'implémentation de CSMA/DCR nécessite de modifier le firmware des cartes réseaux) ou des extensions logicielles. Ainsi, certaines propositions comme FTT-ETHERNET, (Hoang, 2004), ou encore le lissage du trafic proposent l'ajout d'une nouvelle couche protocolaire assurant un contrôle d'admission du trafic en entrée du réseau.

Les deux indicateurs suivants comparent la portée des travaux par rapport à l'objectif d'utiliser Ethernet dans les applications à temps critique. Soit la proposition apporte un certain déterminisme pour l'application, soit elle se contente d'améliorer différents éléments de performances du réseau. Ainsi, les techniques d'accès basées sur le passage de jetons ou un découpage temporel en cycle élémentaire (FTT-ETHERNET) permettent de garantir un accès borné au réseau pour les trames critiques. Cette notion de déterminisme reste toutefois implicite, et se base sur différentes formes de garantie. Si dans Ethereal la garantie porte sur une réservation de la bande passante, CSMA/DCR garantit simplement un mécanisme de résolution des collisions déterministe. Parmi les améliorations du service initial offert par les réseaux Ethernet, on peut citer la limitation de l'effet de capture de l'accès par une station avec la technique BLAM, la réduction de la probabilité d'occurrence d'une collision avec les régulateurs de trafic ou encore l'optimisation de la distribution de la charge dans (Rondeau *et al.*, 2001).

Si l'on regarde maintenant les effets secondaires de ces propositions, on constate qu'elles conduisent à une surcharge protocolaire supplémentaire plus ou moins importante. Les mécanismes d'accès basés sur le passage de jeton ou de type maître/esclave conduisent ainsi à l'émission de données non utiles, ce qui induit une consommation inutile de bande passante. De plus, les travaux qui ne suppriment pas totalement les collisions (comme le lissage de trafic) ne permettent pas d'inhiber l'occupation temporelle du médium liée aux collisions.

Le choix des deux critères suivant repose alors sur le fait que le succès d'Ethernet est basé en partie sur son faible coût et sa simplicité de gestion. Ces deux caractéristiques sont en effet dues à la simplicité des mécanismes mis en jeu ainsi qu'à l'équité de l'accès CSMA/CD. Aussi, l'ajout de mécanismes plus avancés qui permettront de tendre vers un Ethernet déterministe conduit inversement à compliquer le protocole. Des solutions basées sur un mécanisme d'accès comme TDMA ou maître/esclave nécessitent des messages et piles logicielles supplémentaires. L'implémentation matérielle de ces nouveaux mécanismes (comme dans CSMA/DCR) peut alors permettre de simplifier la lourdeur de cette gestion. Toutefois, comme le montre le tableau, peu de ces travaux ont abouti à une distribution commerciale et la plupart restent des travaux de recherche, à l'inverse d'Ethernet qui jouit d'un facteur d'échelle important lié à sa vaste diffusion dans les environnements informatiques.

TAB. 2.2: Comparaison des propositions visant à rendre Ethernet déterministe (Apports des travaux)

Proposition	*Conformité à la norme*	*Déterminisme*	*Amélioration des performances d'Ethernet natif*	*Surcharge protocolaire*	*Facilité d'implémentation et de gestion*	*Disponibilité commerciale*	*Possibilité d'étude de majorants*
CSMA/DCR	non[a]	oui	-	oui	support matériel	pauvre	oui
PCSMA	non	non	élimination des collisions entre trafic critique et non contraint	oui	lourde	non	non
CABEB, BLAM	non[b]	non	équité accrue[c]	limitée	complexe	quelques produits	non
FTT-ETHERNET	non[d]	oui	-	oui	lourde	non	oui
P-CSMA	non	non	empêche les collisions entre trames de priorité différente	oui	lourde	non	non
Protocoles à jetons de temps	non	oui	-	importante	lourde	pauvre	oui
Protocoles à temps virtuels	non	non	ordonnancement multiple [e]	limitée	complexe	inconnue	non
Protocoles à fenêtres	non	non	ordonnancement multiple	limitée	complexe	inconnue	non
Lissage de trafic	non	non	diminution de la probabilité d'occurrence des collisions	limitée	complexe	non	non
ETHEREAL	non	oui	réservation de bande passante	oui	lourde	non	oui
commutateur à QdS (Hoang, 2004)	non	oui	réservation de bande passante	oui	lourde	non	oui
VLANs (Modlovansky, 1998)	oui	non	diffusion multicast	non	-	oui	non
topologie commutée (Rüping *et al.*, 1999)	oui	non	architectures commutées	non	-	sans objet	non
$(m,k) - firm$ (Song, 2004)	non	non	écartement de paquets intelligent	oui	complexe	non	oui
optimisation topologie (Rondeau *et al.*, 2001), (Krommenacker, 2002)	oui	non	diminution de la charge	non	-	sans objet	non
Notre approche	oui	-	-	-	-	sans objet	oui

[a] matériel (modification du firmware de la carté réseau)
[b] modification du compteur de collisions
[c] limitation de l'effet de capture de l'accès par une station
[d] ajout d'une couche de contrôle d'admission au dessus d'Ethernet
[e] permet la mise en œuvre de différentes politiques d'ordonnancement suivant différents paramètres comme l'échéance des messages

Comme nous l'avons vu au chapitre 1, les études de la stabilité des systèmes contrôlés en réseau nécessitent de pouvoir majorer le service offert par le réseau. Différentes études sont alors possibles : si le lissage de trafic s'appuie sur une étude statistique de la performance du réseau, le mécanisme d'accès FTT permet de borner l'arrivée des données. Toutefois, ces propositions s'appuient sur une modification de la normalisation qui ne permet pas de profiter des qualités d'Ethernet comme l'interopérabilité ou la flexibilité (ajout d'une nouvelle station dans TDMA).

Notre approche vise donc simplement à une étude de majorant des réseaux Ethernet simplement définis dans la normalisation (sans modifications matérielles ou ajout d'une couche logicielle supplémentaire). Il s'agit uniquement d'une évaluation de performances qui n'apporte aucune modification du réseau comme le montre le tableau 2.2, mais qui va offrir une connaissance déterministe *a priori* du comportement du réseau en fonction de son utilisation.

2.5 Conclusion

L'état de l'art mené dans ce chapitre autour des solutions visant à conférer à Ethernet un certain déterminisme montre bien que cette amélioration du réseau se fait au détriment des qualités intrinsèques d'Ethernet (faible coût, flexibilité, équité et simplicité). C'est pourquoi l'étude que nous allons développer par la suite se concentrera sur une évaluation de performances des mécanismes définis dans les normes Ethernet.

Compte tenu des exigences des systèmes contrôlés en réseau en temps-réel, et au vu des nombreux débats lors de conférences entre l'Ethernet partagé (traditionnel) et l'Ethernet commuté, nous notons que ce dernier mode semble être plus adapté. En effet, dans la mesure où nous nous attachons à la normalisation, il est nécessaire de s'affranchir de l'indéterminisme des collisions. Les architectures Ethernet commutées full-duplex micro-segmentées nous offre ce cadre de travail. Par rapport à la figure 1.14 que nous complétons maintenant, la topologie du réseau Ethernet retenue sera basée sur des commutateurs comme le montre la figure 2.10.

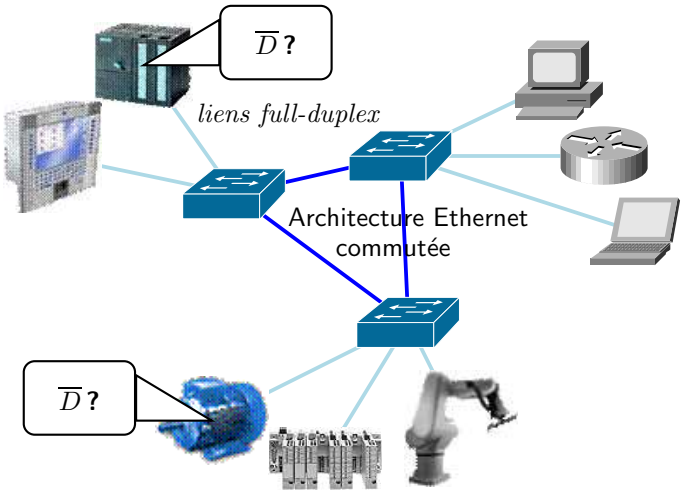


FIG. 2.10: L'architecture du réseau étudiée est de nature commutée full-duplex micro-segmentée

Compte tenu des résultats présentés dans (Jasperneite et Neumann, 2001) et (Krommenacker, 2002), l'architecture privilégiée par la suite sera la structure hiérarchique. Néanmoins, il ne s'agit pas là d'un cadre restrictif et l'évaluation proposée dans la suite pourra aussi bien s'appliquer sur des structures linéaires, redondantes ou maillées.

Enfin, l'analyse des travaux d'évaluation de performances présentée au paragraphe 2.3.5 ne fournit que des délais moyens et qui sont obtenus dans des hypothèses relativement restrictives (taille des trames fixes, modélisation des commutateurs relativement simpliste, ...). Dans l'optique d'une utilisation du réseau Ethernet commuté comme architecture de distribution des systèmes contrôlés en réseau, il est nécessaire de valider l'évaluation de performances en terme de garantie temporelle. La problématique de l'implémentation d'Ethernet dans les réseaux industriels reste donc un problème ouvert. Dans cette thèse, nous proposerons une méthode de calcul de majorant $\overline{D}$ des délais de traversée des réseaux Ethernet commutés standards dans un contexte industriel prenant en compte les différences de longueur et de destination des messages. Pour cela, nous ferons appel à la théorie nommée calcul réseau, initiée par (Cruz, 1991*a*) et (Le Boudec et Thiran, 2001) et issue de la théorie des dïodes Min-Plus, qui comme le montre (Thiele *et al.*, 2000), est utilisable pour les systèmes temps réel.

Deuxième partie

Contributions

Chapitre 3

Modélisation du réseau et du trafic

L'utilisation d'Ethernet commuté comme support de communication d'applications distribuées à temps critique nécessite d'être capable de borner les délais de bout en bout. En effet, il est indispensable de répondre à la question : compte tenu des interactions entre les équipements distribués (liste des communications, volume, fréquence ... et criticité temporelle de l'information), est-ce qu'un réseau Ethernet commuté peut satisfaire les besoins temps-réel de l'application ? En fait, il s'agit de s'assurer que le réseau sera suffisamment bien conçu et performant pour ne pas rendre instable le process.

Dans la mesure où Ethernet commuté n'apporte aucune garantie, la solution consiste à développer une modélisation analytique de ce type de réseau permettant d'estimer le niveau de service offert et de déterminer *a priori* s'il satisfait les contraintes applicatives. L'étude suivante porte sur l'expression des pires délais de bout en bout. Cette information constitue l'un des plus importants critères de performance des applications distribuées. Cela implique que la théorie utilisée soit déterminisme et se focalise sur les pires cas. L'utilisation de théories stochastiques ((Song, 2001),(Lee, 1995)) paraît donc inadéquate. Aussi, nous nous appuierons sur une théorie assez récente, le calcul réseau ou *Network Calculus*. Cette théorie est présentée comme une théorie des systèmes déterministes à file d'attente pour l'Internet.

3.1 Le calcul réseau comme support d'analyse

3.1.1 Introduction

Le calcul réseau est une théorie déterministe des systèmes de bufferisation rencontrés dans les réseaux informatiques. La principale différence avec les théories des systèmes traditionnels, comme par exemple celle qui s'applique à la conception de circuits électroniques, est qu'elle fait appel à une autre algèbre, où les opérations sont transformées comme suit : l'addition devient le calcul du minimum et la multiplication devient l'addition. En fait, les fondements du calcul réseau résident dans la théorie mathématique des dioïdes, et en particulier, le dioïde Min-Plus (autrement appelé algèbre Min-Plus). Cet ensemble de résultats mathématiques, résultant souvent de démonstrations relativement complexes, permet de représenter des systèmes complexes tels que les programmes concurrents, les circuits numériques et bien sûr les réseaux de communication. Cette théorie a été essentiellement développée par Jean-Yves Le Boudec, René Cruz et Cheng-Shang Chang (Chang *et al.*, 2002).

L'une des originalités de la théorie du calcul réseau est qu'elle peut être généralisée et vue comme une

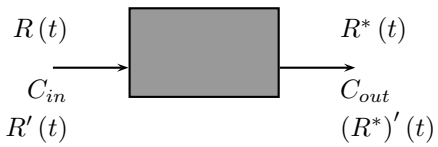
théorie des filtres à l'aide de l'algèbre min-plus (Le Boudec et Thiran, 1998)(Chang *et al.*, 2002)(Chang, 2000). L'utilisation de l'algèbre $(\min, +)$ appliquée aux réseaux de communication n'est pas récente. Elle s'est notamment développée à partir de graphes d'événements temporisés (Boimond, 1999) qui ont notamment été utilisés pour la modélisation de la congestion dans les réseaux ATM (Daoudi *et al.*, 2001). Dans (Baccelli et Hong, 2000), l'algèbre $(max, +)$ permet de caractériser le comportement du protocole réseau TCP. L'utilisation de l'algèbre $(\min, +)$ en tant que théorie des filtres a simultanément été développée dans (Cruz et Okino, 1996)(Chang, 1997)(Le Boudec, 1998).

3.1.2 Concepts

L'originalité du calcul réseau par rapport à d'autres théories plus traditionnelles est portée par la modélisation des flux en entrée des systèmes à files d'attente. Plutôt que de prendre en compte un modèle stochastique (Song, 2001)(Starobinski et Sidi, 2000)(Torab, 2000), le calcul réseau introduit une représentation déterministe du trafic. Il ne s'agit plus de décrire par exemple la fréquence d'arrivée des messages mais de supposer que cette arrivée respecte certaines limites. Cette notion s'apparente à une contrainte sur l'arrivée.

Pour cela, un flux de données est décrit par différentes fonctions cumulatives qui représentent la quantité de données reçues depuis un instant donné. En supposant que les observations commencent à $t = 0$, soit :

- C_{in} la **capacité** (le débit) du lien en **entrée** du système
- C_{out} la **capacité** du lien en **sortie** du système
- $R(t)$ correspond au **nombre de bits cumulés** arrivés à l'**entrée** d'un système durant l'intervalle $[0, t]$. $R(t)$ est appelée fonction d'arrivée des données.
- $R'(t)$ la **vitesse instantané** d'arrivée des données à l'instant t en **entrée** d'un système. A un instant donné, cette vitesse ne peut prendre que deux valeurs : soit elle est nulle (aucune communication), soit elle correspond à C_{in}, la capacité du lien en entrée du système par lequel arrive les données.
- $R^*(t)$ correspond au **nombre de bits cumulés** arrivés à la **sortie** d'un système à l'instant t. $R^*(t)$ est appelée la fonction de départ (la fonction d'arrivée en sortie du système)
- $(R^*)'(t)$ la **vitesse instantané** d'arrivée des données à l'instant t en **sortie** d'un système.



L'analyse sera établie en continu, de sorte que t et $R(t)$ soient réels. (Chang, 2000) présente la même étude en discret.

L'étude suivante reprend les définitions et les propriétés démontrées dans le cadre du *Network Calculus*. Les résultats suivants font appel aux travaux de (Cruz, 1991*a*)(Cruz, 1991*b*)(Cruz, 1995), repris dans (Chang, 2000) et étendu dans (Le Boudec et Thiran, 2001).

3.1.3 Définitions et propriétés établies par Cruz

Définitions

Dans cette partie, les notions de travail et de traitement sont définies. Dans un premier temps, il s'agira de notions génériques du calcul réseau, dans un second temps, spécifiques à l'approche "évaluation de performances".

Pour un système, la notion de *quantité de travail* peut être définie comme la mesure du volume d'informations à traiter. Elle s'applique généralement en entrée d'un système (quantité de travail en entrée) ou à l'intérieur du système (quantité de travail en cours de traitement). Le *traitement* ou *service* étant l'opération réalisée par le système sur l'information. Le système est alors caractérisé par son *taux de service* : vitesse de traitement de l'information.

Arriéré de traitement $x(t)$

L'arriéré de traitement (ou *backlog* dans la littérature anglophone) correspond dans l'étude menée par Cruz à la quantité de données présente dans un système de nature file d'attente. Plus précisément, il correspond à la quantité de bits en attente à l'intérieur d'un système.

L'arriéré de traitement à un instant t d'un système est défini par :

$$x(t) = R(t) - R^*(t) \tag{3.1}$$

Cette définition suppose l'observation simultanée de l'entrée et de la sortie. Cette propriété sera utilisée par la suite pour définir l'état du réseau et en déduire les délais de traversée.

Délai virtuel $d(t)$

Le délai virtuel à un instant t est défini par :

$$d(t) = \inf\{T : T \geq 0 \; et \; R(t) \leq R^*(t+T)\} \tag{3.2}$$

Le délai virtuel $d(t)$ s'apparente au délai subi par un bit arrivé à un instant t si tous les bits arrivés précédemment sont servis avant lui. Ce qui revient à dire que si la politique de traitement est FIFO, alors il s'agit du délai de traversée du système. De plus, si $R^*(t)$ est continu (pas de traitement par lot), alors $R^*(t + d(t)) = R(t)$.

Comme nous l'avons indiqué, les travaux initiaux de Cruz se caractérisent par l'utilisation de contraintes pour décrire les entrées d'un système à file d'attente. La fonction d'arrivée $R(t)$ qui varie au cours du temps n'est *a priori* pas connue et sera complétée par la notion de courbe d'arrivée.

Courbe d'arrivée $\alpha(t)$

La courbe d'arrivée représente la contrainte appliquée à l'arrivée des données d'un flux en entrée d'un système. Elle est définie comme suit.

Soit α, une fonction non décroissante définie pour $t \geq 0$. Un flux R est contraint (régulé) par α si et seulement si

$$\forall s \leq t, \; R(t) - R(s) \leq \alpha(t-s) \tag{3.3}$$

De la même manière, le calcul réseau introduit une définition du service offert par un système à files d'attente au travers d'une fonction qui représente la quantité de données cumulées traitée au cours du temps : la courbe de service.

Courbe de service $\beta(t)$

Soit β, une fonction non négative non décroissante. La quantité de données d'un flux servie par une file est définie par une courbe de service de file (ou *queue service curve* en anglais) β si et seulement si

$$\forall t,\ \exists\ t_0 \leq t \text{ tel que } R^*(t) - R(t_0) \geq \beta(t - t_0) \tag{3.4}$$

Une courbe de service implique donc que $\beta(0) = 0$. Cette appellation de courbe de service de file introduite dans (Parekh et Gallager, 1993) et (Le Boudec, 1996) renvoie au fait que la définition précédente est dédiée au système à file d'attente. Comme le montre la définition de la courbe de service, il s'agit en fait de la courbe de service minimal puisqu'elle représente le service le plus faible offert par le système. L'intérêt de ce type de courbe est qu'elle nous permettra de prendre en compte au travers du pire service, le pire délai. (Le Boudec et Thiran, 2001) présentent la courbe de service maximale qui permet de travailler sur le délai de traversée minimal.

Nous venons de présenter les concepts initiaux introduits par Cruz. Comme nous l'avons indiqué, ces définitions peuvent être exprimées dans l'algèbre $\min +$. Nous donnons ici les expressions présentées par (Le Boudec et Thiran, 2001) à partir des notations suivantes :

- $\inf S$, l'infimum dans $\mathbb{R}$, $\wedge$ le minimum dans $\mathbb{N}$.
- sup le supremum et $a \vee b = \max\{a, b\}$ le maximum.
- $\otimes$, la convolution min-plus

  Soit f et g deux fonctions ou séquences de F. La min-plus convolution de f et g est la fonction :

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

- et $\oslash$, la déconvolution min-plus

  Soit f et g deux fonctions ou séquences de F. La min-plus déconvolution de f par g est la fonction :

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

(Le Boudec et Thiran, 2001) redéfinissent ainsi :

- la courbe d'arrivée $\alpha(t)$

  Soit une fonction α non décroissante définie par $t \geq 0$ ($\alpha \in F$), un flux R est contraint par α si et seulement si

$$\forall s \leq t,\ R \leq R \otimes \alpha$$

- la courbe de service $\beta(t)$

  Soit un système S et un flux traversant S avec comme fonctions d'arrivée et de départ R et R^*. S offre au flux une courbe de service β si et seulement si

$$\beta \in F,\ R^* \geq R \otimes \beta$$

3.1.4 Fondements du calcul réseau

Nous utiliserons notamment les écritures et formules données dans (Le Boudec et Thiran, 2001).

Majorant de l'arriéré de traitement

Définissons la fonction $W_{C_{out}}(R')(t)$ pour $-\infty < t < +\infty$ telle que

$$W_{C_{out}}(R')(t) = \max_{s \leq t} \left[R(t) - R(s) - C_{out}(t-s) \right] \tag{3.5}$$

$W_{C_{out}}(R')(t)$ correspond à la taille de l'arriéré de traitement à l'instant t dans un système qui reçoit des données au taux $R'(t)$ et les retransmet à un taux C_{out}. Son interprétation graphique est la suivante. Sur la figure 3.1, on retrouve dans un premier temps le chronogramme d'arrivée du travail. Cette arrivée est conditionnée par la vitesse du lien si bien que la durée d'arrivée des paquets est proportionnelle à leur taille. Il est alors à noter qu'à l'instant t', la courbe de pente C_{out} n'a pas encore rejoint $R(t)$. Cela signifie qu'une partie des données reçues n'a pas encore été traitée. A cet instant, $W_{C_{out}}(R')(t) > 0$: du travail est en retard (notion que nous attribuerons à de la congestion).
La figure 3.1 illustre alors l'apparition de l'arriéré de traitement. L'arrivée des données en entrée du système est défini par le graphe du bas. Il représente la vitesse d'arrivée instantanée des trames $R'(t)$. De sorte que chacun des rectangles correspond à une nouvelle trame. La vitesse arrivée d'une trame est liée à la capacité du lien en entrée, C_{in}.

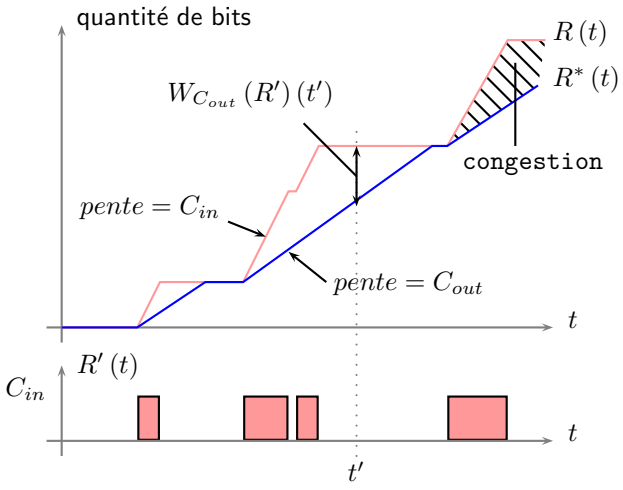


FIG. 3.1: Interprétation graphique de l'arriéré de traitement

La figure 3.1 confirme ce que nous avions vu précédemment, à savoir que plus la valeur de l'arriéré de traitement sera importante, plus la congestion sera grande, et plus le délai de traversée des données sera long. Le troisième paquet souffrira ainsi d'un délai plus grand que le second puisqu'à son arrivée, le système n'avait pas fini de traiter le second paquet. Dans le cas où la congestion augmente et que le système devient incapable de stocker la quantité de travail en attente, nous parlerons de saturation.

Enfin, l'arriéré de traitement peut être majoré en fonction des courbes d'arrivée et de service. Ainsi, pour un flux contraint par une courbe d'arrivée α offrant une courbe de service β, l'arriéré est majoré

par :

$$\forall t,\ R(t) - R^*(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\} \tag{3.6}$$

La démonstration est donnée en annexe. Par rapport à $W_{C_{out}}(R')(t)$, la définition d'une courbe d'arrivée nous permet d'écrire :

$$W_{C_{out}}(R')(t) \leq \sup_{s \geq 0} \{\alpha(s) - C_{out}(s)\}$$

Autrement dit, l'arriéré est majoré par la distance verticale maximale séparant les courbes d'arrivée et de service comme le montre la figure 3.1.

Majorant du délai

En suivant les mêmes hypothèses, (Le Boudec et Thiran, 2001) montrent que le délai est majoré par :

$$\forall t,\ d(t) \leq \sup_{s \geq 0} (\inf \{T : T \geq 0\ et\ \alpha(s) \leq \beta(s+T)\}) \tag{3.7}$$

La partie droite de l'inégalité correspond en fait à la distance horizontale entre les courbes d'arrivée et de service comme le montre la figure 3.2.

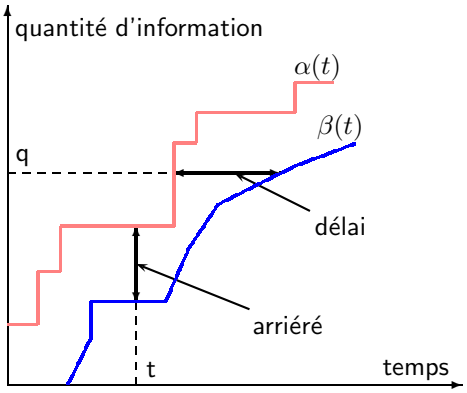


FIG. 3.2: Délais et arriéré : distances horizontales et verticales entre les courbes d'arrivée et de service

3.1.5 Exploitation

Les résultats présentés dans ce paragraphe établissent la possibilité d'obtenir un majorant du délai de traversée à partir des seules courbes d'arrivée et de service. La problématique qui se pose alors est l'obtention de ces courbes. Plus particulièrement, comment caractériser les réseaux Ethernet commuté afin d'en définir la courbe de service minimale. Etant donnée la définition d'un commutateur, sa courbe de service n'est pas intuitive.

A la base, les fonctions d'arrivée ou de service ne répondent pas forcément à un type de fonctions particulières. Elles doivent simplement être croissantes au sens large, c'est-à-dire que $f(s) \leq f(t)$ pour tout $s \leq t$. L'ensemble des fonctions admissibles est regroupé dans F, l'ensemble des fonctions croissantes au sens large pour lesquelles $\forall t < 0,\ f(t) = 0$. De plus, il est convenu que la fonction $f = \{f(t), t \in \mathbb{R}\}$ est continue à gauche. Par conséquent, l'intervalle de définition des fonctions de F est $\mathbb{R}^+ = [0, +\infty]$. Les fonctions *rate-latency server* telles que $\beta(t) = \beta_{R,T}(t) = R[t-T]^+$ sont traditionnellement utilisées pour

définir le service des routeurs (Le Boudec, 1998) ou des commutateurs (Jasperneite *et al.*, 2002)(Watson, 2002)(Watson et Jasperneite, 2003). Ce type de fonction présente l'avantage de simplifier les calculs mais ne représente qu'une modélisation sommaire d'un commutateur. De plus le problème est simplement repoussé à la définition des paramètres de latence et de capacité conformes à ce qu'un commutateur va réellement offrir.

3.2 Modélisation de la commutation : identification d'une courbe de service

3.2.1 *802.1D*, ou qu'est ce qu'un commutateur ?

Le concept de commutation, ou pont couche MAC (Media Access Control) est défini dans le standard IEEE 802.1D (IEEE, 1998). Le process de retransmission d'un pont MAC permet le stockage de trames en file d'attente avant leur soumission pour retransmission sur les entités MAC individuelles de chaque port du pont. Le standard précise les étapes de la commutation comme indiqué sur la figure 3.3.

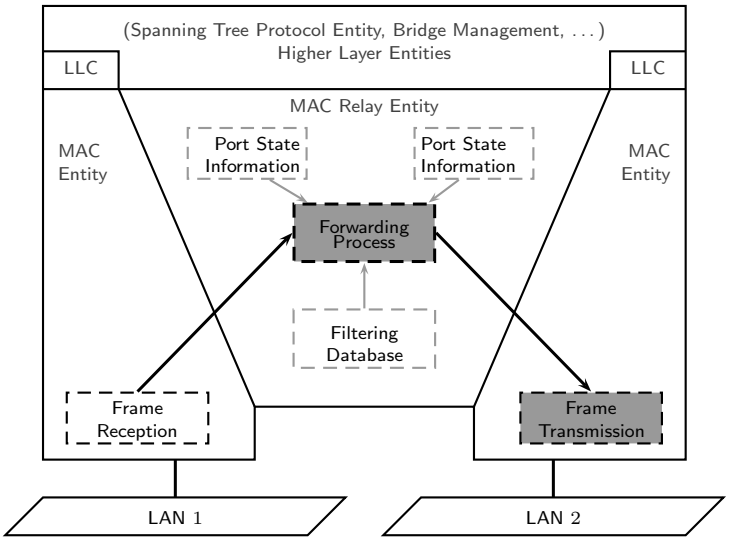


FIG. 3.3: Relais d'une trame par un pont *802.1D* (IEEE, 1998, figure 7-4)

La figure 3.3 présente une partie des fonctions mises en œuvre pour assurer la commutation d'une trame. Elle identifie ainsi les 3 points suivants :

- réception de la trame
- traitement de la retransmission
- transmission de la trame

Cette figure identifie également une architecture interne au pont permettant la succession des points précédents, ainsi que d'autres fonctionnalités que nous mettrons en évidence par la suite. Notre but est d'ici de déterminer un modèle de commutateur adapté et le plus proche des fonctionnalités d'un commutateur 802.1D. Plutôt que de recourir à une vue abstraite d'un commutateur (Watson et Jasperneite, 2003), nous chercherons à définir un modèle à partir d'une analyse fonctionnelle des opérations

successives réalisées par un commutateur. La figure 3.3 indique les différentes étapes du processus de retransmission d'une trame entre deux ports. Il importe ainsi de noter que le terme anglais *transmission* vaut pour l'émission d'une trame en sortie du commutateur et non pour l'intégralité du processus. De la même façon, sous l'anglicisme *forward process*, on privilégiera la notion de processus de retransmission. L'ordonnancement interviendra alors juste avant cette *transmission*, comme l'art de sélectionner les trames pour émission immédiate.

Les commutateurs sont donc des systèmes complexes qui introduisent différents mécanismes et différentes technologies. (Phipps, 1999*a*) et (Seifert, 2000) décomposent les plates-formes de commutation en trois fonctions principales :

- le **modèle de bufferisation** [*queuing model*] se réfère à la bufferisation et aux mécanismes de congestion localisés dans les commutateurs,
- l'**implémentation de commutation** [*switching implementation*] fait quant à elle appel au processus de décision à l'intérieur des commutateurs (c'est-à-dire comment et où la décision de commutation est prise),
- et enfin, la **fabrique de commutation** [*switching fabric*] est le mécanisme utilisé pour faire passer les données d'un port à un autre.

Toutefois, il existe de multiples solutions pour construire chacune de ces fonctions à l'intérieur des commutateurs. Aussi dans cette étude, nous allons nous limiter à un type d'architecture commutée qui intégrera les mécanismes les plus implémentés par les constructeurs. Nous reprenons alors les trois étapes clés du processus de commutation identifiées par (Phipps, 1999*a*) et (Seifert, 2000). Enfin, divers phénomènes font des commutateurs des éléments potentiellement bloquants. L'étude suivante tiendra compte des aspects suivants :

blocage ou surabonnement (en anglais, *blocking or oversubscription*) condition dans laquelle la totalité de la bande passante des ports est supérieure à la capacité de la fabrique de commutation,

blocage de tête de ligne (en anglais, *head of line blocking*) quand la congestion sur un port de sortie limite la sortie à des ports non saturés.

3.2.2 Technologies ...

Ces états inspirent alors les solutions technologiques du marché dans le sens où elles tentent d'éliminer ces blocages. Notons également que seules sont concernées les technologies de niveau 2 - sous-couche MAC (et non celles de la couche réseau). Les informations suivantes sont issues de (Phipps, 1999*b*).

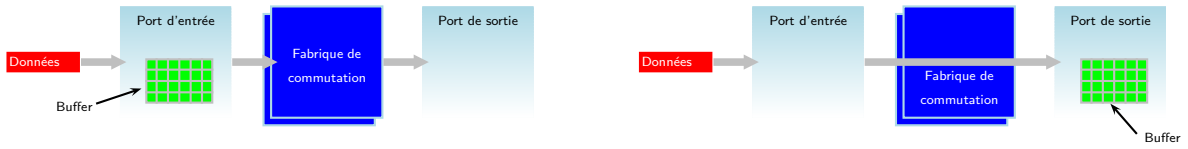
Le premier travail des commutateurs consiste à la réception de la trame. Une trame doit être stockée dans une file d'attente avant d'être sujette à retransmission. Ceci peut être pénalisant dans le cas où le port de sortie est prêt à retransmettre la trame et que l'arriéré de traitement du commutateur est nul. Ce constat a amené à proposer différentes politiques de retransmission.

- *store & forward*. Dans ce mode, le commutateur doit attendre la réception complète de la trame avant d'effectuer toute étape de la commutation.
- *cut-through*. La commutation est activée dés la réception de l'adresse de destination, soit à la réception du $14^{ème}$ octet (voir format des trames Ethernet).
- *fragment-free*. Le commutateur attend le $64^{ème}$ octet, c'est-à-dire la réception complète d'une trame de longueur minimale.

L'avantage du mode *store & forward* est l'implémentation des fonctionnalités de la sous-couche LLC comme indiqué sur la figure 3.3. L'intérêt est de pouvoir détecter des trames erronées (calcul du CRC) et d'empêcher de les relayer. Néanmoins, elle ajoute un retard de traitement d'autant plus important que la trame est longue. Le mode *cut-through* réduit ce délai à sa valeur minimale qui est indépendante de la longueur de la trame. Sous l'hypothèse d'une faible fréquence d'apparition de trames erronées, le mode *cut-through* sera envisagé dans la mesure où il améliore les délais de traversée (et non de bufferisation !).

. . . de mémorisation

La gestion de la congestion (données en attente de traitement à l'intérieur d'un équipement) est nécessaire lorsque plusieurs ports d'entrée concourent au même port de sortie. De longues rafales de trafic en entrée peuvent également surcharger le commutateur (Seifert, 2000, p 587). Ainsi, dans le cas où un commutateur est bloquant, c'est-à-dire que la somme des débits des ports d'entrée est supérieure au débit interne du commutateur, la congestion croît de plus en plus rapidement jusqu'à complètement saturer le commutateur. C'est pour cela que les commutateurs sont dotés de files capables de supporter cette congestion. L'un des problèmes à résoudre est donc de déterminer comment cette mise en attente est gérée à l'intérieur des commutateurs. Cette question va en fait dépendre de la localisation physique des éléments de mise en attente : les buffers. Il existe trois possibilités (Phipps, 1999*a*) ; l'emplacement des buffers n'étant pas anodin.



(a) en entrée (b) en sortie

FIG. 3.4: Modèle de bufferisation

mise en attente en entrée Dans 3.4(a), les paquets sont stockés au niveau du port d'entrée. Le problème de cette implémentation vient du fait que le phénomène de blocage en tête de ligne peut apparaître et peut réduire jusqu'à 60 % la bande passante.

Ce blocage en tête de ligne peut survenir lorsqu'un port de sortie devient congestionné suite à une concentration du trafic vers celui-ci. Dans le cas d'une mise en attente en entrée, et de l'arrivée de deux paquets en entrée sur le même port (le premier vers le port de sortie congestionné et le second vers un autre port), le second paquet ne peut être transmis du fait de l'attente du premier paquet alors que son port de sortie est disponible.

mise en attente en sortie Dans 3.4(b), le stockage des paquets est cette fois réalisé en sortie.

Dans cette configuration, le blocage de tête de ligne est éliminé puisque si l'on reprend l'exemple précédent, les deux paquets entrent directement dans le coeur de commutation et comme en sortie, ils sont dirigés vers deux port différent, le second paquet n'a pas à souffrir de la congestion de l'autre sortie. Néanmoins, les queues de sortie peuvent toujours être de taille insuffisante dans le cas de rafales.

De ces deux implémentations, la seconde reste la plus appropriée et comme on le verra par la suite, elle se conjugue parfaitement avec la technologie de fabrique de commutation nommée mémoire partagée.

De plus, elle supporte davantage l'intégration de la différenciation de services par l'ajout de multiples files en sortie comme représenté sur la figure 3.5.

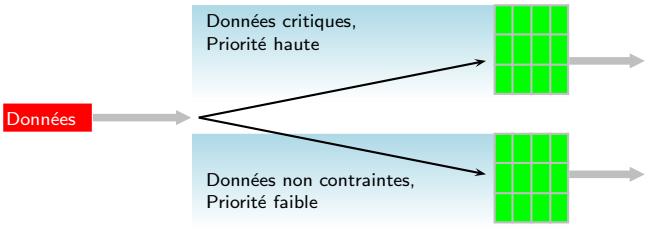
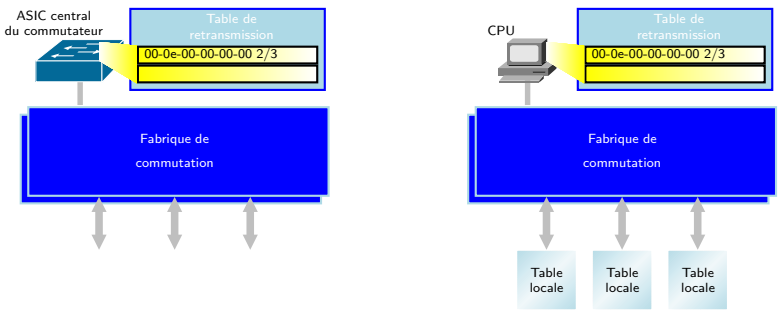


FIG. 3.5: Implémentation de plusieurs files sur chaque port

mise en attente dans une mémoire partagée Les buffers peuvent enfin être implémentés entre l'entrée et la sortie. Cette dernière implémentation, appelée *shared-memory queuing* présente l'avantage d'éliminer les blocages en tête de ligne générés par les buffers attachés aux ports d'entrée. Elle fournit également une plus grande bande passante que dans le schéma de bufferisation par port. Aussi, la bufferisation à mémoire partagée est l'organisation que nous retenons pour modéliser des architectures commutées.

. . . de décision

L'étude de l'implémentation de la commutation vise à déterminer à quel endroit la décision de retransmettre un paquet vers tel ou tel port de sortie est prise. Cette opération consiste à exécuter une recherche au niveau de la table de redirection qui est maintenue par le processus d'apprentissage des adresses sources. Le résultat est un vecteur de sortie composé d'un port de sortie unique, de multiples ports de sortie, du port interne du commutateur ou alors d'aucun port. L'implémentation peut être de deux natures : centralisée ou distribuée.



(a) Centralisée (b) Distribuée

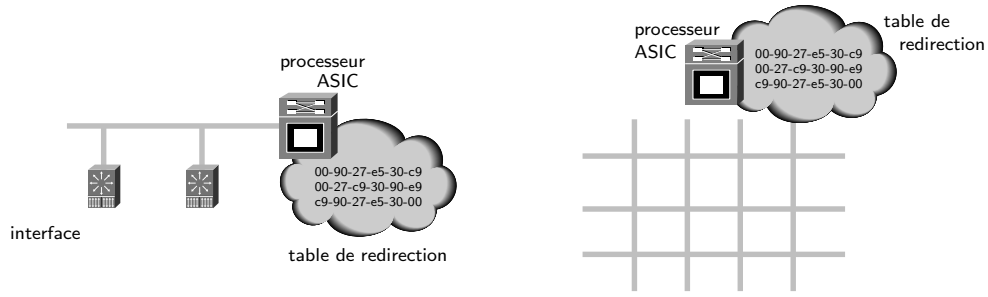
FIG. 3.6: Implémentations de commutation

Dans le cas de l'implémentation centralisée, une table centrale unique de redirection est utilisée. Cela fournit un contrôle central de la commutation et facilite l'apprentissage des chemins. A l'inverse dans l'architecture distribuée, la décision de commutation doit être directement prise par le port (ou le module). Cela induit la constitution de plusieurs tables de redirection ainsi qu'un maintien et une

synchronisation relativement plus complexe. Comme nous allons le voir, l'architecture centralisée est davantage appropriée lors de l'utilisation d'une fabrique à mémoire ou à bus partagé.

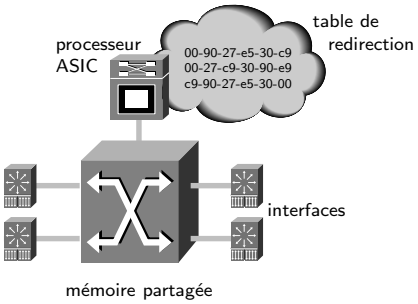
. . . de fabrique de commutation

La fabrique de commutation a pour fonction le transfert des trames entre les ports d'entrée et de sortie du commutateur. Elle peut être de trois types.



(a) à bus unifié

(b) à matrice de commutation



(c) à mémoire partagée

FIG. 3.7: Fabriques de commutation

à bus unique Dans cette architecture, un seul élément compose la fabrique : le bus (figure 3.7(a)). Chaque port doit gérer son accès au bus, mais l'acheminement des paquets en diffusion s'avère relativement simple. (Phipps, 1999*b*) met également en avant le fait que cette architecture n'engendre pas de surabonnement non prévisible.

à matrice La figure 3.7(b) montre que cette fabrique peut être vue comme un empilement de fabrique à bus puisque cette fois plusieurs bus sont autorisés de manière à constituer la grille.
Typiquement, cette fabrique ne présente pas de blocage. Néanmoins, la réalisation de la commutation pour des paquets à diffuser s'avère relativement complexe, de même que la consultation de la table de redirection. Le grand avantage de cette fabrique est de pouvoir supporter simultanément plusieurs flots de trafic.

à mémoire partagée Implémentation la plus répandue, l'architecture à mémoire partagée est illustrée sur la figure 3.7(c).
Les données sont mémorisées à l'intérieur de la fabrique. L'accès à la mémoire est gouverné par un processeur ASIC et cette architecture fait appel à des mémoires très rapides. Le cœur de commutation réalise l'accès à la table, la résolution de la destination *via* des pointeurs de la mémoire et

enfin, commute les paquets. La mémoire partagée est constituée de plusieurs buffers qui peuvent être fixes ou dynamiques. Les données entrent *via* les modules du commutateur dans la mémoire. Le processeur résout la destination et commute le paquet vers le module de sortie. Dans le cas de paquets en diffusion, la phase de commutation vers le module de sortie est répétée autant de fois que nécessaire.

La fabrique à mémoire partagée présente le plus faible niveau de complexité de commutation et d'implémentation. De fait, elle fournit la solution la moins chère et la plus répandue dans les commutateurs. L'implémentation centralisée conduit à une zone unique de mémorisation des trames.
Si le commutateur utilise une mémoire partagée, la trame peut être directement stockée dans cette mémoire. Le commutateur maintient alors une liste de pointeurs d'adresses mémoire qui constituent les seuls éléments des files des ports de sortie. Ceci évite les recopies mémoire des trames contrairement aux autres fabriques qui nécessitent la bufferisation des trames pour chaque port. Les trames sont alors mémorisées suivant deux cas :

- la trame attend la fin du processus de recherche du port de sortie qui est à ce stade inconnu. La mémoire est utilisée comme zone de stockage temporaire.
- la trame qui a été insérée dans la file de sortie appropriée attend d'être émise.

La notion de classes de service est implémentée au niveau des files de sortie ; en utilisant généralement une file par classe. Le traitement de chacune de ces files est alors défini par la priorité de la classe et la politique d'ordonnancement utilisée. C'est également au niveau des files de sortie que les trames de gestion des commutateurs (annonces BPDU du protocole Spanning Tree par exemple) sont insérées par le processeur central.

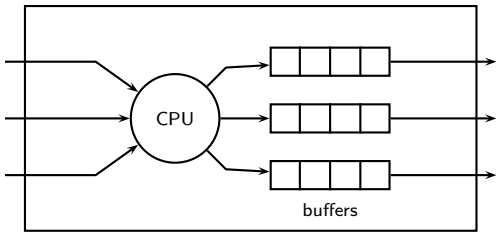
Du fait de l'implémentation centralisée, la capacité totale d'un commutateur implémentant une fabrique à mémoire partagée est définie par les caractéristiques de cette mémoire. A titre indicatif, considérons l'exemple suivant. Soit une mémoire à 32 *bit* de données et une horloge de 100 MHz. Comme chaque trame doit traverser deux fois la mémoire (à l'écriture depuis de le port d'entrée et à la lecture par le port de sortie), la capacité totale est donc limitée à $32\ bits * 100\ MHz\ /\ 2 = 1.6\ Gb/s$. La bande passante utilisable de la mémoire varie généralement de 50 à 70 % de ce maximum. Enfin, il est à noter que la majeure partie du délai de traversée d'un commutateur correspond à la concurrence d'accès en retransmission sur le port de sortie, et non à la concurrence d'accès à la mémoire partagée.

L'étude précédente nous permet de regrouper les différentes fonctionnalités de la commutation autour d'un modèle de commutateur.

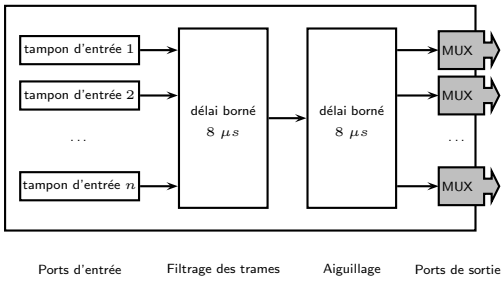
3.2.3 Modèles de commutateurs

Plusieurs modèles de commutateur ont déjà été proposés. (Song, 2001) propose ainsi un modèle (figure 3.8(a)) composé d'un serveur chargé d'assurer la commutation des trames en entrée vers les ports de sortie, et de buffers sur chaque port de sortie représentant l'attente avant retransmission.

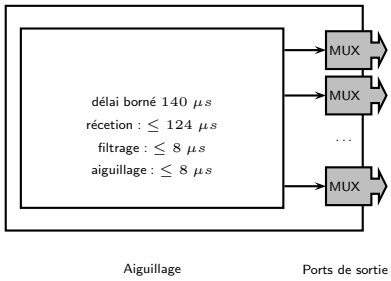
L'utilisation de ce modèle pour l'évaluation de performances se fait en deux parties. Le serveur, qui représente une implémentation centralisée, est associé à la latence du commutateur (qui est considérée comme égale à 10 μs) ainsi qu'à la latence de retransmission d'une trame (qui est fonction du mode de retransmission - voir paragraphe 3.2.2). Le modèle s'appuie sur une technique de bufferisation en sortie. Outre le fait que la théorie utilisée par la suite est de type stochastique, le modèle introduit par



(a) Proposé par (Song, 2001)



(b) Proposé par (Grieu, 2004)



(c) Modèle synthétisé (Grieu, 2004)

FIG. 3.8: Modèles de commutateur

(Song, 2001) ne détaille pas les technologies de commutation utilisées et ne prend donc pas en compte les limites physiques de l'architecture comme la capacité de la mémoire partagée.

(Grieu, 2004) propose alors de reprendre les différentes tâches accomplies par un commutateur et en déduit un modèle de commutateur prenant en compte différentes étapes de la commutation IEEE 802.1D. Comme le montre la figure 3.8(b), ce modèle introduit la réception des trames, le filtrage, l'aiguillage et l'émission en sortie. Le contexte industriel (Airbus) de ces travaux apporte alors différentes informations supplémentaires. Le cahier des charges imposé aux fabricants de commutateurs spécifie ainsi que chaque port du commutateur doit être capable de filtrer et d'aiguiller 125 trames en une milliseconde. Les auteurs synthétisent le modèle en deux éléments, le premier introduisant simplement un délai borné lié aux fonctionnalités du commutateur comme le montre la figure 3.8(c). Le multiplexeur FIFO est destiné à représenter l'émission de la trame sur le port de sortie, et plus particulièrement le goulet d'étranglement que représentent les ports de sortie.

La modélisation proposée dans (Grieu, 2004) ne s'applique que dans le cas particulier où l'on dispose de spécifications temporelles fournies par le fabricant. Le contexte de notre travail se veut toutefois plus généraliste comme nous l'avons défini au chapitre 2. Dans la mesure où le seul élément de départ reste la norme (IEEE, 1998), la modélisation que nous proposons se doit de développer davantage les étapes de la commutation.

3.2.4 Modélisation d'un noeud de commutation par R. Cruz

(Cruz, 1991*b*) propose de modéliser un commutateur en utilisant uniquement des composants élémentaires de multiplexage et de démultiplexage. Chaque port d'entrée du commutateur est relié à un démultiplexeur dont les sorties sont reliées à autant de multiplexeurs que de ports de sortie du commutateur. La figure 3.9 montre le principe de modélisation sur un commutateur 2 ports. Ce modèle prend comme hypothèse que le commutateur fonctionne en mode *virtual cut-through* (c'est-à-dire, sans bufferisation des trames

en entrée du commutateur (Kermani et Kleinrock, 1979)) puisqu'il n'intègre pas de buffer sur les ports d'entrée.

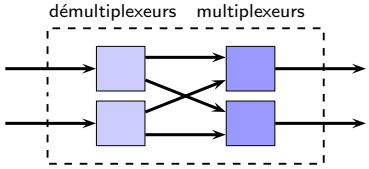


FIG. 3.9: Modèle n°1 de commutation proposé par (Cruz, 1991*b*)

Bien que le modèle initié par Cruz se présente davantage comme un modèle idéal de la commutation plus qu'un modèle réaliste des commutateurs du marché, l'analyse de ce modèle permet de mettre en lumière les points clés d'un modèle de commutateur 802.1D. Le problème posé est de regarder l'adéquation entre le modèle de commutation défini dans les travaux de R. Cruz et les techniques de commutation sur Ethernet décrites précédemment. Par rapport aux trois approches de fabrique de commutation précédemment énumérées, on peut éliminer les principes de la mémoire partagée et de la gestion par bus puisque la procédure de commutation est complètement distribuée. Elle s'apparente donc plus à une architecture de type matrice où chaque port d'entrée est directement mis en correspondance avec toutes les sorties. On peut alors remarquer que le modèle ne représente pas la notion d'arbitrage centralisé de la fabrique. En effet, il considère possible la commutation simultanée de deux ports d'entrée vers le même port de sortie. Or ceci renvoie à un problème technologique de gestion des collisions et n'est donc pas envisageable pour un commutateur Ethernet. L'incidence sur les temps de réponse du modèle sera que certains majorants seront inférieurs à ceux réellement constatés. A partir de là, nous allons présenter deux nouvelles propositions.

3.2.5 Nos propositions (Georges *et al.*, 2003*b*),(Georges *et al.*, 2003*c*)

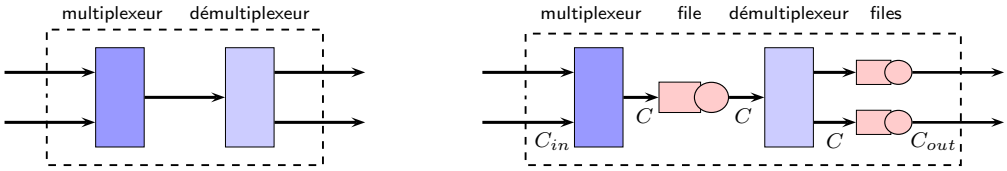
Vers un modèle centralisé

Le modèle précédent qui s'associe plutôt à une architecture matricielle ne nécessite pas de ressources partagées comme le bus et la mémoire. D'un point de vue macroscopique, il est possible de représenter simplement un point de rendez-vous de fabrique de commutation basée sur une architecture soit à bus unifié soit à mémoire partagée en sérialisant un multiplexeur et un démultiplexeur. La figure 3.10(a) représente ce second modèle.

De la même manière que pour le modèle n°1, seul le multiplexeur génère un retard. L'expression du délai de traversée sera donc identique, mais néanmoins les entrées seront différentes. En effet, dans le deuxième modèle, le multiplexeur en tête de ligne absorbe l'ensemble du trafic alors que dans la première solution, celui-ci est réparti sur les multiplexeurs de sortie.

Vers un modèle à mémoire partagée

Le modèle de Cruz et notre première proposition représentent un modèle fonctionnel de la commutation sans prendre en considération les approches technologiques internes d'un commutateur Ethernet,



(a) Modèle de commutation n°2 (b) Modèle de commutation n°3

FIG. 3.10: Vers un modèle de commutateur 802.1D

même si elles peuvent s'y apparenter comme nous avons tenté de le faire. Notre seconde proposition s'appuie sur la première en ajoutant entre le multiplexeur et le démultiplexeur une file FIFO qui va représenter à la fois la mémoire partagée et l'ASIC qui gère la table de retransmission. L'introduction de ce nouvel élément va nous permettre de différencier les vitesses de travail internes et externes au commutateur. Ainsi, le lien entre le multiplexeur et la file va correspondre à la vitesse du bus du commutateur C. De plus, la frontière entre les capacités de traitement interne et externe (C_{in} et C_{out}) du commutateur est modélisée par l'intermédiaire de files FIFO placées en sortie de chaque port du démultiplexeur (figure 3.10(b)).

Conclusion

L'étude précédente nous a permis d'identifier différents modèles de commutateur. Notre objectif étant porté sur l'évaluation de performances des réseaux basés sur le standard IEEE 802.1D, les modèles décrivent les différentes étapes de la commutation et doivent représenter le plus fidèlement possible la réalité. Dans la suite du document, chacun des modèles sera validé. Nous verrons que le modèle $n°3$ (figure 3.10(b), second modèle proposé avec une file FIFO pour la mémoire partagée et une file FIFO par port de sortie) sera jugé comme le plus pertinent. Ce modèle vaut pour une implémentation plus précise fondée sur une mémoire partagée. Le mode de retransmission pris en compte est de type *cut-through*. Le modèle est en ce point idéal dans la mesure où il n'intègre pas d'élément retardateur lié au temps de réception des 14 premiers octets (technique virtual cut-through) qui est considéré ici suffisamment négligeable. Ce modèle peut toutefois aisément être complété pour prendre en compte l'attente de réception de la trame en y incorporant une file de réception.

3.2.6 Conclusion : courbe de service et majorant du délai

L'écriture des résultats du calcul réseau dans l'algèbre $\min, +$ permet la détermination d'un majorant du délai de bout en bout. Ainsi, le délai de traversée d'un élément y est défini comme la distance horizontale entre les courbes d'arrivée et de service et est majoré par :

$$d(t) \leq \inf \{d \geq 0 : (\alpha \oslash \beta)(-d) \leq 0\} \tag{3.8}$$

Il est toutefois à noter que l'opération de déconvolution min-plus $\oslash$ peut être difficile à réaliser lorsque les fonctions étudiées ne présentent pas de propriétés simplificatrices. De plus, l'obtention seule de la courbe de service β associée à chacun des composants élémentaires des modèles proposés au paragraphe 3.2 n'est en aucun cas explicite. Si l'on cherche ainsi à déterminer le délai de traversée d'un multiplexeur à m entrées pour chacun des flux en utilisant la relation précédente, la courbe de service β associée doit

définir la quantité de travail minimale du flux considéré (et non totale) qui sera traitée par le multiplexeur. Cela signifie que pour chaque composant, il est nécessaire de définir autant de courbes de service que de communications. Une solution consiste alors à agréger les différents flux traversant le système en un super flux. Dans ce cas, la courbe de service globale du système peut être utilisée. Si le problème est simplifié, cela conduit à envisager le même majorant du délai de traversée pour chaque flux indépendamment des spécifités des communications comme la longueur des trames, ce qui apporte une sur-estimation des délais. De plus, cela ne permet pas de diagnostiquer le réseau (même expression du délai pour plusieurs flux) et d'isoler les points de congestion problématiques du réseau.

La difficulté de définir la courbe de service pour chacun des composants élémentaires du modèle d'un commutateur vient du fait qu'Ethernet n'inclut pas de garantie de service. Si c'était vrai, il serait alors possible de décrire le service minimal en utilisant les ressources garanties à un flux. C'est ce que nous verrons lors de l'étude de la Classification de Service.

L'alternative repose alors sur les travaux initiaux de Cruz (Cruz, 1991*a*). Le majorant du délai n'est plus directement considéré par rapport à la courbe de service, mais par rapport à l'arriéré de traitement. Aussi, la technique d'évaluation des délais de bout en bout dans les systèmes distribués basés sur un réseau Ethernet commuté que nous allons proposer se fonde sur les résultats présentés dans (Cruz, 1991*a*)(Cruz, 1991*b*). Le paragraphe suivant présente la courbe d'arrivée qui sera par la suite utilisée. Cette courbe sera notamment choisie pour ses propriétés mathématiques de simplification des calculs, de prise en compte de la spécificité des communications dans les SCRs et enfin pour son intégration des phénomènes sources de la congestion.

3.3 Modélisation du trafic : identification d'une courbe d'arrivée basée sur le volume des rafales

Voyons maintenant comment une courbe d'arrivée peut être retenue en vue de l'identification d'un flux. La méthode varie suivant les informations disponibles *a priori*.

3.3.1 Généralités

Dans le cas d'un *trafic périodique*, la fonction la plus proche reste la fonction en escalier (figure 3.11) définie par $\alpha(t) = v_{T,\tau}(t) = \lceil (t+\tau)/T \rceil$.

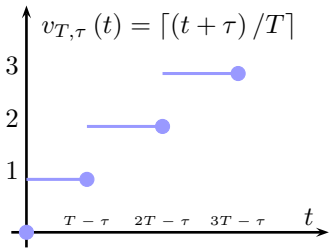


FIG. 3.11: Courbe d'arrivée en escalier

A partir de là, il s'agit de s'intéresser à la taille des messages. Dans le graphe de la figure 3.11, cette taille est supposée constante et égale à une unité de données. Si elle varie indépendamment, il sera

nécessaire d'utiliser la taille maximale. A partir de cette première considération, il convient également d'associer la durée de la période T. Afin de prendre en compte une éventuelle gigue, la courbe en escalier inclut une tolérance τ qui traduit une avance de l'arrivée de l'instance suivante. Cette tolérance devra donc être préalablement évaluée. On le voit donc bien ici, il s'agit une fois la fonction choisie, d'identifier chacun des paramètres sous l'hypothèse déterministe de limitation de l'arrivée des données. La courbe en escalier n'est toutefois pas la seule possible, et peut être remplacée par une fonction affine. Ainsi, si un flux R est contraint par une courbe d'arrivée en escalier $\alpha(t) = kv_{T,\tau}(t)$ où k correspond à la taille constante ou maximale des paquets, alors R est également contraint par la courbe d'arrivée affine $\gamma_{r,b}(t)$ telle que :

$$\alpha(t) = \gamma_{r,b}(t) = \left(\frac{k}{T}\right) t + k\frac{(\tau + T)}{T}$$

La figure 3.12 montre cette transposition entre fonctions affine et en escalier.

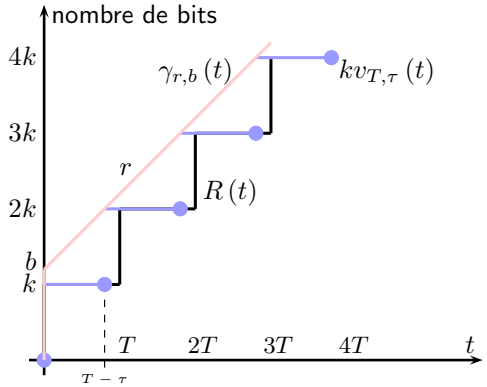


FIG. 3.12: Courbes d'arrivée possibles pour un flux périodique

Si l'imprécision sur la gigue est telle que la tolérance τ vaut une demi fois la période, la majoration de l'arrivée conduira à considérer qu'à l'instant $t = T$, deux paquets sont arrivés augmentant ainsi la valeur à l'origine de la fonction affine. Néanmoins, le fait que l'enveloppe affine présente comme on le verra par la suite des propriétés intéressantes lors du calcul, conduira à rendre son utilisation plus fréquente.

Si la seule information dont on dispose correspond cette fois à une *trace* ou observation de $R(t)$, on peut obtenir une courbe d'arrivée *minimale* à partir de cette trace. (Le Boudec et Thiran, 2001) montrent ainsi que la fonction $R \oslash R$ est une courbe d'arrivée minimale du flux R. Néanmoins, il est évident que la courbe dépend de l'observation. D'ailleurs, il n'est pas forcément intéressant d'obtenir l'optimum, mais on recherchera un optimum local, c'est-à-dire une courbe minimale appartenant à une famille de fonctions aux particularités (de calcul) intéressantes, comme par exemple les fonctions affines (voir (Naudts, 2000)).

Dans le cas d'un échange unitaire, c'est-à-dire seulement constitué d'un message, une modélisation établie autour d'une fonction échelon pourra être utilisée. La hauteur de l'échelon k permettra de définir $ku_T(t)$ ou T permet d'inclure la date d'arrivée.

Pour les autres trafics, il s'agira en fait d'utiliser une fonction de son choix permettant d'inclure directement les informations connues. Par exemple, dans le cas du contrôle d'admission, la courbe sera directement donnée par la politique de régulation.

Enfin rappelons que pour améliorer la précision de la modélisation de l'arrivée des données, on peut combiner différentes courbes répondant chacune à une information précise. Cette remarque peut se tra-

duire pour deux contraintes affines : si un flux est contraint par deux courbes affines $\gamma_{r_1,b_1}(t)$ et $\gamma_{r_2,b_2}(t)$, alors ce flux est contraint par

$$\alpha(t) = \min\{\gamma_{r_1,b_1}(t), \gamma_{r_2,b_2}(t)\} = (r_1 t + b_1) \wedge (r_2 t + b_2)$$

3.3.2 Spécificités des systèmes contrôlés en réseau

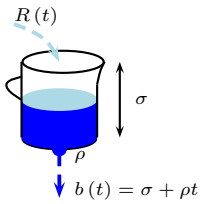
Dans le cas général pour lequel aucun protocole temps-réel est implémenté, la notion de courbe d'arrivée est associée à des régulateurs de trafic dans la mesure où le trafic est inconnu et doit être limité ((Caponetto *et al.*, 2002) utilise ainsi des régulateurs (σ, ρ) pour le trafic Internet).

Dans le contexte des systèmes contrôlés en réseau, deux classes de communications peuvent être identifiées. La première regroupe les échanges périodiques (souvent relatifs au contrôle : transport de la commande et des mesures) qui sont exactement connus en terme de volume et de fréquence puisqu'ils sont défini par l'application. La seconde classe de trafic représente le trafic apériodique qui est de plus en plus important dans les nouveaux systèmes industriels. Ainsi, un opérateur de maintenance intervenant dans le cadre de la réparation d'une machine manufacturière pourra télécharger le guide technique depuis une station à proximité. Ce document enrichit le type de données échangées (images, voix, vidéo ...) et surcharge le support de transmission.

Cette taxonomie n'est pas basée sur la contrainte temporelle associée au trafic, mais sur le niveau de connaissance des informations échangées sur le réseau. Outre les échanges périodiques, d'autres communications, comme les alarmes, sont à temps-critique. En supposant que les alarmes ne sont pas de type événementiel mais sont liées à une supervision périodique d'un état, il est possible de regrouper dans une première classe tout le trafic temps-réel (contrôle du process industriel et gestion des alarmes).

Le trafic de contrôle du système comme les messages échangés par les automates à chaque temps de cycle s'apparente donc à un trafic périodique. Il est alors possible d'envisager dans un premier temps une fonction en escalier comme courbe d'arrivée. Cette fonction sera toutefois remplacée par une fonction affine qui comme nous l'avons vu peut être aisément obtenue. L'intérêt sera notamment une simplification des calculs et des expressions des retards.

Les contraintes de courbes d'arrivée de trafic trouvent leur origine dans le concept du seau percé (*leaky bucket* dans la littérature anglophone) et des GCRA (Generic Cell Rate Algorithms) (Le Boudec, 1992). (Le Boudec et Thiran, 2001) montrent que les seaux percés correspondent en fait à des courbes d'arrivée affines et que les GCRA sont basés sur des fonctions en escalier. Le concept du leaky bucket controller (Le Boudec et Thiran, 2001, définition 1.3.2), illustré sur la figure 3.13, peut être utilisé pour modéliser la contrainte d'arrivée. Ce mécanisme de régulation de l'arrivée des données s'apparente au mécanisme présenté dans (Cohen *et al.*, 1991).



$$b(t) = \sigma + \rho t$$

$\forall x, y\,;\ y \geq x,\ x \geq 0$, alors
$R \sim b \Leftrightarrow R(y) - R(x) \leq \sigma + \rho(y - x)$

FIG. 3.13: Le concept de seau percé

La contrainte déterministe retenue implique que le nombre de bits émis par une source à un instant t

n'est pas supérieur à la valeur $b(t)$. Si le seau est plein, tous les messages supplémentaires seront écartés (perdus). $b(t)$ sera ainsi assimilé ($\sim$) à la borne supérieure du nombre de bits envoyés à t.

Le trafic périodique engendré par les systèmes contrôlés en réseau pourra aisément être modélisé par une courbe d'arrivée affine en utilisant les spécifications de l'application. Nos travaux se focalisent en priorité sur ce trafic. La prise en compte d'un éventuel trafic apériodique sera également modélisée par une fonction affine qui pourra être obtenue à partir d'informations supplémentaires ou de régulateurs de trafic comme les seaux percés. Dans le cas de l'utilisation de protocoles temps-réel, les paramètres σ et ρ de la fonction affine seront déterminés à partir des spécificités du protocole. Par exemple, dans le protocole RSVP ; la réservation de mémoire dans les équipements correspond à la détermination de σ. D'autant plus que comme (Le Boudec et Thiran, 2001) l'explique, de tels protocoles sont également basés sur des contraintes d'arrivés.

La courbe d'arrivée qui sera utilisée dans la suite du document sera donc de type fonction affine. Le paragraphe suivant met en relation les paramètres de cette fonction avec différentes propriétés des réseaux.

3.3.3 Notion de rafale, ou avalanche de données

Le paragraphe 3.1 nous a permis de noter que le délai de traversée d'un système était lié à l'arriéré de traitement de ce système. L'arriéré dépend quant à lui du rapport entre le volume de données reçues en entrée et la capacité du système. Autrement dit, plus le volume en entrée sera important, plus l'arriéré sera grand. Il est alors intéressant de noter que l'entrée va varier et des pics de données vont pouvoir apparaître. Comme le montre la figure 3.1, ces pics sont consécutifs à une arrivée continue et successive de données : les rafales ou avalanches.

Cette analyse conduite par Cruz l'a conduit à caractériser l'arrivée de donnés par une contrainte de la rafale maximale des données (*burstiness constraint* en anglais). Cette contrainte est similaire au seau percé, excepté que les paquets non conformes sont placés en attente et non simplement écartés. Ceci implique que σ correspond à la quantité de données maximale contenue dans une rafale et ρ au taux moyen d'arrivée des données. Un flux de données sera donc décrit par ces deux caractéristiques, et nous nous intéresserons plus particulièrement à l'évolution des rafales tout au long du réseau.

Enfin, l'arrivée de données contenues dans une rafale reste toutefois limitée par la capacité C du lien en entrée du système. D'où une seconde contrainte dite de stabilité (Cruz, 1991*a*), telle que $b(x) \leq Cx$. Cette correction est nécessaire pour tenir compte du fait qu'à l'instant $t = 0_+$, $R(0_+) \ll \sigma$. En utilisant la propriété de combinaison des courbes d'arrivée (paragraphe 3.3.1), la courbe d'arrivée $b(t)$ (borne supérieure du nombre de bits envoyés à l'instant t) sera donc exprimée par :

$$b(t) = min\{Ct, \sigma + \rho t\} \tag{3.9}$$

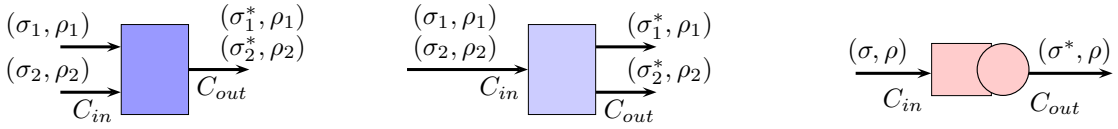
La courbe d'arrivée de l'équation (3.9) va maintenant être utilisée pour exprimer un majorant du délai de traversée de chaque composant élémentaire des modèles de commutateur. Nous utiliserons pour cela les travaux développés dans (Cruz, 1991*a*).

3.4 Etude des délais de composants FIFO

Au paragraphe 3.2.6, nous avons introduit que du fait qu'Ethernet n'inclut pas de garantie de service, les expressions des majorants du délai ne seront pas directement considéré par rapport à la courbe de service, mais par rapport à l'arriéré de traitement. Dans ce cas, le délai correspond au temps de traitement de l'arriéré et le majorant est obtenu par la division du pire arriéré sur le service de traitement offert par le système. L'expression du pire arriéré s'apparente alors à une recherche du pire cas.

3.4.1 Introduction

Ce paragraphe présente les expressions de majorants du délai en fonction des rafales des différents flux. Le principe général consiste dans un premier temps à la détermination d'un majorant de l'arriéré de traitement introduit par les composants identifiés à la figure 3.14.



(a) Multiplexeur (b) Démultiplexeur (c) File

FIG. 3.14: Composants élémentaires du réseau

Nous utiliserons pour cela les notations du tableau 3.1, ainsi que les expressions notamment présentées dans (Georges *et al.*, 2003*a*),(Georges *et al.*, 2005*a*).

TAB. 3.1: Notations

symbole	*définition*	*unités*
$b_i(t)$	courbe d'arrivée du flux i à l'instant t	bits
σ_i	avalanche maximale du flux i	bits
ρ_i	taux moyen d'arrivée des données pour le flux i	bits/sec
L	longueur maximale des trames (1526 *octets*)	bits
L_i	longueur maximale des trames du flux i	bits
C_{in}	capacité des liens en entrées	bits/sec
C_i	capacité du port d'entrée i	bits/sec
C_{out}	capacité des ports de sortie	bits/sec
$x(t)$	arriéré de traitement à l'instant t	bits
$D(t)$	délai de traversée à l'instant t	sec
$\overline{D_i}$	majorant du délai de traversée des trames du flux i	sec
u_i	durée de l'avalanche de données maximale du flux i	sec

Dans cette partie, il est supposé que les systèmes étudiés sont à politique non-oisive (*work-conserving* dans la littérature anglophone), c'est-à-dire sans vacations[1]. Si $x(t)$ représente la valeur de l'arriéré de traitement à un instant t, cela signifie que la vitesse instantané de sortie à l'instant t correspond à la

[1] vacation : intervalle de temps pendant lequel la quantité de donnée servie est nulle alors que l'arriéré de traitement est non nul ($\exists t \in [t_1, t_2] / x(t) > 0$ et $R^*(t) = R^*(t_1)$)

capacité de sortie du système C_{out}.

$$x(t) > 0 \Rightarrow (R^*)'(t) = C_{out}$$

La politique de retransmission considérée dans ce chapitre est la politique *premier arrivé, premier sorti*, abrégée FIFO.

La *stabilité* des systèmes est également supposée. Cette stabilité correspond à la non saturation (ou non blocage d'un commutateur - voir page 54). Pour un système à deux entrées, la condition suffisante généralement retenue définit que :

$$\lim_{x\to\infty} b_1(x) + b_2(x) - C_{out}x = -\infty$$

Etudions maintenant les traversées des trois composants élémentaires illustrés à la figure 3.14. Le premier sera la file FIFO.

3.4.2 Système à file d'attente

Nous nous plaçons ici dans le contexte général d'une file d'attente à une entrée et une sortie 3.14(c). Pour une file FIFO recevant les données au taux $R'(t)$ et les retransmettant au taux C_{out}, nous avons vu au paragraphe 3.1.4 que l'arriéré de traitement est défini par :

$$W_{C_{out}}(R')(t) = \max_{s\leq t}[R(t) - R(s) - C_{out}(t-s)]$$

La relation entre le délai et l'arriéré est alors donnée (Cruz, 1991*a*) par :

$$d(t) = \frac{1}{C_{out}} W_{C_{out}}(R')(t)$$

En appliquant le majorant de l'arriéré de traitement défini à l'équation (3.6), nous avons donc directement :

$$d(t) \leq \frac{1}{C_{out}} \max_s \{\alpha(s) - \beta(s)\}$$

La courbe d'arrivée considérée à l'équation (3.9) nous donne donc en notant C_{in} la capacité du lien en entrée de la file,

$$d(t) \leq \frac{1}{C_{out}} \max_s \{\min\{C_{in}s; \sigma + \rho s\} - C_{out}s\}$$

Puisque C_{out} est fixe, cela dépend simplement de l'évolution de la courbe d'arrivée $b(u)$. Comme

$$\frac{db}{du}(u) = \begin{cases} C_{in}, & \text{si } u < \frac{\sigma}{C_{in}-\rho} \\ \rho, & \text{si } u > \frac{\sigma}{C_{in}-\rho} \end{cases}$$

La partie de droite de l'inéquation sera maximale pour $s = \frac{\sigma}{C_{in}-\rho}$. Et comme à cet instant $C_{in}s = \sigma + \rho s$, on obtient finalement le délai maximal de traversée de la file :

$$\begin{aligned} d(t) &\leq \frac{1}{C_{out}} \left[C_{in}\left(\frac{\sigma}{C_{in}-\rho}\right) - C_{out}\left(\frac{\sigma}{C_{in}-\rho}\right)\right] \\ &\leq \frac{1}{C_{out}} \frac{(C_{in}-C_{out})}{C_{in}-\rho}\sigma = \overline{D_{file}} \end{aligned}$$

Cette dernière ligne n'est valable que sous l'hypothèse que $C_{in} \geq C_{out}$. Il est bien entendu que si $C_{in} \leq C_{out}$, le système considéré n'engendrera aucun délai ($\overline{D} = 0$). Nous rappelons également que ce résultat

est obtenu sous les hypothèses suivantes : $C_{in} \geq \rho$ (le taux d'arrivée des données est limité par la capacité du lien en entrée), $C_{out} \geq \rho$ (le système est suffisament dimensionnée), la taille du buffer est supérieure à la valeur maximale de l'arriéré de traitement et que l'arrivée des données $R(t)$ est contraint par $R(y) - R(x) \leq \min\{C_{in}(y-x), \sigma + \rho(y-x)\}$.

Ce même résultat peut être obtenu à partir des expressions fournies notamment dans (Le Boudec, 1996) (voir début du chapitre). Dans ce cas, le délai correspond à la distance horizontale entre les fonctions d'arrivée et de départ du trafic.

$$d(t) \leq \inf\{T : T \geq 0, R(t) \leq R^*(t+T)\}$$

Comme nous l'avons vu (figure 3.2, 52), ce délai est majoré par la distance horizontale entre la courbe d'arrivée et la courbe de service minimal offert par le système.

$$\forall t, \quad d(t) \leq \sup_{s \geq 0}\{\inf\{T : T \geq 0, \alpha(s) \leq \beta(s+T)\}\}$$

Dans notre cas d'étude, la courbe d'arrivée est liée à la contrainte d'avalanche des données et est limité par la capacité du port d'entrée C_{in}. Le service offert par la file d'attente est non-oisif et est liée au taux de sortie C_{out} comme indiqué sur la figure 3.15.

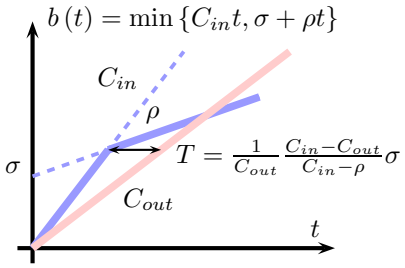


FIG. 3.15: Majoration du délai dans une file d'attente

Par conséquent, le délai est majoré par :

$$d(t) \leq \sup_{s \geq 0}\{\inf\{T : T \geq 0, \min(C_{in}s, \sigma + \rho s) \leq C_{out}(s+T)\}\}$$

Pour les mêmes raisons que dans l'analyse précédente, nous pouvons alors aisément identifier que la valeur de T qui majore $d(t)$ est définie par :

$$T = \frac{1}{C_{out}} \frac{C_{in} - C_{out}}{C_{in} - \rho}\sigma$$

Nous venons ici de majorer le délai de traversée dans le cas d'une file d'attente. Pour cela, nous avons introduit deux méthodes d'analyses en étudiant l'évolution du délai au travers de celle de l'arriéré de traitement ou alors en utilisant directement les résultats génériques du calcul réseau. Nous avons pu ainsi illustrer que l'on retrouve bien les mêmes résultats, ce qui nous permettra par la suite d'utiliser la technique la plus appropriée.

3.4.3 Multiplexage

Le multiplexeur (figure 3.14(a)) est un système à file d'attente particulier qui comporte plusieurs liens d'entrées. (Cruz, 1991*a*) présente une analyse des multiplexeurs selon leur politique de retransmission. Il note ainsi que pour un multiplexeur général avec vacations V, le délai est majoré pour deux entrées par :

$$\overline{D} = \sup\left\{\alpha : \alpha \geq 0,\ b_1(\alpha) + b_2(\alpha) + C_{out}V \geq C_{out}\alpha\right\}$$

Dans le cas de courbe d'arrivée (σ, ρ) et sous l'hypothèse $\rho_1 + \rho_2 < C_{out}$, l'expression est simplifiée comme suit.

$$\overline{D} = \frac{\sigma_1 + \sigma_2 + C_{out}V}{C_{out} - \rho_1 - \rho_2}$$

Cette expression est ainsi utilisée par (Grieu, 2004) pour caractériser le délai de traversée d'un multiplexeur sans vacations ($x(t) > 0 \Rightarrow (R^*)'(t) = C_{out}$). (Cruz, 1991*a*) propose également une expression du majorant du délai dans le cas où la politique est FIFO. Le délai subi par les données d'un flux qui arrive sur le port 1 est majoré par :

$$\overline{D_{mux,1}} = \frac{1}{C_{out}} \max_{u \geq 0} \left\{ b_1(u) + b_2\left(u + \frac{L_2}{C_2}\right) - C_{out}u \right\} \tag{3.10}$$

Dans cette équation (illustrée à la figure 3.16), la valeur maximale de l'arriéré de traitement (quantité de données arrivées moins la quantité déjà retransmise) est divisée par le taux de retransmission du multiplexeur. Puisque la politique de retransmission est FIFO et que nous nous intéressons au pire cas, il est supposé ici que si deux trames entrent simultanément par les ports 1 et 2, la trame issue du port 2 sera la première à être traitée. Ce qui explique le terme $\frac{L_2}{C_2}$.

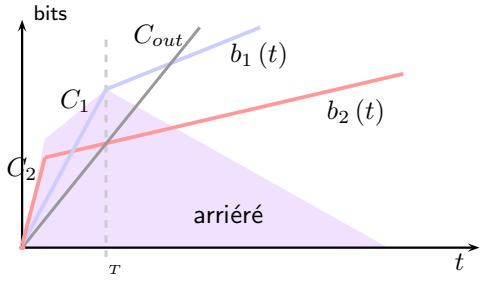
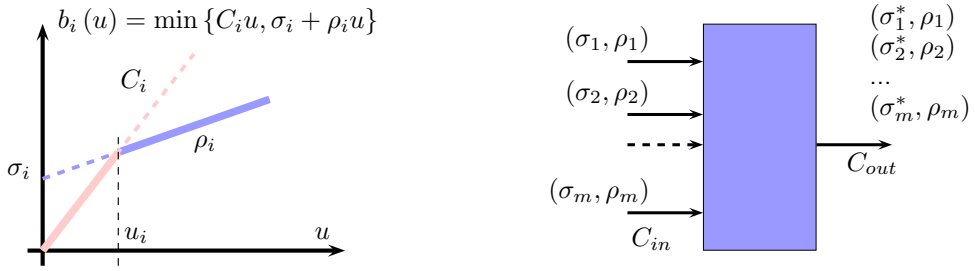


FIG. 3.16: Arriéré de traitement d'un multiplexeur FIFO à 2 entrées

Notons qu'à $t = 0_+$, la quantité des données arrivées *via* le port 1 sera limitée par $C_1 t$. La figure 3.16 montre alors que la valeur de l'arriéré est maximale à l'instant $T = \frac{\sigma_1}{C_1 - \rho_1}$. En remplaçant dans (3.10) les fonctions d'arrivée $b(t)$ par les contraintes d'avalanche de données associées (voir équation (3.9)), la formulation du majorant peut être réécrite sous l'hypothèse $C_i = C_{out}$ ($C_1 = C_2 = C_{out} = C$) comme suit (Cruz, 1991*a*, théorème 4.1) :

$$\overline{D_{\text{mux},1}} = \frac{1}{C} \min \begin{cases} \sigma_1 + \rho_1 \frac{\sigma_2}{C - \rho_2} + (C - \rho_1)\frac{L}{C} & \text{si } \left(\frac{\sigma_2}{C-\rho_2} - \frac{L}{C}\right) > \frac{\sigma_1}{C-\rho_1} \\ \sigma_2 + \rho_2 \frac{\sigma_1}{C - \rho_1} + \rho_2 \frac{L}{C} & \text{sinon} \end{cases}$$

Nous allons proposer une extension de cette formulation pour un multiplexeur à m entrées dans le cadre générique où les flux de données en entrée et les capacités d'entrée et de sortie peuvent être différents.



(a) Courbe d'arrivée des flux.

(b) Multiplexeur FIFO à m entrées.

FIG. 3.17: Paramètres d'entrée et de sortie du multiplexeur.

Cette expression sera établie pour le flux empruntant le port en entrée 1. Le trafic arrivant sur chaque port d'entrée est contraint par la courbe d'arrivée $b_i(u) = \min\{C_i u; \sigma_i + \rho_i u\}$ comme indiqué à la figure 3.17(a).

Dans la suite du document, nous utiliserons les notions suivantes :

arriéré de traitement : quantité de données (travail) en attente dans le multiplexeur, c'est-à-dire, la quantité de données reçues moins la quantité de données retransmises à un instant donné (en bits).

avalanche (rafale) de données : quantité de données pouvant arriver successivement sans rupture (en bits).

durée de l'avalanche de données : intervalle de temps durant lequel des données arrivent les unes après les autres sans discontinuité (en secondes).

Nous supposerons également :

- $C_i \geq C_{out}$. Cas défavorable puisqu'il aggrave les effets des avalanches de données sur l'arriéré de traitement du multiplexeur et augmente ainsi sa congestion.
- $\rho_1 + \rho_2 + \ldots + \rho_m < C_{out}$. Hypothèse de non saturation : on suppose que la quantité de données reçues sera inférieure à la capacité du multiplexeur. Par conséquent, $\rho_i < C_{out}$.

A partir de là, la première étape consiste à rechercher un ou plusieurs majorants du délai de traversée d'un multiplexeur FIFO.

Majorants du délai de traversée

Pour un multiplexeur FIFO à deux entrées, Cruz a montré que le délai de traversée du multiplexeur pour un bit arrivant sur le port 1 est majoré par l'équation (3.10).

$$\overline{D_{mux,1}} = \frac{1}{C_{out}} \max_{u \geq 0} \left\{ b_1(u) + b_2\left(u + \frac{L_2}{C_2}\right) - C_{out} u \right\}$$

Il est alors aisé de développer cette équation pour m entrées.

$$\overline{D_{mux,1}} = \frac{1}{C_{out}} \max_{u \geq 0} h_u$$

Le délai est ici défini comme le temps nécessaire au traitement de l'arriéré. L'arriéré de traitement $h(u)$ correspond à la différence entre la somme des arrivées $b_i(u)$ moins la sortie $C_{out} u$. On peut écrire :

$$h(u) = b_1(u) + b_2\left(u + \frac{L_2}{C_2}\right) + \ldots + b_m\left(u + \frac{L_m}{C_m}\right) - C_{out} u \tag{3.11}$$

La courbe d'arrivée d'un flux i est définie par la fonction convexe à deux parties $b_i(u)$ telle que $b_i(u) = \min\{C_i u; \sigma_i + \rho_i u\}$ où C_i correspond à la capacité du port i avec $C_i \geq C_{out}$ et ρ_i au débit d'arrivée des données avec $\rho_i < C_i$. L'arriéré $h(u)$ vaut donc :

$$\begin{aligned} h(u) = & \min\{C_1 u; \sigma_1 + \rho_1 u\} \\ & + \min\left\{C_2\left(u + \frac{L_2}{C_2}\right); \sigma_2 + \rho_2\left(u + \frac{L_2}{C_2}\right)\right\} \\ & + ... \\ & + \min\left\{C_m\left(u + \frac{L_m}{C_m}\right); \sigma_m + \rho_m\left(u + \frac{L_m}{C_m}\right)\right\} - C_{out} u \end{aligned}$$

Il est alors possible de différencier l'expression de l'arriéré $h(u)$ suivant u. En effet, la courbe d'arrivée d'un flux i peut s'écrire :

$$b_i(u) = \begin{cases} C_i u & \text{si } u \leq \frac{\sigma_i}{C_i - \rho_i} \\ \sigma_i + \rho_i u & \text{sinon} \end{cases}$$

Soient $u_1 = \frac{\sigma_1}{C_1 - \rho_1}$ et $\forall i$, $u_i = \frac{\sigma_i}{C_i - \rho_i} - \frac{L_i}{C_i}$. Nous avons alors :

$$\begin{aligned} h(u) &= \begin{cases} C_1 u + \min\left\{C_2\left(u + \frac{L_2}{C_2}\right); \sigma_2 + \rho_2\left(u + \frac{L_2}{C_2}\right)\right\} + ... & \text{si } u \leq u_1 \\ \sigma_1 + \rho_1 u + \min\left\{C_2\left(u + \frac{L_2}{C_2}\right); \sigma_2 + \rho_2\left(u + \frac{L_2}{C_2}\right)\right\} + ... & \text{sinon} \end{cases} \\ &= \begin{cases} C_1 u + C_2\left(u + \frac{L_2}{C_2}\right) + ... & \text{si } u \leq u_1 \text{ et } u \leq u_2 \\ C_1 u + \sigma_2 + \rho_2\left(u + \frac{L_2}{C_2}\right) + ... & \text{si } u \leq u_1 \text{ et } u > u_2 \\ \sigma_1 + \rho_1 u + C_2\left(u + \frac{L_2}{C_2}\right) + ... & \text{si } u > u_1 \text{ et } u \leq u_2 \\ \sigma_1 + \rho_1 u + \sigma_2 + \rho_2\left(u + \frac{L_2}{C_2}\right) + ... & \text{sinon} \end{cases} \end{aligned}$$

On voit alors bien qu'en poursuivant la décomposition précédente de l'équation (3.5), 2^m expressions de $h(u)$ peuvent être identifiées.

$$\begin{aligned} h_1(u) &= C_1 u + \sum_{i=2}^{m} C_i\left(u + \frac{L_i}{C_i}\right) \\ h_2(u) &= \sigma_1 + \rho_1 u + \sum_{i=2}^{m} C_i\left(u + \frac{L_i}{C_i}\right) \\ &... \\ h_{2^m - m}(u) &= C_1 u + \sum_{i=2}^{m}\left(\sigma_i + \rho_i\left(u + \frac{L_i}{C_i}\right)\right) \\ &... \\ h_{2^m - 1}(u) &= \sigma_1 + \rho_1 u + \sum_{i=2}^{m-1}\left(\sigma_i + \rho_i\left(u + \frac{L_i}{C_i}\right)\right) + C_m\left(u + \frac{L_m}{C_m}\right) \\ h_{2^m}(u) &= \sigma_1 + \rho_1 u + \sum_{i=2}^{m}\left(\sigma_i + \rho_i\left(u + \frac{L_i}{C_i}\right)\right) \end{aligned}$$

En utilisant les propriétés de distributivité de + suivant min et d'associativité de + et min, notons que :

$$h(u) = \min\{h_1(u), h_2(u), ..., h_{2^m}(u)\} - C_{out} u \tag{3.12}$$

L'expression précédente de $h(u)$ permet de dégager rapidement plusieurs majorants du délai. En effet, puisque :

$$\min\{h_1(u), h_2(u), ..., h_{2^m}(u)\} \leq \min\{h_{2^m-m}(u), h_{2^m}(u)\}$$

nous obtenons :

$$\overline{D_{mux,1}} \leq \frac{1}{C_{out}} \max_{u\geq 0}\{\min\{h_{2^m-m}(u), h_{2^m}(u)\} - C_{out}u\}$$

Comme l'une des hypothèses impose que $\rho_1 < C_{out}$, $\overline{D_{mux,1}}$ sera maximum pour $u = \frac{\sigma_1}{C_1-\rho_1}$, d'où :

$$\overline{D_{mux,1}} \leq \frac{1}{C_{out}}\left[\sum_{i=2}^{m}\left(\sigma_i + \rho_i\left(\frac{\sigma_1}{C_1-\rho_1} + \frac{L_i}{C_i}\right)\right) + (C_1 - C_{out})\frac{\sigma_1}{C_1-\rho_1}\right] \tag{3.13}$$

L'inéquation (3.13) nous donne un premier majorant du délai. Néanmoins, d'autres majorants sont possibles à partir de l'équation de référence (3.12). En effet,

$$\begin{aligned}
\overline{D_{mux,1}} &\leq \frac{1}{C_{out}} \max_{u\geq 0}\{\min\{h_{2^m-m+1}(u), h_{2^m}(u)\} - C_{out}u\} \\
&\leq \frac{1}{C_{out}}\left[\begin{array}{l} \sum_{i=1;i\neq 2}^{m}\left(\sigma_i + \rho_i\left(\frac{\sigma_2}{C_2-\rho_2} - \frac{L_2}{C_2} + \frac{L_i}{C_i}\right)\right) \\ + (C_2 - C_{out})\left(\frac{\sigma_2}{C_2-\rho_2} - \frac{L_2}{C_2}\right) - \rho_1\frac{L_1}{C_1} + L_2 \end{array}\right] \\
&... \\
\overline{D_{mux,1}} &\leq \frac{1}{C_{out}} \max_{u\geq 0}\{\min\{h_{2^{m+1}-1}(u), h_{2^{m+1}}(u)\} - C_{out}u\} \\
&\leq \frac{1}{C_{out}}\left[\begin{array}{l} \sum_{i=1;i\neq m}^{m}\left(\sigma_i + \rho_i\left(\frac{\sigma_m}{C_m-\rho_m} - \frac{L_m}{C_m} + \frac{L_i}{C_i}\right)\right) \\ + (C_m - C_{out})\left(\frac{\sigma_m}{C_m-\rho_m} - \frac{L_m}{C_m}\right) - \rho_1\frac{L_1}{C_1} + L_m \end{array}\right]
\end{aligned}$$

Tous ces (m) majorants sont réunis en un seul en prenant le plut petit d'entre eux. Nous proposons donc :

$$\overline{D_{mux,1}} \leq \frac{1}{C_{out}} \min_k \overline{B_k} \tag{3.14}$$

où $\overline{B_k}$ correspond aux différents majorants de l'arriéré, c'est-à-dire :

$$\begin{aligned}
\overline{B_1} &= \sum_{i=2}^{m}\left(\sigma_i + \rho_i\left(\frac{\sigma_1}{C_1-\rho_1} + \frac{L_i}{C_i}\right)\right) + (C_1 - C_{out})\frac{\sigma_1}{C_1-\rho_1} \\
\forall\ 1 < k \leq m,\ \overline{B_k} &= \sum_{i=1;i\neq k}^{m}\left(\sigma_i + \rho_i\left(\frac{\sigma_k}{C_k-\rho_k} - \frac{L_k}{C_k} + \frac{L_i}{C_i}\right)\right) \\
&\quad + (C_k - C_{out})\left(\frac{\sigma_k}{C_k-\rho_k} - \frac{L_k}{C_k}\right) - \rho_1\frac{L_1}{C_1} + L_k
\end{aligned}$$

Minorants du délai maximum

Nous avons vu (équation (3.5)) que l'arriéré de traitement $h(u)$ ne dépend que des courbes d'arrivées des données en entrée.

$$h(u) = b_1(u) + b_2\left(u + \frac{L_2}{C_2}\right) + ... + b_m\left(u + \frac{L_m}{C_m}\right) - C_{out}u$$

Pour chaque port d'entrée, le flux entrant est contraint par $b_i(u) = \min\{C_i u; \sigma_i + \rho_i u\}$. Par conséquent, les variations des arrivées sont définies telles que :

$$\frac{db_i}{du}(u) = \left\{\begin{array}{ll} C_i & \text{si } u \leq \frac{\sigma_i}{C_i-\rho_i} \\ \sigma_i + \rho_i & \text{sinon} \end{array}\right.$$

Dans la première partie, l'arrivée des données est permanente, mais limitée par le débit du port d'entrée C_i pour lequel on a supposé que $C_i \geq C_{out}$. Dans ce cas, la charge est incessante et conduit à une augmentation de l'arriéré de traitement. Dans la seconde partie, l'avalanche de données est terminée, et le multiplexeur va pouvoir commencer à éliminer l'arriéré en attente puisque l'hypothèse de non saturation suppose que $\rho_i < C_{out}$.

Puisque l'hypothèse de non saturation conduit également à ce que $\rho_1 + \rho_2 + ... + \rho_m < C_{out}$, la taille de l'arriéré augmente jusqu'à ce que l'avalanche la plus importante soit totalement traitée. Dans la mesure où nous considérons un multiplexeur à m entrées, il y a m possibilités. Soit u_i *la durée d'avalanche de données du flux* i tel que :

$$u_1 = \frac{\sigma_1}{C_1 - \rho_1}$$
$$\forall i,\ 1 < i \leq m,\ u_i = \frac{\sigma_i}{C_i - \rho_i} - \frac{L_i}{C_i}$$

Nous avons donc :

$$\overline{D_{mux,1}} = \frac{1}{C_{out}} \max\left\{ h\left(\frac{\sigma_1}{C_1 - \rho_1}\right), h\left(\frac{\sigma_2}{C_2 - \rho_2} - \frac{L_2}{C_2}\right), ..., h\left(\frac{\sigma_m}{C_m - \rho_m} - \frac{L_m}{C_m}\right) \right\}$$
$$= \begin{cases} \frac{1}{C_{out}} \left[\begin{array}{l} \sum_{i=2}^{m} \left(\sigma_i + \rho_i \left(\frac{\sigma_1}{C_1-\rho_1} + \frac{L_i}{C_i}\right)\right) \\ + (C_1 - C_{out}) \frac{\sigma_1}{C_1-\rho_1} \end{array} \right] & \text{si } u_1 > \max_{k\neq 1} u_k \\ \frac{1}{C_{out}} \left[\begin{array}{l} \sum_{i=1;i\neq 2}^{m} \left(\sigma_i + \rho_i \left(\frac{\sigma_2}{C_2-\rho_2} - \frac{L_2}{C_2} + \frac{L_i}{C_i}\right)\right) \\ + (C_2 - C_{out}) \frac{\sigma_2}{C_2-\rho_2} - \rho_1 \frac{L_1}{C_1} + C_{out} \frac{L_2}{C_2} \end{array} \right] & \text{si } u_2 = \max u_k \\ ... & \\ \frac{1}{C_{out}} \left[\begin{array}{l} \sum_{i=1;i\neq m}^{m} \left(\sigma_i + \rho_i \left(\frac{\sigma_m}{C_m-\rho_m} - \frac{L_m}{C_m} + \frac{L_i}{C_i}\right)\right) \\ + (C_m - C_{out}) \frac{\sigma_m}{C_m-\rho_m} - \rho_1 \frac{L_1}{C_1} + C_{out} \frac{L_m}{C_m} \end{array} \right] & \text{si } u_m = \max u_k \end{cases}$$

On peut alors constater que le délai maximum de traversée d'un multiplexeur FIFO à m entrées est donc supérieur ou égal à chacun des différents cas. Nous retrouvons donc la formulation complémentaire à l'équation (3.14).

$$\overline{D_{mux,1}} \geq \frac{1}{C_{out}} \min_k \overline{B_k} \tag{3.15}$$

Conclusion

La réunion des équations (3.14) et (3.15) nous donne donc la proposition suivante. Le délai de traversée d'un multiplexeur FIFO à m entrées d'un flux arrivant sur le port 1 est défini par :

$$\overline{D_{mux,1}} = \frac{1}{C_{out}} \min_k \overline{B_k}$$

où $\overline{B_k}$ correspond aux différents majorants de l'arriéré, c'est-à-dire :

$$\overline{B_1} = \sum_{i=2}^{m} \left(\sigma_i + \rho_i \left(\frac{\sigma_1}{C_1 - \rho_1} + \frac{L_i}{C_i}\right)\right) + (C_1 - C_{out}) \frac{\sigma_1}{C_1 - \rho_1}$$
$$\forall\ 1 < k \leq m,\ \overline{B_k} = \sum_{i=1;i\neq k}^{m} \left(\sigma_i + \rho_i \left(\frac{\sigma_k}{C_k - \rho_k} - \frac{L_k}{C_k} + \frac{L_i}{C_i}\right)\right)$$
$$+ (C_k - C_{out}) \left(\frac{\sigma_k}{C_k - \rho_k} - \frac{L_k}{C_k}\right) - \rho_1 \frac{L_1}{C_1} + L_k$$

Cette relation peut également s'exprimer de la façon suivante :

$$\overline{D_{mux,1}} = \frac{1}{C_{out}} \min_k \overline{B_k}$$

où $\overline{B_k}$ correspond aux différents majorants de l'arriéré. Suivant que la durée de l'avalanche de données du flux considéré (ici, 1), notée u_1 est la plus longue, l'arriéré est majoré par :

$$\overline{B_1} = \sum_{i=2}^{m} \left(\sigma_i + \rho_i \left(u_1 + \frac{L_i}{C_i} \right) \right) + (C_1 - C_{out})\, u_1$$

avec la durée d'avalanche $u_1 = \frac{\sigma_1}{C_1 - \rho_1}$. Sinon, s'il s'agit de l'avalanche du flux k, $\forall\ 1 < k \leq m$, nous avons :

$$\overline{B_k} = \sum_{i=1; i \neq k}^{m} \left(\sigma_i + \rho_i \left(u_k + \frac{L_i}{C_i} \right) \right) + (C_k - C_{out})\, u_k - \rho_1 \frac{L_1}{C_1} + L_k$$

avec la durée d'avalanche $u_k = \frac{\sigma_k}{C_k - \rho_k} - \frac{L_k}{C_k}$.

La différence entre les deux expressions vient du fait que dans le second cas, on ajoute l'hypothèse pessimiste, qu'un paquet du flux considéré devra attendre le traitement d'un paquet arrivé simultanément.

3.4.4 Démultiplexage

Le démultiplexeur 3.14(b) aiguille les différents flux vers le port de sortie approprié. Dans les réseaux Ethernet, cette tâche revient à lire l'adresse de destination spécifiée dans la trame et à retrouver dans la table de retransmission le port de sortie associé. A ce niveau, le « routage » est supposé fixe. Aussi, il est admis que cette opération de redirection est accomplie instantanément et c'est pourquoi nous supposerons que $\overline{D_{\text{demux}}} = 0$.

La même hypothèse sera suggérée pour les liens d'interconnexion entre nœuds terminaux et commutateurs et de commutateurs à commutateur. Plus particulièrement, nous supposons que l'impact du temps de propagation est négligeable comparé au retard lié à la concurrence d'accès au medium. Il est également à noter que l'ensemble de notre étude se place dans le contexte où les pertes physiques de trames sont nulles. Cela sous-entend que le réseau est suffisamment dimensionné (cela suppose ainsi qu'aucun écartement de trames ne soit dû à la saturation d'un buffer) et que les perturbations physiques extérieures sont nulles.

3.5 Conclusion

Dans le chapitre 2, nous avons choisi de réduire le périmètre de cette thèse aux architectures Ethernet commutées. Nous avons également montré l'intérêt d'une évaluation de performances temporelles de ces architectures puisqu'elle n'offre aucune garantie sur les délais. Ce chapitre jette donc les bases d'une majoration des délais de bout en bout dans de tels environnements. Nous avons ainsi présenté la théorie sous-jacente à cette détermination, le calcul réseau. Classiquement, celle-ci s'appuie sur un modèle des données et un modèle des systèmes.

L'adaptation de cette théorie aux systèmes contrôlés en réseau basés sur Ethernet commuté nous a conduit à caractériser l'arrivée des données et les commutateurs. En ce qui concerne l'arrivée des données, nous avons choisi d'utiliser une courbe d'arrivée des données de type seau percé qui prend en compte la

contrainte d'avalanche maximale des données définies par Cruz. Compte tenu de la multitude des solutions techniques et fonctionnelles implémentées dans un commutateur, le cadre d'étude défini au chapitre 2 ne nous permet pas d'obtenir directement un modèle de commutateur. Aussi, nous avons cherché dans ce chapitre à représenter les différentes fonctionnalités. A ce stade, nous avons retenu trois modèles (qui seront comparés au chapitre suivant).

Si nos travaux mettent principalement l'accent sur une évaluation par calcul réseau, ceci est dû à la nécessité d'apporter des garanties temporelles de la traversée du réseau. Pour une configuration donnée, la démarche analytique nous parait donc être la plus appropriée à exprimer une garantie temporelle. Ceci n'exclut pas toutefois les autres possibilités d'étude, comme la simulation et le prototypage, qui doivent permettre de compléter l'information obtenue par calcul. Comme nous allons le voir au chapitre suivant, l'approche calcul réseau s'intéresse au pire cas et ne reflète donc forcément l'état du réseau le plus courant. Une démarche complémentaire de simulation peut permettre de comparer les majorants obtenus avec les valeurs de délai les plus fréquentes. L'approche par prototypage complète alors cette information dans la mesure où elle permet plus clairement de définir la fréquence d'apparition de ce pire cas ainsi que la précision des majorants obtenus par calcul réseau. La complémentarité de ces trois démarches permet alors à la personne en charge de l'exploitation du réseau de dimensionner au mieux le réseau et d'identifier plus finement les taux de charge des différents équipements. Le principal intérêt de cette démarche unifiée est de pouvoir clairement exprimer le coût de la contrainte de déterminisme (relativement forte dans le cadre des systèmes contrôlés en réseau) par rapport à l'utilisation du réseau (le dimensionnement d'un réseau en fonction du pire cas conduit à une sous-utilisation de celui-ci dans la majorité des cas).
Si nos travaux concernent uniquement une démarche analytique basée sur le calcul réseau pour les systèmes contrôlés en réseau, d'autres travaux (Branicky *et al.*, 2003) exploitent une démarche basée sur la simulation du réseau. Il serait alors intéressant par la suite d'identifier la conséquence sur la commande du système (à espace d'état continu ou à évennements discrets) de l'approche " *protectionniste* " du calcul réseau, notamment sur la réactivité de la commande.

Si l'on reprend alors l'état d'avancement illustré à la figure 2.10, nous pouvons remplacer les stations par des seaux percés et les commutateurs par un des modèles. La figure 3.18 exprime cette nouvelle vue du réseau à partir du modèle n°3. A ce stade, les communications sont donc définies par leur courbe d'arrivée initiale et les composants élémentaires des modèles de commutateur qui seront traversés.

La dernière partie de ce chapitre a permis de développer une expression d'un majorant du délai de traversée pour chacun des composants élémentaires inclus dans le modèle de commutateur. Comme le montre la figure 3.18, l'objectif du chapitre suivant sera alors de poursuivre cette évaluation en proposant des expressions de majorants du délai de traversée de bout en bout du réseau Ethernet commuté.

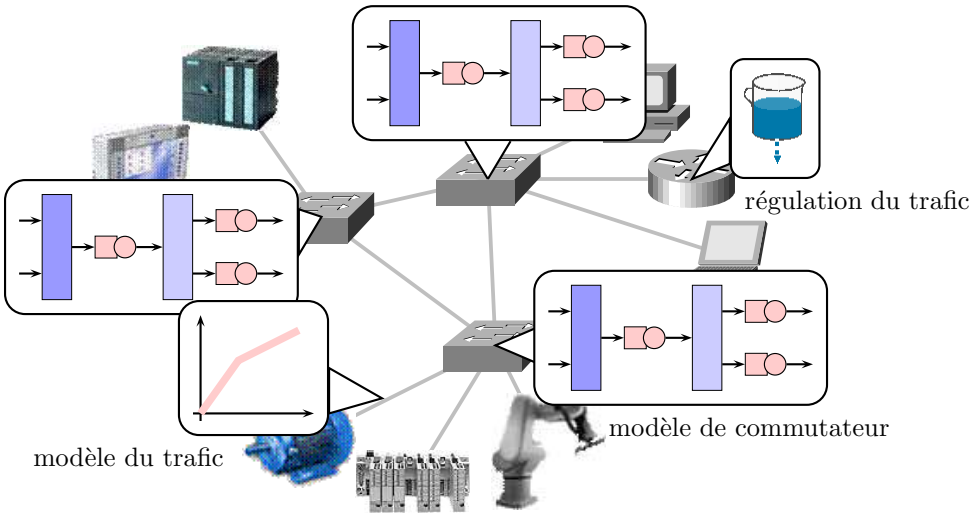


FIG. 3.18: Vue modélisée du système contrôlé en réseau

Chapitre 4

Détermination des délais maxima de bout en bout

Le concept de calcul réseau présenté au chapitre précédent repose principalement sur les notions d'arriéré de traitement, de courbes d'arrivée et de service. L'étude des spécificités des applications distribuées à temps critique nous a conduit à supposer que le trafic respecte une contrainte d'avalanche de données. Les délais de traversée de plusieurs composants élémentaires appliquant la politique FIFO ont été majorés à partir de la valeur maximale de l'arriéré de traitement. Cet arriéré dépend alors des valeurs des avalanches de données des flux tout au long du réseau. Dans ce chapitre, une méthodologie de calcul des délais de bout en bout est présentée. L'un des points fondamentaux fut illustré par (Cruz, 1991*b*) : la recherche d'un majorant du délai de bout en bout met en lumière une inter-dépendance de l'évolution de l'avalanche de données propre à chaque flux. Cette inter-dépendance devra alors être résolue et mettre en œuvre un calcul matriciel qui sera également présenté dans ce chapitre.

4.1 Méthodologie générale

La méthode de majoration des délais de bout en bout d'un réseau Ethernet commuté s'appuie dans un premier temps sur un résultat intermédiaire : la majoration de la sortie des données d'un système.

4.1.1 Courbe de départ $\alpha^*(t)$

Dans le chapitre précédent, des expressions de majorant du délai de traversée ont été données pour chaque composant élémentaire du modèle de commutateur. Dans ces équations, le délai maximum $\overline{D}$ dépend des paramètres de la courbe d'arrivée : la quantité de données maximale σ contenue dans une avalanche et un majorant ρ du taux d'arrivée moyen des données. En conséquence, il est nécessaire de connaître les valeurs de l'enveloppe (σ, ρ) en chaque point du réseau. Comme le montre la figure 4.1, le problème est qu'initialement, les contraintes liées à l'arrivée de données d'un flux ne sont déterminées qu'à l'entrée du réseau. Si les paramètres de l'enveloppe $\left(\sigma^0, \rho^0\right)$ de la courbe d'arrivée initiale sont connus, les paramètres $\left(\sigma^1, \rho^1\right)$ suite à la traversée d'un premier système (équipement réseau) ne le sont pas, et plus généralement nous ne bénéficions d'aucune hypothèse sur la fonction pouvant être adoptée pour

caractériser de nouveau le flux.

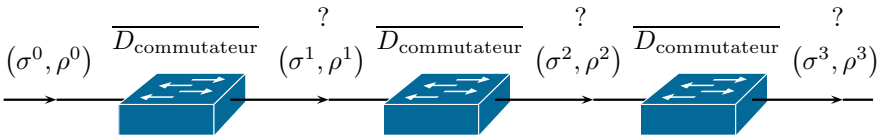


FIG. 4.1: Evolution de la contrainte d'avalanche de données tout au long d'un réseau Ethernet commuté.

Il est donc nécessaire de déterminer la courbe d'arrivée qui sera utilisée par le système suivant (le second commutateur sur la figure 4.1). Cette courbe sera appelée la courbe de départ. La courbe de départ d'un système correspondra donc à la courbe d'arrivée du prochain système traversé par le flux. La différence résidera simplement dans le fait que pour la courbe d'arrivée, il sera admis que l'arrivée des données est également contrainte par la capacité du lien entrant. Pour déterminer cette courbe de départ, nous nous intéressons au troisième théorème fondamental du calcul réseau présenté notamment dans (Le Boudec et Thiran, 2001) et inclus dans l'annexe.

Le flux de sortie d'un système est contraint par une courbe de départ α^* (ou courbe d'arrivée qui contraint le flux de sortie conformément à la définition d'une courbe d'arrivée 49), donnée par :

$$\alpha^*(t) = \alpha(t) \oslash \beta(t) = \sup_{v \geq 0} \{\alpha(t+v) - \beta(v)\} \tag{4.1}$$

L'équation précédente pourrait donc être directement appliquée et permettre ainsi de déterminer l'évolution des caractéristiques des flux. Toutefois, on peut remarquer que l'expression de la courbe de départ s'appuie sur la courbe de service du système. Or l'analyse précédente ne nous a pas conduit à définir la courbe de service de chaque composant élémentaire, et en particulier pour le multiplexeur. Il est par conséquent impossible d'utiliser directement cette propriété. Notons également l'importance de la nature de la fonction correspondant à la courbe de départ. En effet, les différentes expressions du délai de traversée obtenues au chapitre précédent ne sont vraies que pour des courbes d'arrivée affines. Si la courbe de départ obtenue est différente, une étape intermédiaire devra être envisagée que ce soit pour formuler de nouvelles expressions basées sur de telles courbes, soit pour déduire une fonction affine majorant la courbe de départ (dans ce cas, il convient d'utiliser la méthodologie générale d'évaluation par calcul réseau induite à l'équation (3.8)).

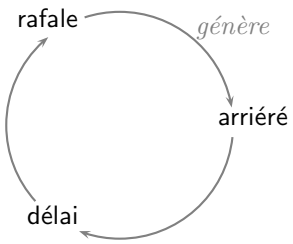
(Cruz, 1991*a*) établit alors une autre expression de la sortie des données d'un flux qui se base sur l'évolution de l'avalanche des données en sortie d'un système, et plus généralement tout au long d'un réseau. Les résultats suivants sont introduits dans le cadre d'un système pour lequel l'arrivée des données est contrainte par $b_{in}(t)$ ($R_{in} \sim b_{in}$) et pour lequel le majorant du délai de traversée $\overline{D}$ est fini ($\overline{D} < +\infty$). Cruz note alors que la sortie des données est contrainte par b_{out} ($R_{out} \sim b_{out}$) où $b_{out}(t)$ est défini pour tout t positif par :

$$b_{out}(t) = b_{in}\left(t + \overline{D}\right) \tag{4.2}$$

L'équation suivante peut alors être détaillée comme suit en utilisant la relation (3.9).

$$\begin{aligned} \sigma_{out} &= \sigma_{in} + \rho_{in}\overline{D} \\ \rho_{out} &= \rho_{in} \end{aligned} \tag{4.3}$$

Cruz motive l'équation (4.2) par le constat que si $\overline{D}$ est faible, alors le flux de sortie sera fortement semblable au flux d'entrée. Par rapport au précédent majorant de la courbe de départ, cette expression est conforme au résultat de l'opération $\alpha(t) \oslash \beta(t)$ lorsque $\beta(t) = \delta_{\overline{D}} = \begin{cases} 0 & \text{si } t \leq \overline{D} \\ \infty & \text{sinon} \end{cases}$ ($\delta_{\overline{D}}$ s'apparente à une impulsion de Dirac appliquée à l'instant $\overline{D}$), c'est-à-dire que le service correspond à un retard fixe. L'interprétation de cette équation est double : le taux d'arrivée moyen des données reste (évidemment) le même et le délai est transformé en avalanche de données supplémentaires. Le raisonnement général peut alors se représenter comme suit :



Un flux arrive sur un système avec un certain niveau de rafale. L'avalanche de données introduite par ce flux participe alors à l'augmentation de l'arriéré de traitement du système. Cette quantité de travail en attente donne lieu au délai de traversée du système. L'attente donne lieu à une augmentation de la quantité de données appartenant au flux, qui pourra dans le pire des cas être finalement retransmise en continu. D'où l'apparition en sortie d'une avalanche de données plus importante qu'en entrée, cette augmentation étant proportionnelle au délai. Ce phénomène se répétera par la suite de système en système.

L'application de cette propriété à la figure 4.1 implique ainsi que la courbe de départ du premier commutateur sera définie par $(\sigma^1, \rho^1) = (\sigma^0 + \rho^0 \overline{D_{commutateur}}, \rho^0)$. Des délais de traversée de plusieurs systèmes successifs peuvent ainsi être exprimés en réitérant plusieurs fois cette équation. Nous constatons alors que le majorant du délai de traversée du commutateur correspond simplement à la somme des majorants des délais de traversée des composants élémentaires constitutifs du modèle.

$$\begin{aligned} \sigma^*_{\text{commutateur}} &= \sigma^*_{\text{port de sortie}} \\ &= \underbrace{\left(\underbrace{(\sigma + \rho\overline{D_{\text{mux}}})}_{\sigma^*_{\text{mux}}} + \rho\overline{D_{\text{mémoire}}} \right)}_{\sigma^*_{\text{mémoire}}} + \rho\overline{D_{\text{port de sortie}}} \\ &= \sigma + \rho\underbrace{(\overline{D_{\text{mux}}} + \overline{D_{\text{mémoire}}} + \overline{D_{\text{port de sortie}}})}_{\overline{D_{\text{commutateur}}}} \end{aligned} \tag{4.4}$$

Ainsi il est possible de déduire le délai de traversée d'un système de bout en bout à partir de l'augmentation totale de l'avalanche maximale du flux.

Cette technique est également utilisée dans (Grieu, 2004) pour calculer la courbe de départ suite à la traversée d'un multiplexeur. Dans ce cas, des composants comme le multiplexeur sont vus comme de simples retardateurs.

4.1.2 Inter-dépendances des avalanches de données de bout en bout

Dans le chapitre précédent, l'importance de connaître l'état des avalanches de données a été établie pour calculer les délais de traversée des différents composants du réseau. La relation liant l'entrée et la sortie (équation (4.2)) permet d'exprimer la valeur de ces avalanches en sortie d'un système. L'un des points essentiels de cette relation est qu'elle lie le niveau de l'avalanche en sortie au majorant du délai de traversée, qui dépend lui même de la valeur des avalanches des flux en entrée. Considérons alors la figure 4.2.

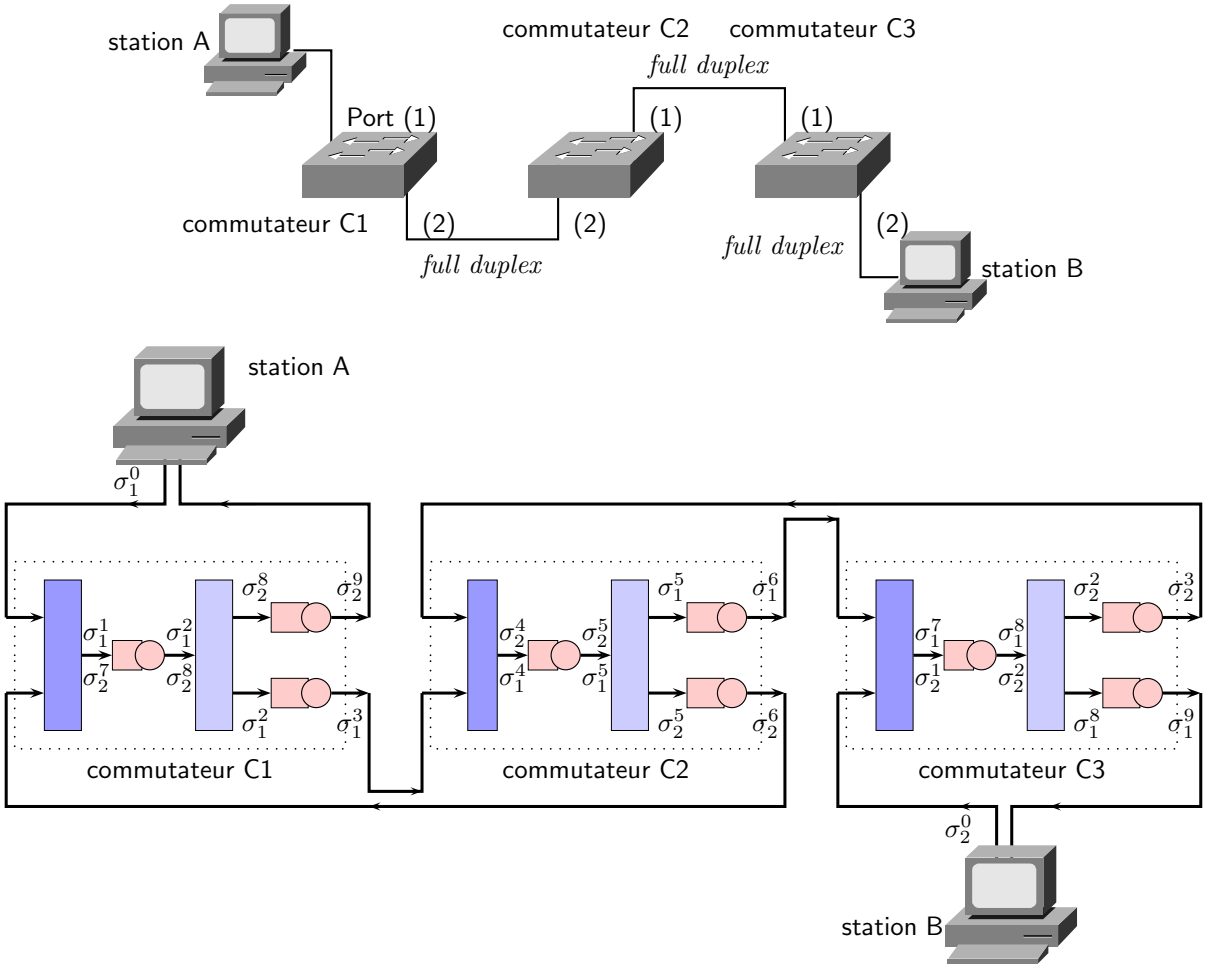


FIG. 4.2: Confluences croisées des flux

Dans le cadre simple d'une communication de la station A vers la station B, cette relation peut être directement appliquée et le délai de traversée de bout en bout peut être directement obtenu en répétant cette méthode. Toutefois, si l'on considère maintenant que la station B génère également un échange de données vers la station A, l'utilisation directe de cette relation n'est plus possible. L'expression du délai de traversée du multiplexeur du commutateur C2 dépend ainsi de deux avalanches σ_1^3 et σ_2^3 qui sont initialement inconnues. Pour déterminer ces inconnues, nous remontons au commutateur précédent. On note alors que σ_1^1 dépend de σ_1^0 (connu) et de σ_2^7 qui est fonction de σ_1^4, et par conséquent σ_1^1. Ce

blocage montre l'inter-dépendance des avalanches de données. Cette dépendance est issue des différents points de rendez-vous ou confluences des flux. Les réseaux ne faisant pas intervenir ce type de confluences croisées sont appelés *feed-forward*. De manière générale, les réseaux Ethernet commutés ne possèdent pas de qualité d'évitement de cette situation. Quand bien même certaines topologies et scenarii sur Ethernet commuté ne font pas apparaître cette caractéristique, il nous faut considérer une méthode générale dans laquelle cette situation soit prise en compte.

La résolution du problème nécessite alors de regrouper les différentes expressions des avalanches de données sous la forme d'une équation matricielle.

4.1.3 Calcul des délais de bout en bout

Dans ce nouveau paragraphe, nous présentons une partie importante de nos contributions à l'adaptation du calcul réseau pour l'étude des systèmes contrôlés en réseau. Cette contribution a notamment été présentée dans (Georges *et al.*, 2003*a*),(Georges *et al.*, 2002).
Par convention, la notation σ_i^j est utilisée avec i l'identifiant du flux et j le nombre de commutateurs déjà traversés par les données de ce sous-flux. Au départ, le volume maximum des rafales d'un flux i est donc représenté par σ_i^0.

Une caractéristique particulière des réseaux Ethernet commutés est l'utilisation du protocole Spanning Tree défini dans le standard (IEEE, 1998). L'objectif de ce protocole est de détecter d'éventuelles boucles et de n'autoriser qu'un chemin entre deux entités interconnectées par un réseau Ethernet commuté de façon à former une topologie active en arbre. Cela signifie qu'en régime permanent les routes empruntées par les trames sont fixes et connues. A partir de là, notre méthode de calcul des délais de bout en bout s'appuie sur le constat que le chemin entre deux nœuds terminaux est unique, statique et peut être connu *a priori*.

Méthode de calcul

Les étapes de la méthode sont :

1. Pour chaque entité communicante, identifier chaque flux et déterminer les paramètres initiaux de la contrainte d'avalanche des données.
2. Identifier les chemins de chacun de ces flux.
3. Pour chaque commutateur du réseau, déterminer le lien le plus chargé.
   Dans cette proposition, il est seulement suggéré de prendre les valeurs initiales des flux. Cette supposition est valable puisque :
   $$\frac{\sigma_i}{(C_{out} - \rho_i)} \leq \max_{\forall j \neq i} \left(\frac{\sigma_j}{C_{out} - \rho_j} - \frac{L}{C_j} \right) \leq \frac{\sigma_k}{C_{out} - \rho_k}$$
4. Sur chaque commutateur, formuler les équations des volumes maxima des rafales en sortie des flux. Il est suggéré d'adopter la notation précédente.
5. Définir le système d'équation sous la forme mathématique
   $$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & b_{n2} & \dots & a_{nn} \end{bmatrix} * \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$
6. Calculer les valeurs des volumes des rafales.
7. Déterminer les délais maxima de bout en bout à l'aide de la formule :
   $$\overline{D}_i = \frac{\sigma_i^h - \sigma_i^0}{\rho_i}$$
   où h représente le nombre de commutateurs traversés moins un.

La première étape de cette méthode s'appuie sur l'hypothèse de départ : même si le trafic entrant est inconnu, il est supposé que la quantité de données transmise au réseau à un instant t est majorée par l'enveloppe (σ, ρ). D'où la nécessité de déterminer les valeurs de σ et ρ. La seconde étape s'appuie sur le protocole Spanning Tree pour identifier les chemins sur l'architecture modélisée comme sur la figure 4.2. Une fois les conditions initiales satisfaites, la méthode s'attache à la formulation des niveaux d'avalanche de données de chacun des flux en tout point du réseau. En fait, ces expressions sont obtenues à partir des délais de traversée de chacun des composants élémentaires du modèle de commutateur identifié au chapitre précédent. Ces équations sont regroupées dans une matrice (de dépendances) des avalanches de données des flux. Un point critique de la résolution de cette équation matricielle est l'inversion de cette matrice ... qui devra être démontrée pour chaque étude.
L'équation matricielle de l'étape 5 représente bien la démarche que nous développons au travers des différentes étapes de l'analyse d'un réseau donnée. Cette équation illustre l'inter-dépendance de l'évolution des avalanches de données de plusieurs flux. Les éléments des matrices A et B sont liés *via* la relation $\sigma^* = \sigma + \rho\overline{D}$ aux expressions du majorant du délai de traversée comme le présente l'analyse de la première plateforme expérimentale de la figure 4.2.

L'obtention des valeurs d'avalanches de données permet alors de déterminer un majorant du délai entre deux points du réseau. Le dernier point de la méthode utilise ainsi l'équation (4.3) pour calculer un majorant du délai de bout en bout qui est proportionnel à l'augmentation de l'avalanche tout au long de sa traversée du réseau.

A ce stade, une méthode de majoration des délais de bout en bout permet d'estimer la performance du réseau Ethernet commuté. Cette méthode s'appuie sur différents modèles de commutateurs qui se basent sur une politique d'ordonnancement FIFO. Nous proposons alors d'utiliser cette méthode afin de déterminer le modèle le plus pertinent.

4.2 Comparaison des modèles (Georges *et al.*, 2003*c*),(Georges *et al.*, 2003*b*)

Dans le chapitre précédent, trois modèles de commutation ont été proposé (figure 3.8). L'objectif de ce paragraphe est de les évaluer à partir d'un scénario de communication industrielle donné. La référence de comparaison sera définie ici par les résultats d'un outil de simulation réseau : *Comnet III* (figure 4.3).

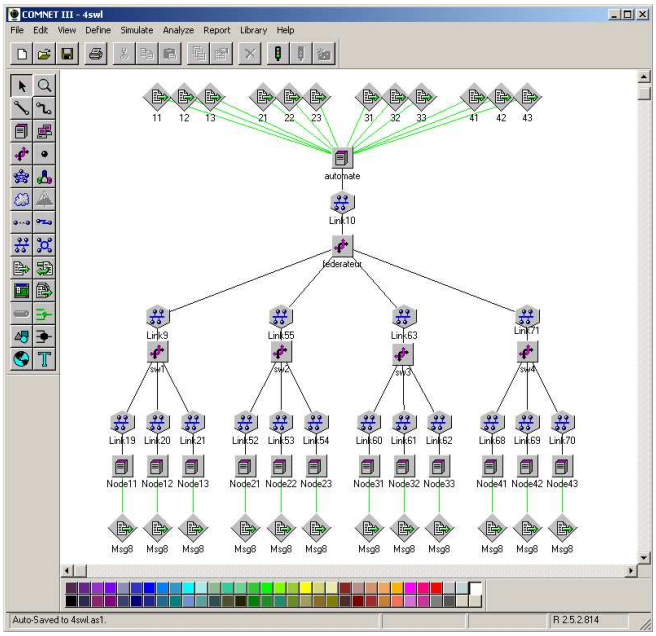


FIG. 4.3: L'outil de simulation *Comnet III*

Dans cet outil, la modélisation d'un équipement réseau tel un commutateur (figure 4.4) est réalisée au moyen de buffers qui représentent les ports d'entrée et de sortie. Une taille et des temps de traitement sont ensuite associés à ces buffers (CACI Products Company, 1998). Un bus interne est également utilisé pour représenter le transfert des trames d'un port à l'autre. Le temps de traitement issu du processeur central est pris dès qu'un paquet quitte son buffer d'entrée afin d'être redirigé *via* le bus.

En résumé, les ports sont modélisés par des buffers, le processeur central par l'accès à un bus unique et la fabrique de commutation par différent bus internes. Les calculs de délai sont alors réalisés au moyen d'une théorie d'attente stochastique.

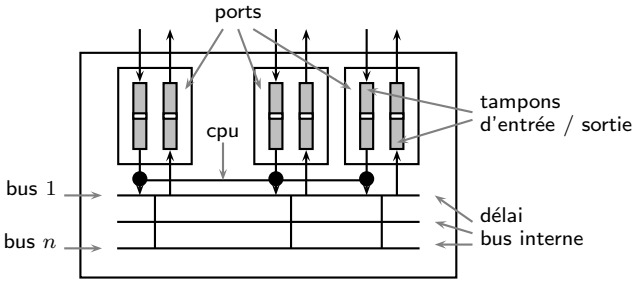


FIG. 4.4: Modélisation d'un commutateur dans *Comnet*

Les outils de simulation réseau sont fréquemment utilisés dans les phases de conception de réseau afin de vérifier les propriétés de l'architecture (Jasperneite et Neumann, 2001). Ici, l'utilisation de cet outil comme référence vise simplement à établir un comparatif entre modèles et résultats de simulation. Cela permet en particulier d'identifier le modèle le plus pertinent. La validation complète du calcul réseau établi dans ces travaux sera quant à elle fondée sur des expérimentations réelles présentées au paragraphe 4.4.

4.2.1 Scénario de communication

L'architecture de communication (figure 4.5) repose sur un commutateur fédérateur sur lequel sont raccordés l'automate et les autres commutateurs de second niveau qui connectent les cartes d'entrées-sorties.

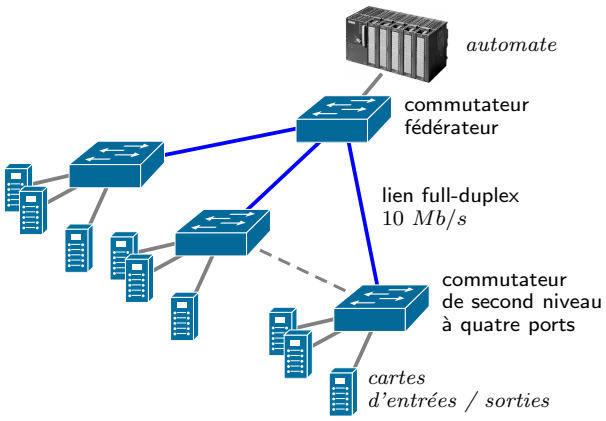


FIG. 4.5: Réseau Ethernet commuté d'étude

Les commutateurs de deuxième niveau ont quatre ports sur lesquels trois cartes d'entrées-sorties peuvent être raccordées. Les liens entre les commutateurs ont des débits identiques de 10 Mb/s et sont configurés en full duplex. Les échanges entre l'automate et les cartes d'entrées-sorties se font de la manière suivante : l'automate envoie périodiquement sur chaque carte déportée un message et les cartes envoient aussi périodiquement un message vers l'automate. Les périodes d'émission de ces messages sont fixes et identiques, et sont égales à 1 ms. La taille d'un message est fixe et est égale à 46 octets. En ajoutant les informations d'encapsulation Ethernet (18 octets), la longueur de la trame est de 72 octets et correspond à la taille minimale d'une trame Ethernet (préambule compris). En considérant que nous travaillons avec

des cartes d'entrées-sorties tout ou rien, chaque message peut alors transporter jusqu'à 368 informations binaires.

Nous allons étudier les délais d'acheminement maxima lorsque l'architecture contient successivement 1, 2, 3 puis 4 commutateurs de second niveau. Davantage de commutateurs produisent sur le lien entre l'automate et le commutateur fédérateur des charges dépassant sa capacité (10 Mb/s).

4.2.2 Résultats

Après simulation, nous obtenons le graphe de la figure 4.6 qui montre des résultats très différents en fonction du type de modèle. Le premier modèle se situe en dessous des valeurs obtenues par simulation, tandis que les deux autres modèles se trouvent au-dessus. Le comportement du premier modèle n'est pas surprenant car comme nous l'avions précédemment indiqué, la notion d'arbitrage centralisée inhérente aux commutateurs Ethernet n'est pas implémentée. Cela a pour effet de paralléliser les transactions à l'intérieur des commutateurs et a pour incidence d'éliminer l'impact de la charge globale du commutateur sur un flux particulier. C'est pourquoi, nous retrouvons des délais d'acheminement maxima très en deçà de la simulation et des deux autres modèles.

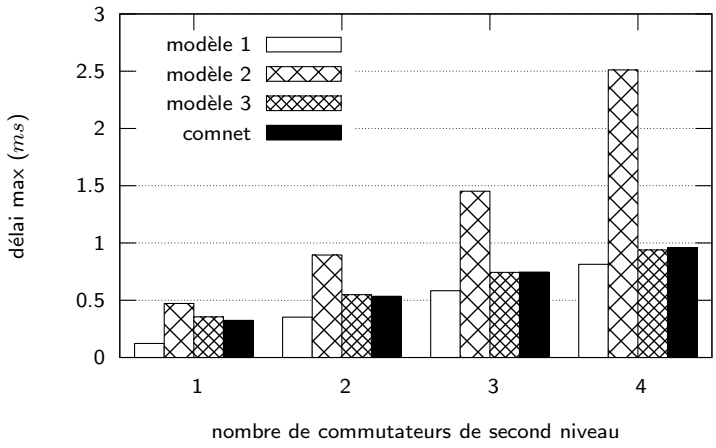


FIG. 4.6: Evaluation des modèles

Le deuxième modèle donne des résultats nettement supérieurs à ceux produits par la simulation. Ce constat est logique dans le sens où le simulateur travaille avec des commutateurs ayant une capacité interne à 1 Gb/s (vitesse du bus) et que cette vitesse n'est pas représentée dans le modèle n°2.
Finalement, le troisième modèle fournit des délais maxima de bout en bout très proches des simulations et ouvre une première piste sur le fait que la surestimation introduite par la théorie du calcul réseau n'est pas très importante lorsque le système à étudier est finement représenté.

4.2.3 Conclusion

Cette étude nous a permis de mettre en avant le modèle n°3 qui vaut pour un certain type d'implémentation de la commutation. La définition de ce modèle a été établie dans le cadre du standard IEEE 802.1D.
L'objectif était simplement ici d'évaluer la pertinence de la *sélection* des composants élémentaires introduite par les différents modèles de commutateur. Cette comparaison vise simplement à montrer que ces

modèles n'intègrent pas l'ensemble des caractéristiques de la commutation (le modèle n°2 ne prend pas en compte le fait que la retransmission des trames au niveau des ports de sortie est limitée par la capacité du lien de sortie) ou alors que ces modèles correspondent à une technique de commutation bien précise. Nous avons ainsi pu vérifier ce que nous avions observé auparavant, à savoir que les deux premiers modèles ne sont pas suffisament précis ou alors qu'ils correspondent à un autre type de commutateur. Plus qu'une comparaison, cette partie prolonge les remarques intuitives établies au chapitre précédent.

Toutefois, ceci ne conduit pas à une validation du modèle n°3. Elle sera réalisée à la fin du chapitre de manière expérimentale.

Dans le paragraphe suivant, nous allons chercher à compléter ce modèle afin d'incorporer les éléments de classification de service introduite au chapitre 2.

4.3 La classification de service

Les travaux présentés dans ce paragraphe ont été publié dans (Georges *et al.*, 2004*c*), (Georges *et al.*, 2004*a*) et (Georges *et al.*, 2005*b*).

4.3.1 Etude de la standardisation

La gestion des priorités sur Ethernet apparaît avec le concept des VLANs et le standard IEEE 802.1Q (IEEE Computer Society, 2003). Afin de bien en comprendre les termes, considérons tout d'abord la figure 4.7.

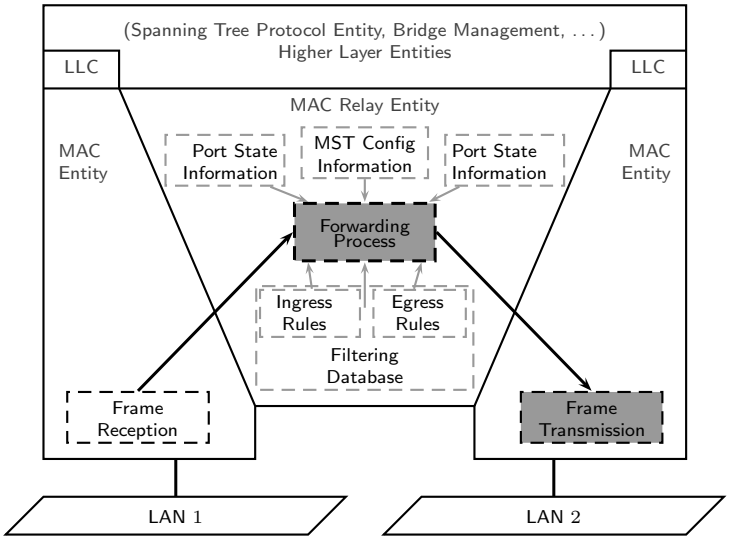


FIG. 4.7: Relais d'une trame par un pont *802.1D/p* (IEEE Computer Society, 2003, figure 8-4).

L'ordonnancement intervient alors juste avant la *transmission*, comme l'art de sélectionner les trames pour émission immédiate. Le paragraphe de cette norme (page 49) précise alors son mode de fonctionnement.

> *"**8.6.6 Transmission selection***
>
> *The following algorithm shall be supported by all Bridges as the default algorithm for selecting frames for transmission :*
>
> *a) For each Port, frames are selected for transmission on the basis of the traffic classes that the Port supports. For a given supported value of traffic class, frames are selected from the corresponding queue for transmission only if all queues corresponding to numerically higher values of traffic class supported by the Port are empty at the time of selection ;*
>
> *b) For a given queue, the order in which frames are selected for transmission shall maintain the ordering requirement specified in 8.6.5.*
>
> *Additional algorithms, selectable by management means, may be supported as an implementation option so long as the requirement of 8.6.5 are met."*

Le premier point (a) nous apprend ainsi que la discipline de service définie par défaut est la politique à priorité stricte. Le second point (b) complète cette définition en précisant la politique d'ordonnancement qui s'applique à l'intérieur d'une file donnée. Le standard (IEEE Computer Society, 2003, page 47) précise ainsi :

> *"**8.6.5 Queuing for transmission***
>
> *The Forwarding Process provides storage for queued frames, awaiting an opportunity to submit these for transmission to the individual MAC Entries associated with each Bridge Port. The order of frames received on the same Bridge Port shall be preserved for :*
>
> *a) Unicast frames with a given user_priority [...] for a given combination of destination_address and source_address ;*
>
> *b) Group-addressed frames with a given user_priority [...] for a given destination_address."*

Ce paragraphe précise en fin de compte que la discipline de service d'une seule file doit rester FIFO.

En fait, le standard n'oblige l'algorithme de sélection des trames pour émission sur le port de sortie qu'au respect d'un ensemble de règles conduisant à l'écartement de trames. Ceci semble donc laisser la porte ouverte à un ensemble relativement vaste d'algorithmes. La seule contrainte étant qu'en modifiant l'algorithme par défaut (priorité stricte), on réduise le service offert par le commutateur d'un point de vue global, et que l'on soit ainsi amené à écarter davantage de trames.

Les priorités vont intervenir lors de la gestion de la congestion d'un commutateur, gestion qui est réalisée à travers l'implémentation de buffers. Pour les flux les plus importants (plus sensibles aux délais), un traitement préférentiel sera accordé en fonction du niveau de priorité. Néanmoins, la prioritisation ne peut pas pallier à une insuffisance de capacité de traitement d'un commutateur.

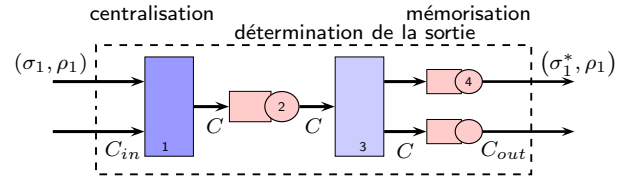
Deux méthodes de prioritisation pourront être considérées :

- d'accès, dans ce cas le niveau de priorité est fixé au niveau de l'accès au réseau pour un équipement ;
- d'utilisateur, la priorité est définie par l'application pour un ensemble de trames ou de façon spécifique, pour une trame.

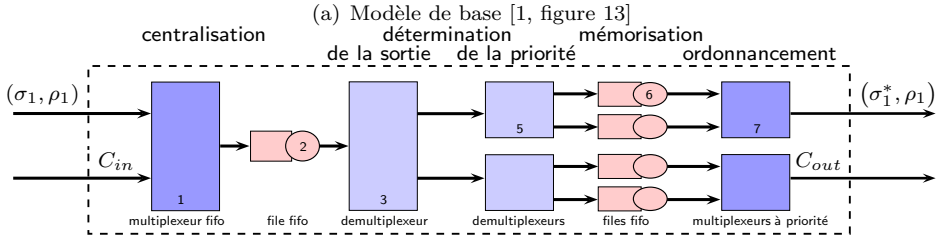
Dans ces deux cas, un mécanisme de classification global devra être au préalable défini. Conformément à l'étude menée au chapitre 2, l'étude suivante sera consacrée aux politiques à priorité stricte (SP) et WFQ (weighted fair queueing).

4.3.2 Modélisation du commutateur à priorité

Comme nous l'avons vu au chapitre précédent, le point dur de l'évaluation des performances d'un réseau commuté est la caractérisation du service offert par un commutateur. Dans un premier temps, la modélisation a concerné un commutateur *802.1D* classique (figure 4.8(a)). Nous nous intéressons alors aux modifications du modèle $n°3$ induites par la prise en compte de la CdS normalisée par *802.1p*.



(a) Modèle de base [1, figure 13]



(b) Modèle de commutation à priorité

FIG. 4.8: Le mécanisme de prioritisation modifie la mise en attente au niveau des ports de sortie

La prise en compte des priorités consiste essentiellement en l'ajout sur chaque port de sortie d'autant de buffers que de priorités. Ces buffers sont modélisés par des files d'attente (figure 4.8(b), composant 6). La commutation sur ces files est réalisée par un démultiplexeur (composant 5) relativement aux priorités des trames. Cette opération est définie par le mapping entre priorité et files introduit dans le standard. Finalement, le dernier multiplexeur (composant 7) représentant un port de sortie gère la politique d'ordonnancement entre les différentes files. La politique de retransmission utilisée par ces multiplexeurs de sortie sera donc fixée par l'algorithme d'ordonnancement à priorité stricte ou Weighted Fair Queueing. Les files d'attentes en sortie correspondent donc simplement à des files logiques dans lesquelles les trames attendront que l'ordonnanceur de sortie les sélectionne pour retransmission. La figure 4.8(b) représente le modèle d'un commutateur deux ports gérant deux niveaux de priorité.

L'étape suivante consiste alors à exprimer des délais de traversée de chacun des composants élémentaires introduits par le modèle de la figure 4.8(b). Nous nous intéressons simplement au port de sortie, les expressions définies au chapitre précédent pour les composants en amont restant vraies. Dans le chapitre précédent, la pauvreté d'Ethernet nous a conduit à formuler le majorant du délai à partir de l'arriéré de traitement. Dans le cas présent, même si la classification de service n'introduit aucune réservation formelle de ressources, elle détaille les informations concernant le service offert à un flux au vu de sa priorité. Ainsi, il devient possible de déterminer la courbe de service minimale offerte par le port de sortie.

Dans la suite, nous nous intéresserons à un nœud à trois entrées. La capacité des ports d'entrée sera définie par C_{in} b/s et la capacité de la sortie par C b/s, tel que $C_{in} \geq C$. Chaque flux est contraint par une enveloppe $(\sigma_i, \rho_i, C_{in})$ avec $\sum_j \rho_j < C$. De plus, un poids ϕ_i est attribué à chaque flux. Nous supposons également que $\phi_i > \phi_{i+1}$. Concernant la courbe de sortie d'un flux, nous pouvons déjà écrire

que $R_i^*(t) \leq Ct$. Les paragraphes suivants présentent alors l'analyse des services offerts par ce nœud selon que la politique d'ordonnancement est à priorité stricte ou WFQ (weighted fair queueing).

4.3.3 Ordonnancement à priorité stricte (SP)

Dans la priorité stricte, aucune garantie explicite de réservation de ressources n'est spécifiée pour un flux de données. L'ordre de sélection des trames dépend simplement de l'ordre des priorités (ou poids). C'est pourquoi l'étude du service offert par le multiplexeur sera distincte selon le flux considéré. Les calculs suivants sont illustrés sur la figure 4.9.

Priorité haute

La politique à priorité stricte garantit aux trames du flux identifié par la priorité la plus haute (nous considérerons qu'il s'agit du flux 1) d'être sélectionnées en priorité. Puisque la retransmission d'une trame ne peut être préemptée sur le réseau, les trames du flux 1 pourront être mises en attente le temps de finir la retransmission d'une trame de priorité inférieure. Par conséquent, la courbe de service associée au flux 1 de priorité haute est liée à la longueur maximale des trames de priorité inférieure. Elle est donnée par :

$$\begin{aligned} \beta_1(t) =& R(t-T)^+ \\ & T = \max\{L_{2,max}, L_{3,max}\}/C,\ R = C \end{aligned} \tag{4.5}$$

Priorité intermédiaire

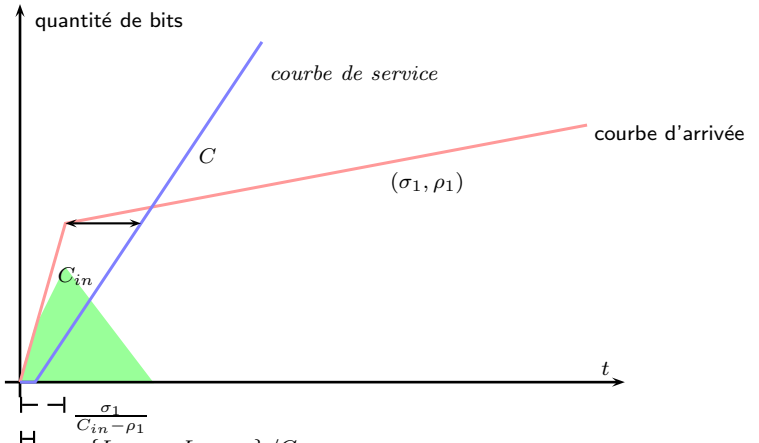
Les trames du flux 2 sont traitées avant les trames (de priorité inférieure) du flux 3. Néanmoins, ces trames doivent attendre que plus aucune trame (de priorité supérieure) du flux 1 ne soit également en attente de retransmission. Une première partie de la latence correspond donc au temps de traitement de l'avalanche de données du flux 1. De plus, il est impossible de stopper la retransmission d'une trame du flux 3 dans le but de servir une trame du flux 2 qui serait juste arrivée. Finalement, puisque le nœud servira d'abord les trames du flux 1, le taux de sortie des trames de priorité intermédiaire est limité par $C - \rho_1$. Par conséquent :

$$\begin{aligned} \beta_2(t) =& R(t-T)^+ \\ & T = \frac{\sigma_1}{C-\rho_1} + \frac{L_{3,max}}{C},\ R = C - \rho_1 \end{aligned} \tag{4.6}$$

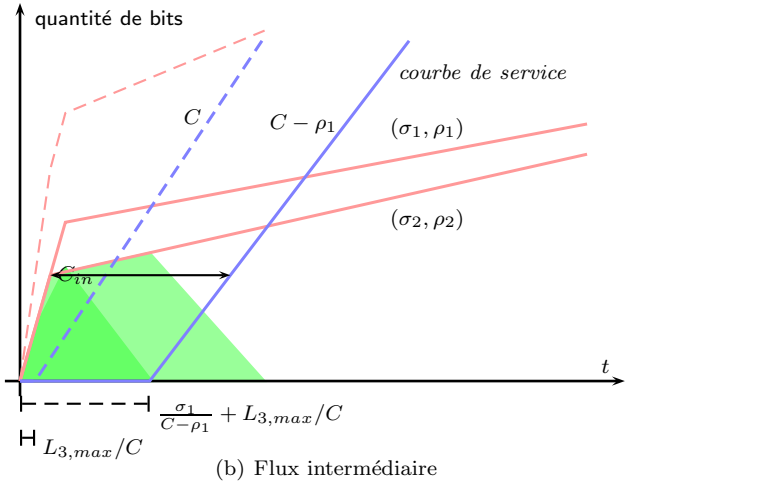
Priorité basse

Une trame du flux 3 sera traitée sous la condition qu'aucune trame des flux 1 et 2 soit encore en attente. Cela signifie que la latence est définie par le temps de traitement de l'avalanche de données correspondant à l'union des flux prioritaires. De plus, le taux de service offert au flux 3 sera limité dans ce cas par $C - \rho_1 - \rho 2$. D'où :

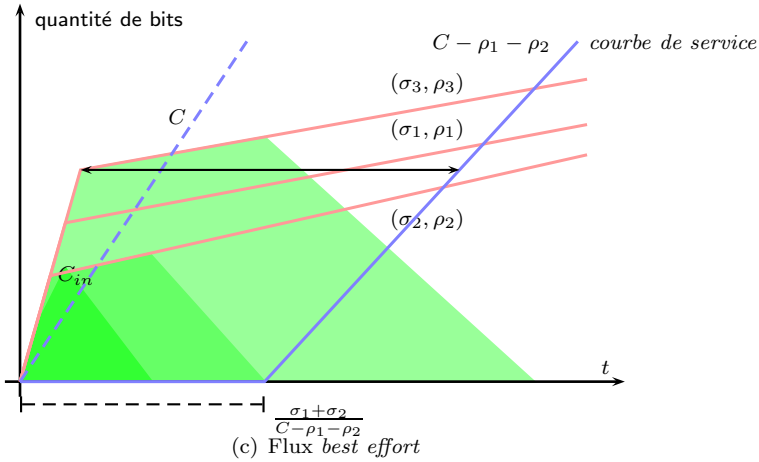
$$\begin{aligned} \beta_3(t) =& R(t-T)^+ \\ & T = \frac{\sigma_1 + \sigma_2}{C-\rho_1-\rho_2},\ R = C - \rho_1 - \rho_2 \end{aligned} \tag{4.7}$$



(a) Flux prioritaire



(b) Flux intermédiaire



(c) Flux *best effort*

FIG. 4.9: Courbe de service & arriéré de traitement pour un ordonnancement statique à priorité stricte.

Les équations (4.5), (4.6) et (4.7) montrent qu'avec la politique à priorité stricte, le service offert à un flux dépend des autres flux. De surcroît, l'équation (4.7) démontre que le service offert au flux de priorité basse peut tendre vers zéro (c'est l'effet "famine" engendré par cet algorithme d'ordonnancement).

4.3.4 Ordonnancement WFQ

Algorithmes équitables

Le Weighted Fair Queueing, initialement proposé dans (Demers *et al.*, 1989), est également connu sous la dénomination Packetized Generalized Processor Sharing (PGPS) (Parekh, 1992). Il est basé sur l'algorithme conceptuel appelé Generalized Processor Sharing (GPS) (Parekh et Gallager, 1993). Un nœud GPS est caractérisé par n réels positifs $\phi_1, \phi_2 \dots \phi_n$. Il opère à un taux fixe C et est non oisif. Chaque flux i possède un taux de service garanti noté c tel que :

$$c = \frac{\phi_i}{\sum_{j=1}^{n} \phi_j} C$$

La politique GPS est intéressante puisqu'elle est équitable, flexible (le nombre ϕ_i permet de modifier le service offert à un flux donné, et par conséquent aux autres flux) et elle présente des propriétés d'analyse et de majoration. Par exemple, dans le cas où $n = 2$ et sous l'hypothèse que R_2 est majoré par (σ, ρ), (Chang, 2000) montre que la courbe de service offerte au flux 1 est définie par :

$$\beta_1(t) = \max\left\{(C-\rho)\,t - \sigma, \frac{\phi_1}{\phi_1+\phi_2} Ct\right\}$$

Toutefois, l'algorithme GPS est une politique d'ordonnancement idéaliste qui suppose que les paquets sont infinitivement divisibles. Aussi, (Parekh et Gallager, 1993) présentent un schéma de transmission paquet par paquet qui approxime GPS : PGPS. Lorsque il est prêt, un nœud PGPS sélectionne le paquet qui serait le premier retransmis dans la simulation GPS comme le montre la figure 4.10

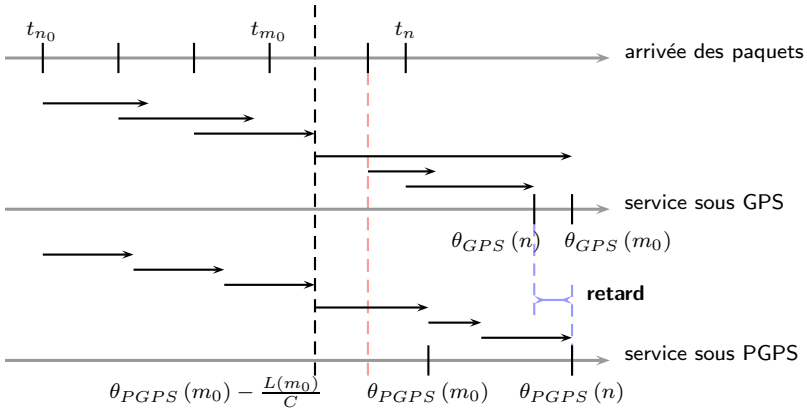


FIG. 4.10: Contexte de retardement du service offert à un paquet sous PGPS par rapport à GPS.

La figure 4.10 montre que lors de la sélection du paquet à retransmettre à un instant t, PGPS n'étudie que les paquets arrivés à cet instant ce qui peut conduire à une différence dans l'ordre de fin de traitement des paquets.
L'implémentation pratique majeure de PGPS est le temps virtuel (virtual time, voir (Parekh et Gallager, 1993)) qui se comporte comme un serveur événementiel, un événement étant une arrivée ou une sortie sous GPS. Le codage de l'algorithme réagit à deux types d'événements.

- *à chaque arrivée d'un paquet, le paquet est étiqueté avec sa date de fin de retransmission.*
- *à chaque fin de retransmission d'un paquet, l'ordonnanceur choisit le paquet présentant l'étiquette temporelle la plus petite parmi l'ensemble des paquets disponibles.*

Comme le montre la figure 4.10, (Parekh et Gallager, 1993) notent alors que les délais peuvent être plus long sous PGPS qu'avec GPS. Néanmoins, la politique PGPS présente les mêmes avantages que GPS, c'est-à-dire équité, flexibilité et propriétés intéressantes d'analyse. Avec la notation $(x)^+$ pour $\max\{x, 0\}$, (Chang, 2000) a ainsi établi que le service offert à un flux i est défini par :

$$\beta_i(t) = \left(\frac{\phi_i}{\sum_{j=1}^{n} \phi_j} (t - L_{max}) - L_{i,max} \right)^+$$

L'utilisation de PGPS dans un commutateur pose alors le problème de la complexité de l'algorithme en fonction de n, le nombre de priorités. Comme le montrent (Bennett et Zhang, 1997), le problème de PGPS est que la complexité des opérations d'insertion ou de suppression de trames dans la file est $O(\log n)$ et que la pire complexité de calcul du temps virtuel $V(t)$ peut aller à $O(n)$. De ce fait, les implémentations pratiques de WFQ dans les commutateurs actuels sont basées sur une politique Weighted Round Robin qui reste beaucoup plus simple.

Avec la politique round robin, les trames sont poussées dans les files selon leur niveau de priorité. Le nœud traite alors ces différentes files selon une séquence cyclique (en utilisant un ordre prédéfini qui dépend de la priorité des files) dans le but de servir les trames de chaque file non vide. Même si cet algorithme respecte l'équité, il n'intègre pas de mécanisme de flexibilité. De plus, l'équité peut être remise en cause lorsque la taille des trames est variable. Pour améliorer la flexibilité d'une politique à round robin simple, le Weighted Round Robin (WRR) associe un poids w_i à chaque flux i. Le nœud WRR cherchera alors à servir un flux i avec un taux $\frac{w_i}{\sum_j w_j}$ avant de passer à la file suivante. Si l'on compare avec la politique PGPS, il est évident que les délais pourront être plus long puisque si le système est fortement chargé et qu'une trame vient juste de manquer son service, elle devra attendre le prochain tour de sa file, c'est-à-dire le prochain cycle.

Notre étude se portera alors sur une politique WFQ **basée sur une bufferisation par priorité et un ordonnancement weighted round robin**. Cette implémentation est typique de commutateurs du marché, comme le *Cisco Catalyst 2950*. Le paragraphe suivant est consacré à la formulation du service offert par ce type de politique.

Bufferisation équitable et *Weighted round robin*

Le principe d'équité impose que le service offert à un flux ne dépende pas des paramètres (σ, ρ) des autres flux. Si l'on souhaite améliorer le service offert à un trafic à temps-critique, les poids (noté ϕ_i pour un flux de priorité i) pourront être modifiés pour limiter le service offert aux trames de priorité inférieure.

Dans un schéma round robin, le nœud cherche à servir jusqu'à w_i paquets pour chaque file non vide avant de passer à la file suivante. Etant donné que cette solution n'est pas robuste à des trames de longueur variables, il sera supposé que pas plus de ϕ_i unités de données d'un flux soient traitées à chaque fois que ce trafic sera servi. La longueur maximale d'un cycle est par conséquent limitée par $\frac{\sum_j \phi_j}{C}$ et le temps de traitement offert à un flux i par cycle est majoré par $\frac{\phi_i}{C}$. De plus, le principe d'équité impose que le nœud ne fournisse pour un flux donné pas plus de ϕ_i unités de données. Comme la transmission

d'une trame sur le réseau ne peut pas être préemptée, le nœud pourra refuser de sélectionner une trame si sa longueur est supérieure à la quantité d'unités restantes. Si bien que si la file d'attente d'un flux donné ne devient jamais vide, la quantité de données offerte à un flux durant un cycle sera limitée dans le pire cas par $\phi_i - L_{i,max}$. Il est alors important de noter que contrairement à la priorité stricte, le service offert à un flux dépend simplement des poids des flux et non de leurs différentes caractéristiques comme la longueur de l'avalanche de données. Cette politique évite ainsi le phénomène de famine.

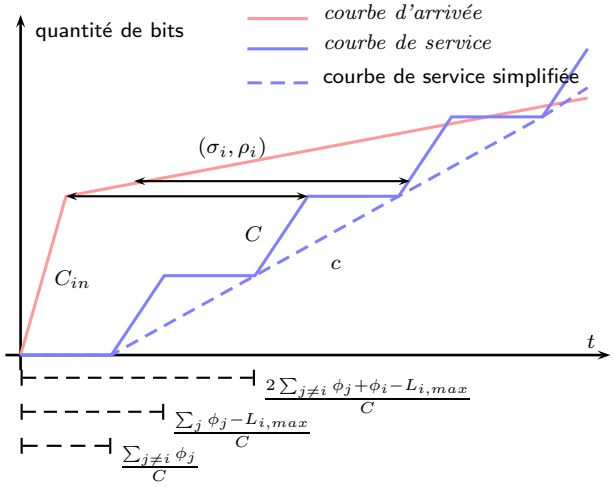


FIG. 4.11: Courbe de service *weighted round robin*.

La définition de la courbe de service minimale offerte à un flux va donc utiliser les propriétés mises en avant ci-dessus. Il est également nécessaire de noter que le pire cas correspond à l'instant où une trame vient juste de manquer la période consacrée à son niveau de priorité si bien qu'elle doit attendre la prochaine période. Dans le pire cas, il sera supposé que les autres files sont toujours non vides et que par conséquent la trame devra attendre $\frac{\sum_{j\neq i}\phi_j}{C}$. La figure 4.11 illustre la courbe de service qui en est déduite et l'équation (4.8) formule le service offert à un flux.

$$\beta(t) = C\left(t - \left\lceil \frac{t}{\frac{\sum_j \phi_j - L_{i,max}}{C}} \right\rceil \frac{\sum_{j\neq i}\phi_j}{C}\right)^+ \tag{4.8}$$

Comme l'équation (4.8) peut engendrer une complexité de calcul, notons qu'une autre courbe de service peut également être proposée.

$$\begin{aligned} \beta_{R,T}(t) =& R(t-T)^+ \\ R = C\frac{\phi_i - L_{i,max}}{\sum_j \phi_j - L_{i,max}},& \ T = \frac{\sum_{j\neq i}\phi_j}{C} \end{aligned} \tag{4.9}$$

La courbe de service proposée à l'équation (4.9) correspond à la fonction de type *rate latency* la meilleure possible, c'est-à-dire qui maximise le service minimal offert sans pour autant améliorer celui définit par la courbe de service définie par l'équation (4.8). L'intérêt dégagé par ce type de courbe de service est sa simplicité par rapport au pseudo-escalier introduit dans l'équation (4.8).
Ces deux équations (4.8) et (4.9) montrent que la courbe de service offerte dans l'ordonnancement weighted round robin pour un flux donné dépend simplement des poids, d'où une certaine équité et flexibilité. Par simplicité, seule l'équation (4.9) sera considérée dans la suite.

La définition de la courbe de service permet alors de déduire le délai de traversée en utilisant le concept du calcul réseau qui lie le délai aux courbes d'arrivée et de service.

4.3.5 Majorant des délais pour SP et WFQ

Comme nous l'avons vu au chapitre précédent, la figure 4.11 illustre que le délai de traversée correspond à la distance horizontale entre la courbe d'arrivée et de service et est défini par :

$$d_i(t) = \inf_{\Delta \geq 0} \left\{ \min\{C_{in}t, \sigma_i + \rho_i t\} = R(t + \Delta - T)^+ \right\}$$

Le délai est majoré par $\overline{D_i} = \max_{t \geq 0} d_i(t)$.

Nous notons alors $\tau_i = \frac{\sigma_i}{C_{in} - \rho_i}$ la durée de l'avalanche de données du flux i. En utilisant l'opérateur maximum $\vee$ ($a \vee b = \max(a, b)$), $\overline{D_i}$ peut être décomposé comme suit :

$$\begin{aligned}
\overline{D_i} =& d(0) \vee \max_{0<t<\tau_i} d(t) \vee \max_{t \geq \tau_i} d(t) \\
=& T \vee \max_{0<t<\tau_i} \left\{ \inf \left\{ \Delta \geq 0 : C_{in}t = R(t + \Delta - T)^+ \right\} \right\} \\
& \vee \max_{t \geq \tau_i} \left\{ \inf \left\{ \Delta \geq 0 : \sigma_i + \rho_i t = R(t + \Delta - T)^+ \right\} \right\}
\end{aligned}$$

Comme pour tout $t > 0$, $\min(C_{in}t, \sigma_i + \rho_i t) > 0$, $R(t + \Delta - T)^+$ doit être supérieur à 0, si bien que $R(t + \Delta - T)^+ = R(t + \Delta - T)$

$$\begin{aligned}
\overline{D_i} =& T \vee \max_{0 \leq t<\tau_i} \{\inf\{\Delta \geq 0 : R\Delta = (C_{in} - R)t + RT\}\} \\
& \vee \max_{t \geq \tau_i} \{\inf\{\Delta \geq 0 : R\Delta = (\rho_i - R)t + \sigma_i + RT\}\} \\
=& T \vee \max_{0 \leq t<\tau_i} \frac{(C_{in} - R)t + RT}{R} \vee \max_{t \geq \tau_i} \frac{(\rho_i - R)t + \sigma_i + RT}{R}
\end{aligned}$$

Supposons maintenant que la courbe de service est inférieure à la capacité d'arrivée des données ($R \leq C_{in}$) et que le système n'est pas saturé ($\sum_j \rho_j \leq R$). Nous avons :

$$\overline{D_i} = T \vee \frac{(C_{in} - R)\tau_i + RT}{R} \vee \frac{(\rho_i - R)\tau_i + \sigma_i + RT}{R}$$

De plus, comme τ_i est défini tel que $C_{in}\tau_i = \sigma_i + \rho_i\tau_i$, on obtient finalement :

$$\begin{aligned}
\overline{D_i} =& T \vee \frac{(C_{in} - R)\tau_i + RT}{R} \vee \frac{(C_{in} - R)\tau_i + RT}{R} \\
=& T \vee \frac{(C_{in} - R)\tau_i + RT}{R} \\
=& T \vee \left\{ (T - \tau_i) + \frac{\sigma_i + \rho_i\tau_i}{R} \right\}
\end{aligned}$$

Comme $\tau_i = \frac{\sigma_i}{C_{in} - \rho_i}$ et $R \leq C_{in}$, nous avons :

$$\frac{\sigma_i + \rho_i\tau_i}{R} = \frac{\sigma_i/\tau_i + \rho_i}{R}\tau_i = \frac{C_{in}}{R}\tau_i \geq \tau_i$$

En conséquence, le délai est majoré par :

$$\overline{D_i} = (T - \tau_i) + \frac{\sigma_i + \rho_i\tau_i}{R} \tag{4.10}$$

4.3.6 Conclusion

L'équation (4.10) nous permet alors de comparer les délais de traversée obtenus dans le cadre d'une politique d'ordonnancement à priorité stricte et à weighted round robin. Pour souci de clarté de l'illustration, l'étude suivante s'applique à un multiplexeur implémentant trois niveaux de priorité (et donc trois ports d'entrée). La notion de flux correspond ici à l'ensemble des données arrivant sur le multiplexeur *via* le même port d'entrée, c'est-à-dire, l'ensemble des données d'un même niveau de priorité.

Tout d'abord, si l'on considère le flux 1, nous avons :

$$\overline{D_{1,PQ}} = \left(\frac{\max\left(L_{2,max} + L_{3,max}\right)}{C} - \tau_1\right) + \frac{\sigma_1 + \rho_1\tau_1}{C}$$

$$\overline{D_{1,WRR}} = \left(\frac{\phi_2 + \phi_3}{C} - \tau_1\right) + \frac{\sigma_1 + \rho_1\tau_1}{C\frac{\phi_1 - L_{1,max}}{\phi_1+\phi_2+\phi_3-L_{1,max}}}$$

Il est alors intéressant de remarquer ici que si $\phi_2 \to 0$ et $\phi_3 \to 0$, le weighted round robin sera proche de la politique à priorité stricte.
Si l'on considère alors le flux 3, l'expression des délais est alors la suivante :

$$\overline{D_{3,PQ}} = \left(\frac{\sigma_1 + \sigma_2}{C - \rho_1 - \rho_2} - \tau_3\right) + \frac{\sigma_3 + \rho_3\tau_3}{C - \rho_1 - \rho_2}$$

$$\overline{D_{3,WRR}} = \left(\frac{\phi_1 + \phi_2}{C} - \tau_3\right) + \frac{\sigma_3 + \rho_3\tau_3}{C\frac{\phi_3 - L_{3,max}}{\phi_1+\phi_2+\phi_3-L_{3,max}}}$$

Dans ce cas, si $\rho_1 + \rho_2 \to C$, le taux de service offert à une trame de priorité best effort tend vers zéro avec la politique à priorité stricte (effet famine). Ce problème n'apparaît pas avec la politique weighted round robin, puisque le taux de service ne dépend que des valeurs des poids. L'optimisation des poids ϕ_j est une perspective de recherche intéressante pour poursuivre ces travaux.

4.4 Validation expérimentale

Le paragraphe 4.2 a permis d'introduire la notion de validité du modèle. Dans le but de conforter la validité de notre modèle et la méthodologie de calcul des délais de bout en bout, une série de mesures expérimentales a été menée afin de confronter les valeurs calculées à la réalité (Georges *et al.*, 2005*a*),(Georges *et al.*, 2005*c*). Dans ces expérimentations, le temps pour transmettre une trame d'un émetteur au récepteur est mesuré à travers un réseau Ethernet commuté. Les résultats sont ensuite comparés aux majorants obtenus par le calcul réseau. De plus, les délais obtenus par simulation réseau (Comnet III) sont donnés à titre indicatif. Ce paragraphe nous permettra enfin d'illustrer la méthode de calcul des majorants du délai de bout en bout présentée à la page 82 sur un exemple simple.

Comme le notent (Pasztor et Veitch, 2001), les mesures unidirectionnelles de délais sont relativement complexes compte tenu des erreurs de mesure possible. Ceci est principalement dû aux problèmes temporels tels la non synchronisation des horloges et à l'ordonnancement des tâches des systèmes d'exploitation. La première problématique revient à s'assurer que les deux sondes (émetteur et récepteur) possèdent la même référence temporelle. Pour cela, il est possible d'utiliser des protocoles de synchronisation (NTP (Mills, 1991), GPS, IEEE 1588). Néanmoins, compte tenu de la gigue de ces protocoles et des faibles valeurs de délai que l'on sera amené à mesurer, nous avons choisi dans une première approche d'exécuter les processus émetteur et récepteur sur le même PC. Pour cela, ce PC est équipé de deux interfaces

Ethernet. L'intérêt est d'utiliser ainsi la même horloge de référence. La conséquence est d'augmenter la problématique de l'ordonnancement des tâches. En effet, les deux processus doivent obtenir un accès privilégié au processeur. De plus, la concurrence d'accès entre ces deux tâches doit être gérée.

Pour cela, le PC d'accueil de ces deux sondes est doté d'un système d'exploitation Linux qui permet de contrôler l'ordonnancement des tâches. Les sondes correspondent à des programmes écrits en C. L'utilisation de l'appel système `sched_setscheduler` de *Linux* (défini dans le standard POSIX (IEEE, 2004)), permet de mettre en œuvre la politique d'ordonnancement `SCHED_FIFO`. De plus, la priorité de ces deux processus est fixée à `sched_get_priority_max`. Enfin, afin de contrôler la concurrence entre les deux sondes, l'émetteur suspend son exécution juste après avoir émis une trame en appelant la fonction `usleep` jusqu'à la prochaine trame. Cela permet ainsi au récepteur d'être prêt et libre de réagir rapidement à l'arrivée d'une trame. Pour accentuer cet aspect, la priorité du processus émetteur pourra également être légèrement diminuée de façon à améliorer la réactivité du récepteur (signifie que l'on admet une certaine liberté sur l'émission des données).

Le modèle et le calcul sont ainsi évalués sur une première plateforme illustrée sur la figure 4.12.

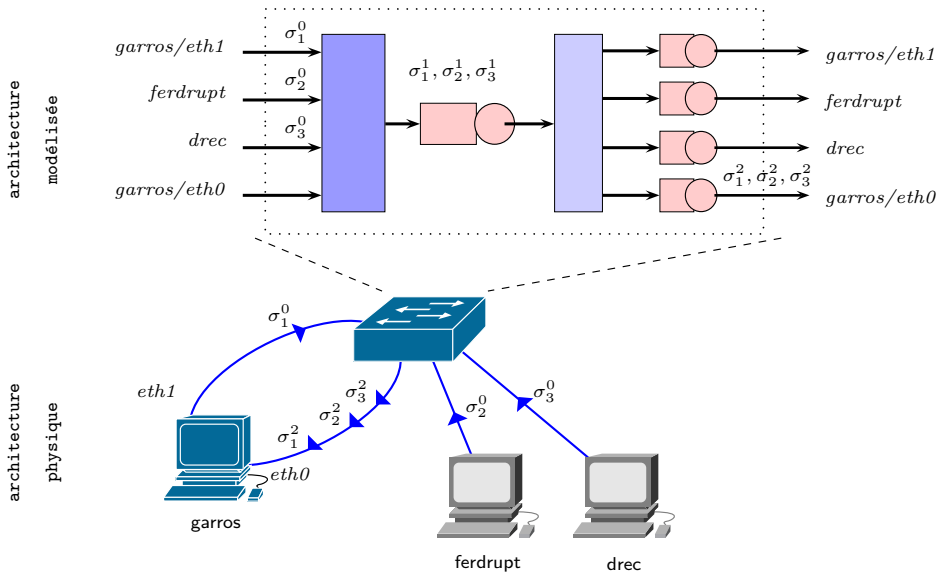


FIG. 4.12: Première plateforme expérimentale.

La plateforme est constituée d'un commutateur *Cisco Catalyst 2912 XL* et de trois PCs nommés *garros*, *ferdrupt* et *drec*. Les liens sont configurés à 10 Mb/s en mode full-duplex. Les communications sont générées par un algorithme exécuté par les trois stations. Cet algorithme qui utilise les raw-sockets permet de construire une trame Ethernet en précisant l'interface réseau de sortie, l'adresse MAC de destination, la longueur des trames et le temps d'inter-arrivée. *garros* émet périodiquement des trames de 72 octets (26 octets d'encapsulation plus la longueur minimale de données d'une trame Ethernet) depuis la première interface Ethernet (`eth1`) à destination de sa seconde interface (`eth0`). La période est fixée à 10 ms. Dans le but de charger le commutateur, un trafic de fond est généré : *ferdrupt* et *drec* émettent des trames de 1526 octets (26 octets d'encapsulation plus la longueur maximale du champ de données d'une trame Ethernet) chaque 5 ms à destination de la seconde interface `eth0` de *garros*.

L'étape initiale de la validation correspond alors à l'application de la méthode de calcul des délais de

bout en bout présentée dans ce chapitre. Tout d'abord, nous identifions les paramètres de la contrainte d'avalanche de données de chaque flux : de *garros*/`eth1` à *garros*/`eth0` (flux 1, contrainte $b_1^0(t)$), de *ferdrupt* à *garros*/`eth0` (flux 2, $b_2^0(t)$) et de *drec* à *garros*/`eth0` (flux 3, $b_3^0(t)$). Les paramètres des communications nous permettent d'obtenir alors :

$$b_1^0(t) = \sigma_1^0 + \rho_1 t = 72 + 7200t$$
$$b_2^0(t) = \sigma_2^0 + \rho_2 t = b_3^0(t) = \sigma_3^0 + \rho_3 t = 1526 + 305200t$$

Ensuite, il s'agit d'identifier les chemins parcourus par chaque flux. Ici, les chemins sont simples comme le montre la figure 4.12. Il convient également de définir le chemin à l'intérieur des commutateurs, ou plus précisément à l'intérieur du modèle (figure 4.12). A l'entrée, les flux qui arrivent *via* des ports différents sont regroupés dans la mémoire partagée par le multiplexeur. Ils sont ensuite redirigés vers la même file de sortie correspondante à l'interface `eth0` de *garros*. Pour identifier le délai généré par le multiplexeur et la file de sortie, nous définissons σ_1^1, σ_1^2 pour le flux 1 ; σ_2^1, σ_2^2 pour le flux 2 et σ_3^1, σ_3^2 pour le flux 3.

Pour chaque composant élémentaire du modèle, nous formulons alors les avalanches de données de sortie. On obtient le système d'équations suivant :

$$\begin{cases}
\frac{C}{\rho_1}\sigma_1^1 = \left(\frac{\rho_1}{C}+1\right)\sigma_1^0 + \sigma_2^0 + \frac{\rho_1+\rho_2}{C-\rho_3}\sigma_3^0 + \left(L_3 - (\rho_1+\rho_2)\frac{L_3}{C_3} + \rho_2\frac{L_2}{C_2}\right) & (1.1)\\
\frac{C_{out}(C-(\rho_1+\rho_2+\rho_3))}{\rho_1(C-C_{out})}\sigma_1^2 = \left(\frac{\rho_1(C-C_{out})}{C_{out}(C-(\rho_1+\rho_2+\rho_3))}+1\right)\sigma_1^1 + \sigma_2^1 + \sigma_3^1 & (1.2)\\
\frac{C}{\rho_2}\sigma_2^1 = \sigma_1^0 + \left(\frac{\rho_2}{C} + \frac{\rho_1+\rho_3}{C_2-\rho_2}\right)\sigma_2^0 + \sigma_3^0 + \left(\rho_1\frac{L_1}{C_1} + \rho_3\frac{L_3}{C_3}\right) & (2.1)\\
\frac{C_{out}(C-(\rho_1+\rho_2+\rho_3))}{\rho_2(C-C_{out})}\sigma_2^2 = \sigma_1^1 + \left(\frac{\rho_2(C-C_{out})}{C_{out}(C-(\rho_1+\rho_2+\rho_3))}+1\right)\sigma_2^1 + \sigma_3^1 & (2.2)\\
\frac{C}{\rho_3}\sigma_3^1 = \sigma_1^0 + \sigma_2^0 + \left(\frac{\rho_3}{C} + \frac{\rho_1+\rho_3}{C_3-\rho_3}\right)\sigma_3^0 + \left(\rho_1\frac{L_1}{C_1} + \rho_2\frac{L_2}{C_2}\right) & (3.1)\\
\frac{C_{out}(C-(\rho_1+\rho_2+\rho_3))}{\rho_3(C-C_{out})}\sigma_3^2 = \sigma_1^1 + \sigma_2^1 + \left(\frac{\rho_3(C-C_{out})}{C_{out}(C-(\rho_1+\rho_2+\rho_3))}+1\right)\sigma_3^1 & (3.2)
\end{cases}$$

Ces équations sont obtenues à partir des propositions de majorant du délai de traversée définis au chapitre 3. Les deux premières représentent l'évolution de l'avalanche de données du flux 1 : l'équation (1.1) vaut pour le multiplexeur et l'équation (1.2) pour la file de sortie. De la même façon, les équations (2.1), (2.2) sont relatives au flux 2 et les équations (3.1), (3.2) au flux 3. Ce système d'équation qui décrit pourtant un réseau très petit, montre que pour une architecture plus complexe, la dimension du système va sensiblement augmenter. En effet, l'évolution de l'avalanche de données est déterminée à chaque point du réseau et il n'y a pas d'agrégation de flux.

Une fois les valeurs d'avalanches σ calculées, l'étape suivante est de calculer $\overline{D_1} = \frac{\sigma_1^2 - \sigma_1^0}{\rho_1}$ afin de déterminer le délai maximum de traversée du réseau de bout en bout pour le flux 1. Le calcul donne : 3080 μs. Cette référence de 3080 μs va maintenant être comparé aux mesures expérimentales. Les résultats sont présentés sur la figure 4.13.

A titre de comparaison, nous avons également reproduit le scénario de communication avec l'outil de simulation réseau Comnet III. Pour cet exemple, l'outil donne un délai maximum de 450 μs. Si l'on compare alors ce résultat directement avec le majorant obtenu par le calcul réseau (3080 μs), la surestimation introduite par le calcul réseau est très importante. Une première explication peut alors être formulée au regard de la politique de retransmission du simulateur. Le traitement de l'historique des dates d'arrivées du simulateur ne permet pas de prendre en compte des arrivées simultanées répétées de deux trames. Si bien que le pire cas n'est dans cet exemple jamais considéré. Le délai de 450 μs correspond simplement à la somme des temps d'émission puisque la concurrence entre ces trames n'apparaît pas.

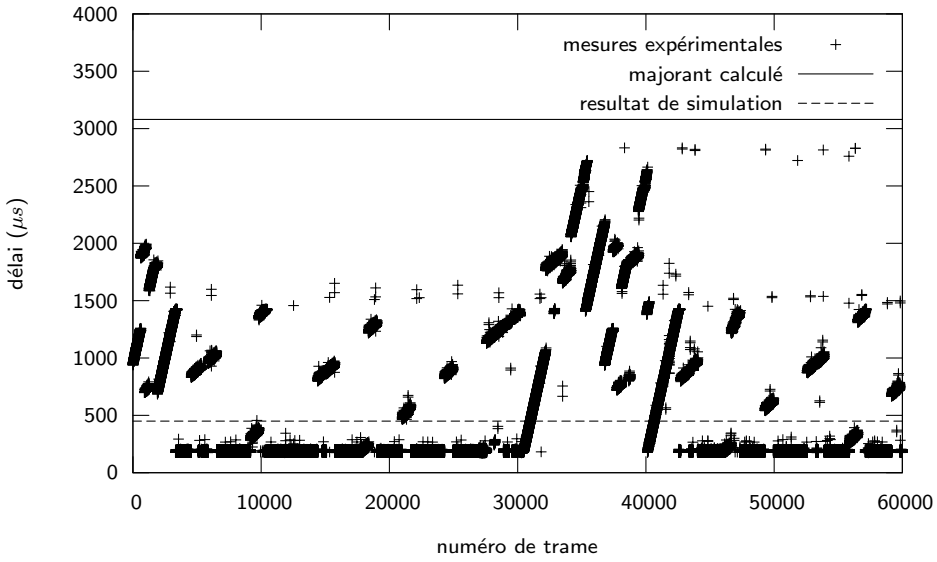


FIG. 4.13: Mesures des délais de bout en bout des trames émises par garros (flux 1) en μs.

Si l'on compare maintenant le majorant du calcul réseau aux mesures expérimentales présentées sur la figure 4.13, on peut cette fois remarquer que même si une très grande majorité des mesures est très inférieure à ce majorant, plusieurs mesures l'approchent, confirmant ainsi le majorant obtenu par calcul. En effet, plus de 56 % des mesures sont inférieures à 450 μs, la valeur de simulation (la moyenne des mesures est de 664 μs). Néanmoins, certaines observations de délais atteignent la valeur de 2832 μs, soit près de 6 fois plus. Cela montre qu'une analyse simpliste des capacités (la charge du lien entre le commutateur et la seconde interface `eth0` de *garros* est inférieure à 50 %) n'est pas suffisante pour valider le respect de contraintes temps-réel par un réseau Ethernet. De plus, cette expérimentation montre que le majorant du calcul réseau reste relativement proche des mesures les plus grandes puisque la surestimation est seulement de l'ordre de 8 % de la plus grande valeur mesurée. Finalement, cela montre que le calcul réseau répercute bien l'augmentation de la charge du commutateur introduite par le trafic de fond.

Une seconde expérimentation a été réalisée afin d'étendre la topologie à deux commutateurs. La plateforme est désormais composée de deux commutateurs *Cisco Catalyst 2912 XL*. Le reste de l'architecture reste la même comme le montre la figure 4.14.

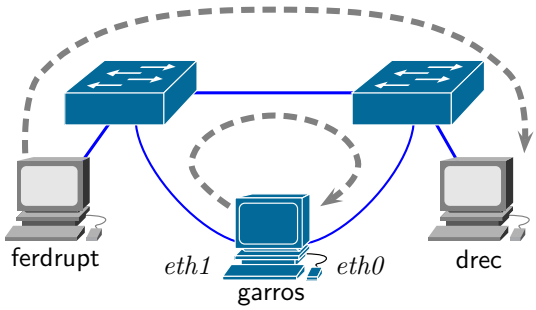


FIG. 4.14: Une seconde plateforme expérimentale.

garros émet des trames de 72 octets chaque seconde de `eth1` à sa seconde interface `eth0` pendant 10 minutes. Après 5 minutes (300 s), *ferdrupt* commence à émettre des trames de 1000 octets de données à *drec* chaque 10 ms. La figure 4.15 donne les mesures de délais observés tout au long des 10 minutes d'expérimentation.

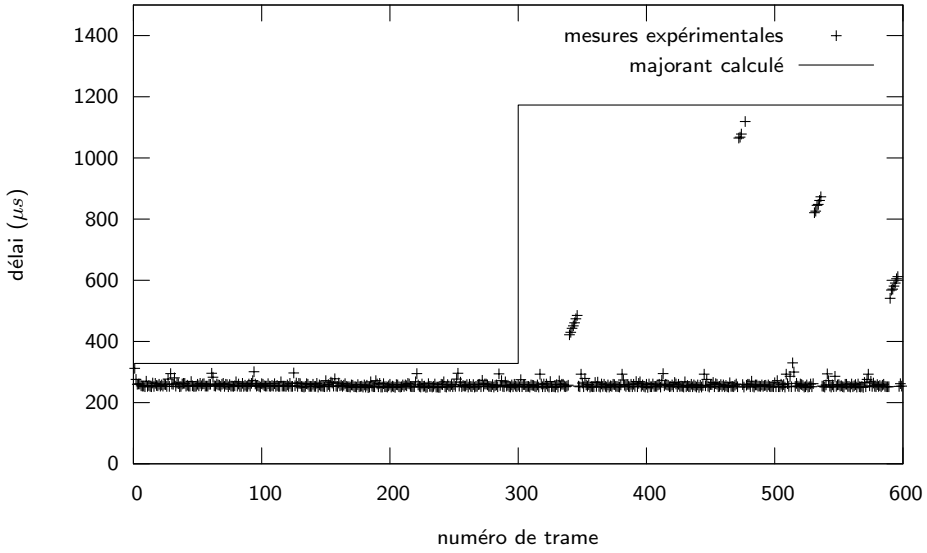


FIG. 4.15: Seconde série de mesures des délais de bout en bout des trames émises par garros (flux 1).

La figure 4.15 montre de nouveau que les mesures confirment le majorant du délai obtenu par le calcul réseau. En effet, durant les 5 premières minutes, les mesures restent inférieures à 312 μs alors que le calcul nous a permis de déterminer une valeur maximale de 328 μs. Pour les 5 dernières minutes (300 s avec le trafic de fond), nous avons obtenu un majorant de 1173 μs alors que le plus grand délai observé est de 1119 μs. En fait, la surestimation est dans les deux cas inférieure à 5 % du délai maximum mesuré. De plus, le graphe montre que notre approche met en lumière des situations défavorables réelles. Durant la deuxième partie de l'expérimentation, même si 91 % des mesures sont restées inférieures à 328 μs (le majorant défini par le calcul pour la première partie), nous observons qu'un simple trafic de fond peut significativement augmenter le pire délai comme déterminé par le calcul. Et même si la fréquence d'apparition est faible, les situations extrêmes envisagées par le calcul sont réellement observables.

Ces expérimentations montrent que les valeurs mesurées proches de la borne supérieure sont effectivement rares en comparaison avec l'ensemble des mesures, mais qu'un résultat en moyenne est relativement éloigné des valeurs obtenues dans le pire cas. La figure 4.13 montre que certaines valeurs correspondent à plus de 300 % de la valeur moyenne. Ceci renvoit au coût de la contrainte du déterminisme des résultats. Baser une évaluation de performances uniquement à partir de résultat en moyenne peut passer à côté de valeurs (rares mais réelles) 3 fois plus grandes, et se baser sur une évaluation à partir du calcul réseau apporte une meilleure garantie mais ne reflète que peu de valeurs. Dans le cadre actuel de la recherche dans le domaine des SCRs, il reste toutefois nécessaire de construire une commande robuste au pire cas. La spécificité de notre démarche d'évaluation basée sur l'augmentation du volume des rafales conduit effectivement à prendre en compte des situations que l'on ne pourra pas toujours observer en réalité,

même dans le pire cas. Toutefois, nous avons pu relever que cette précision des bornes est davantage liée à la précision des modèles (commutateur et flux d'arrivée) et plus particulièrement de la précision de la courbe d'arrivée. Ceci est d'autant plus vrai que notre démarche de calcul est basée sur l'augmentation du volume des rafales ce qui suppose que la contrainte de volume maximal d'avalanche des données en entrée du réseau soit la plus précise.
Le contexte des systèmes contrôlés en réseau offre un cadre de travail où les communications sont parfaitement définies et la loi de départ des trames peut par exemple être directement modélisé par un réseau de Petri ou un automate temporisé. Ceci s'applique ainsi très bien au réseau chargé d'interconnecter des automates où la description des tâches est déjà clairement formalisée. Dans le cadre des systèmes contrôlés en réseau, la courbe d'arrivée affine que nous avons retenue sera généralement précise puisque les échanges sont généralement régient de manière cyclique et qu'il a été montré que les majorants obtenus *via* des courbes d'arrivées affines tendent dans ce cas au même majorant que pour des courbes en escalier. Ainsi, même pour un réseau incorporant 4 commutateurs (voir paragraphe 5.2.3), les bornes restent proches des valeurs mesurées dans le pire cas, la surestimation étant inférieure à 8 %. Si bien qu'en fait, il n'est pas directement possible de fournir un commentaire général concernant la précision des bornes dans la mesure où elle va dépendre de la configuration donnée et de la précision des modèles. De même, la pertinence de ces bornes est à mettre en relation avec la contrainte de déterminisme établie par l'application et les caractéristiques de la commande.

En conclusion, cette série d'expérimentation nous a permis de valider le majorant obtenu par le calcul à partir des mesures réelles. L'association du calcul réseau et du modèle de commutateur proposé dans ces travaux permet d'avoir une bonne idée de la performance de ce type de réseau. Afin de développer cette validation, il sera intéressant de reproduire et d'étendre ces expérimentations à d'autres *scenarii* et d'autres architectures. C'est notamment dans cet esprit que nous avons cherché à valider les majorants obtenus par calcul sur une plateforme expérimentale plus complexe intégrant 15 stations et 4 commutateurs. La plateforme ainsi que les résultats obtenus sont présentés au paragraphe 5.2.3.

4.5 Conclusion

En conclusion, ce chapitre développe les différentes étapes de la méthodologie qui permet de majorer les délais de bout en bout sur un réseau Ethernet commuté. Dans un premier temps, la philosophie générale développée dans ce chapitre repose sur le cycle continu qui veut que dans le pire des cas l'avalanche de données, l'arriéré de traitement et le délai de traversée soient liés et que l'augmentation d'un de ces trois paramètres entraîne l'augmentation des deux autres.

Dans le premier paragraphe, nous avons proposé ainsi un cadre d'étude de majoration des délais de bout en bout. Cette étude présente la particularité de rechercher systématiquement le pire cas, et suppose plus particulièrement la synchronisation des dates d'arrivées des avalanches de données en tout point du réseau. Si cette adaptation du calcul réseau risque d'induire un pessimisme fort, nous pensons que la précision des majorants obtenus dépend de la précision des modèles (arrivée de données et commutateur).

Concernant les modèles initialement proposés, nous avons mis en évidence la pertinence du modèle n°3 par rapport aux deux autres modèles initialement proposées. Ce modèle a également été enrichi afin de prendre en compte les mécanismes de Classification de Service introduit dans le standard 802.1D/p.

Enfin, la dernière partie de ce chapitre a permis de montrer que la méthodologie proposée donne des majorants réalistes et suffisamment précis par rapport au contexte d'utilisation dans les systèmes contrôlés en réseau. Cette validation des résultats est ainsi un encouragement au développement de modèles basés sur une analyse fonctionnelle.
L'étape suivante de ces travaux sera donc la validation expérimentale du modèle de commutateur à priorité proposé dans ce chapitre. Cela reste toutefois complexe compte tenu de la difficulté à paramétrer et à maîtriser la configuration de la classification de service dans les commutateurs existants. En effet, les informations fournies par les constructeurs sont à ce moment trop peu explicites. Néanmoins, de la même façon que pour les commutateurs sans priorité, il est attendu que les mesures expérimentales confirment la validité du modèle.

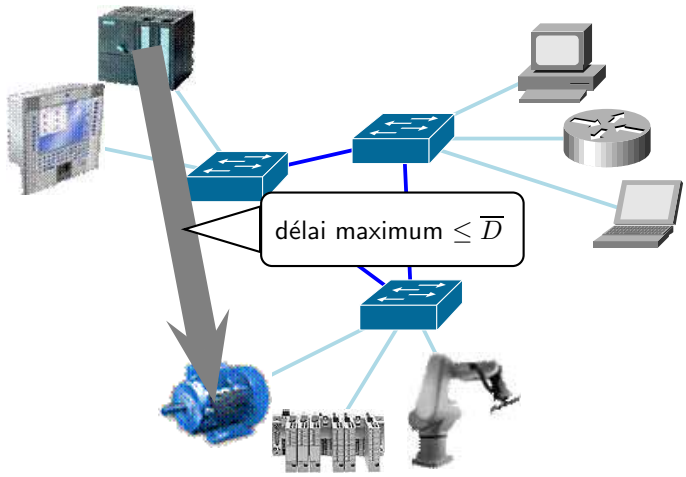


FIG. 4.16: Majoration des délais de traversée de bout en bout pour chaque flux

En conclusion, nous sommes à ce stade capable de répondre à l'interrogation concernant les délais portés à la figure 1.14 en fin de premier chapitre. Comme le montre la figure 4.16, les délais de traversée du réseau de chaque flux peuvent être désormais bornés et il est possible de déterminer si le niveau de performance est suffisant pour la stabilité de l'application distribuée par exemple.

Chapitre 5

Application aux systèmes contrôlés en réseau (Ethernet commuté)

L'objectif de ce dernier chapitre est d'illustrer l'intérêt pour les systèmes contrôlés en réseau de la méthode d'évaluation des performances d'un réseau Ethernet commuté présentée au chapitre 4. Pour cela, trois types d'applications distribuées sont étudiés. La première montre comment le calcul réseau peut être utilisé pour dimensionner différentes caractéristiques comme la quantité de données des flux apériodiques admissibles compte tenu des contraintes temporelles du flux périodique. La seconde application valorise notre méthode dans le cadre de la conception d'architectures Ethernet commutées. Dans cette optique, le calcul réseau est utilisé comme théorie support de la fonction d'évaluation d'une heuristique de partitionnement de graphes. Enfin, la dernière application illustre comment la méthode proposée permet d'identifier l'apport de la classification de service dans les environnements commutés.

Les résultats présentés dans la suite de ce chapitre ont été obtenus à l'aide d'un logiciel  Cerise (*Conception de réseaux industriels switched Ethernet*) développé au laboratoire en Java. La figure 5.1 montre l'interface graphique du programme.

A partir d'une matrice d'échange, le programme applique l'algorithme de détermination de majorants de délais de bout en bout. Dans sa version 1.0, le logiciel est capable d'évaluer les réseaux Ethernet commutés soit en topologie hiérarchique ou linéaire. Le programme intègre également différentes heuristiques de partitionnement de graphes (Krommenacker, 2002) qui seront utilisées dans le paragraphe 5.2.

5.1 Dimensionnement d'architectures pour les systèmes contrôlés en réseau (Georges *et al.*, 2002)

Dans cette première partie, le calcul réseau est employé de manière à caractériser un réseau autour de quatre grands paramètres. La première étude s'intéressera à identifier la charge de trafic supportable par un réseau industriel Ethernet commuté. Il sera également question de déterminer l'influence du nombre de ports des commutateurs. Ensuite, on analysera les conséquences de l'ajout de trafic non-critique. Enfin, le dernier point abordé permettra d'étudier l'influence de la topologie sur les performances du réseau.

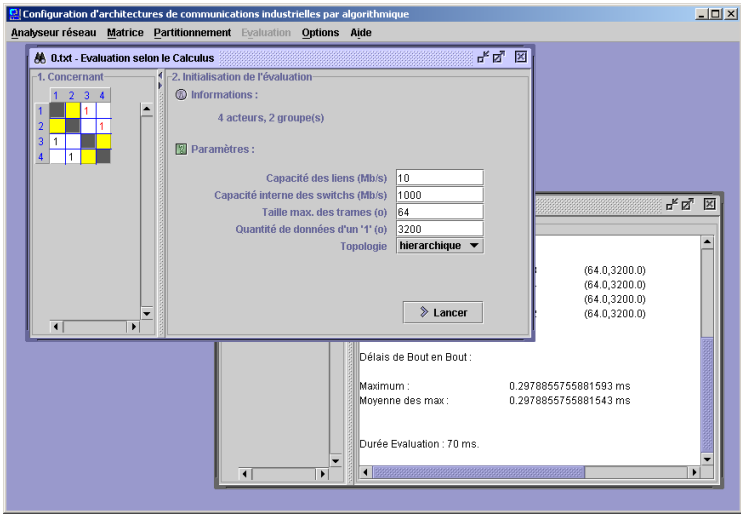


FIG. 5.1: Capture d'écran de l'outil Cerise

Pour cela, considérons comme réseau d'étude, le réseau commuté de la figure 5.2.

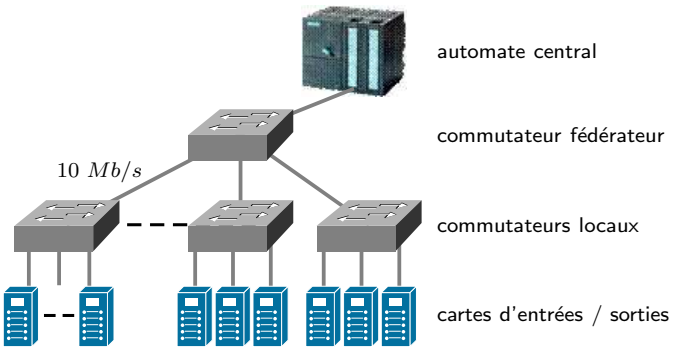


FIG. 5.2: Réseau d'étude à topologie hiérarchique

L'application industrielle étudiée consiste à attacher un automate principal au commutateur fédérateur et des cartes d'entrées / sorties déportées aux autres commutateurs. Le trafic de base supporté par l'architecture est constitué de messages périodiques. Les paquets présentent tous la même taille : 72 octets. Chaque paquet est envoyé toutes les 10 ms (valeur typique des temps de cycle des automates). Les échanges considérés relèvent d'un comportement maître-esclaves. L'automate émet périodiquement des messages vers chacune des cartes, qui lui répondent. Enfin, nous considérerons que le processus de routage des paquets au sein des commutateurs se fait au taux de 1 Gb/s, que les liens sont full-duplex et ont un débit de 10 Mb/s. Les commutateurs ne gèrent pas les priorités.

5.1.1 Passage à l'échelle du réseau et influence de la fragmentation

L'objectif de ce paragraphe est double. Il s'agira dans un premier temps de déterminer le nombre maximum de cartes d'entrées / sorties que l'architecture peut supporter (en garantissant que les délais maxima de bout en bout soient inférieurs au temps de cycle de l'automate) et dans un second temps, de

visualiser l'influence du nombre de ports dans ce passage à l'échelle du réseau. Ces études se justifient par le fait, que sur certains sites comme celui de Peugeot à Mulhouse, on peut compter jusqu'à 4000 équipements à raccorder et que le prix d'un commutateur est en partie déterminé par son nombre de ports.

Le calcul réseau est alors utilisé pour déterminer un majorant du délai de bout en bout de chaque paquet. Si tous ces délais sont inférieurs au temps de cycle de l'automate, alors l'organisation du réseau sera déclarée en accord avec les contraintes applicatives.

Initialement, le trafic de chaque carte est associé à un sous-flux caractérisé par la représentation du sceau percé $b_c^0 = \sigma_c^0 + \rho_c t$ où $\sigma_c^0 = 72$ (la longueur des paquets) et $\rho_c = 7200$ (le taux d'arrivée des données est égal à $7,2\ kb/s$). Le trafic généré par l'automate se révèle être une collection de k (le nombre de cartes) sous-flux, caractérisés par $b_{pc}^0 = \sigma_{pc}^0 + \rho_{pc} t$ où $\sigma_{pc}^0 = 72$ et $\rho_{pc} = 7200$. Dans ce scénario de communication, les liens entre les commutateurs sont toujours les liens les plus chargés. Par souci de simplification du problème, nous supposerons que k est un multiple du nombre de ports des commutateurs, noté n.
Les expressions suivantes du volume maximal d'avalanche de données suite à la traversée d'un composant constitutif d'un même commutateur sont ici regroupées en seule expression du volume maximal d'avalanche de données en sortie du commutateur *via* la relation établie à l'équation 4.4.

En conséquence, la charge des commutateurs locaux est caractérisée par

$$\sigma_c^1 = a_{c,1}\left[(n-1)\sigma_c^0 + B_{c,1}\sigma_c^0 + nl_{c,1}\sigma_{pc}^1 + g_{c,1}\right]$$
$$\sigma_{pc}^2 = a_{pc,2}\left[n\sigma_c^0 + (n-1)\, b_{pc,2}\sigma_{pc}^1 + B_{pc,2}\sigma_{pc}^1 + g_{pc,2}\right]$$

De plus, l'application du théorème 4.3 au niveau du commutateur fédérateur nous donne :

$$\sigma_c^2 = a_{c,2}\left[(k-n)\sigma_c^1 + (n-1)\, b_{c,2}\sigma_c^1 + B_{c,2}\sigma_c^1 + kl_{c,2}\sigma_{pc}^0 + g_{c,2}\right]$$
$$\sigma_{pc}^1 = a_{pc,1}\left[k\sigma_c^1 + (k-1)\, b_{pc,1}\sigma_{pc}^0 + B_{pc,1}\sigma_{pc}^0 + g_{pc,1}\right]$$

Une fois les différentes étapes de l'algorithme 4.1.3 exécutées, l'équation des volume des rafales de sortie s'écrit :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ (n-1)+B_{c,1} & -\frac{1}{a_{c,1}} & 0 & 0 & nl_{c,1} & 0 \\ 0 & (k-n)+(n-1)b_{c,2}+B_{c,2} & -\frac{1}{a_{c,2}} & kl_{c,2} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & k & 0 & (k-1)b_{pc,1}+B_{pc,1} & -\frac{1}{a_{pc,1}} & 0 \\ n & 0 & 0 & 0 & (n-1)b_{pc,2}+B_{pc,2} & -\frac{1}{a_{pc,2}} \end{bmatrix}^{-1} * \begin{bmatrix} \sigma_c^0 \\ \sigma_c^1 \\ \sigma_c^2 \\ \sigma_{pc}^0 \\ \sigma_{pc}^1 \\ \sigma_{pc}^2 \end{bmatrix} = \begin{bmatrix} 64 \\ g_{c,1} \\ g_{c,2} \\ 64 \\ g_{pc,1} \\ g_{pc,2} \end{bmatrix}$$

La résolution de cette équation est répétée pour différentes valeurs de k et de n. Les résultats sont donnés sur la figure 5.3.

Le graphe ci-dessus montre alors que le réseau est capable de connecter jusqu'à 3000 cartes d'entrées / sorties pour des commutateurs 24 ports, jusqu'à 1800 pour des commutateurs 12 ports et jusqu'à 1200 pour des commutateurs 8 ports. Au-dessus de ces valeurs, on constate que les délais de bout en bout franchissent la barre des 10 ms.

En ce qui concerne le passage à l'échelle, nous observons donc qu'une architecture Ethernet commutée autorise un grand nombre de cartes tout en respectant la contrainte de temps de cycle de l'automate. Il est également à remarquer l'importance de la fragmentation du réseau (c'est-à-dire, du nombre de

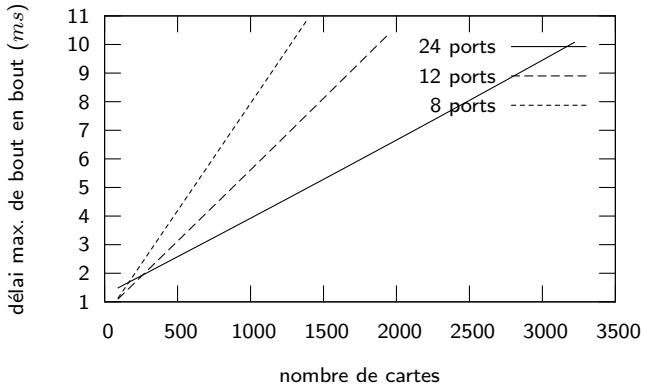


FIG. 5.3: Evolution de l'acceptation de charge du réseau de la figure 5.2

ports des commutateurs) étant donné qu'elle réduit par deux le nombre de cartes entre les solutions à commutateurs 24 ports et 8 ports.

5.1.2 Capacité d'accueil de trafic non-critique

Nous étudions désormais une organisation du réseau constituée de commutateurs 24 ports connectant 276 cartes d'entrées / sorties. Dans ce cas, nous avons cherché à calculer la quantité maximale de trafic apériodique supplémentaire au trafic périodique pouvant être admise. Il est supposé que la longueur des trames apériodiques est fixée à 1526 octets. Notre modèle ne prenant pas encore en compte la différenciation de services, l'ajout de ce trafic non-critique gonfle artificiellement la charge du réseau et réduit ses performances. C'est ce que montre la figure 5.4.

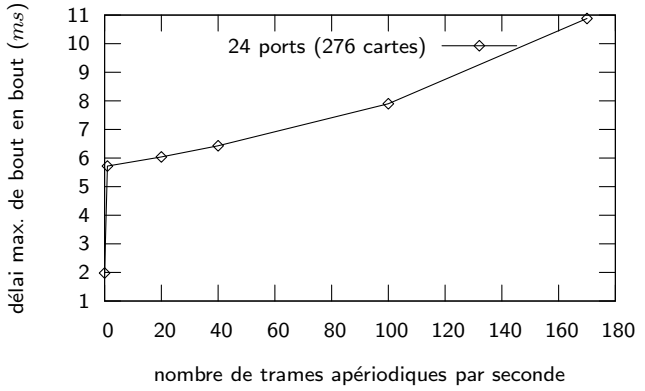


FIG. 5.4: Influence du trafic apériodique sur le réseau de la figure 5.2

La figure 5.4 nous montre que la limite du nombre de trames apériodiques admissibles par seconde est proche de 150. Dans ce cas, on constate donc que seulement $150 * 1526 \simeq 230\ ko/s$ de trafic supplémentaire peut être ajouté sur chaque carte pour garantir que les délais maxima de traversée des trames du flux périodique restent inférieurs au temps de cycle de l'automate. La figure 5.4 fait également apparaître que cet ajout de trafic n'est pas sans conséquence puisque le simple ajout d'un paquet apériodique au niveau de chaque carte produit une nette augmentation des délais.

5.1.3 Comparaison des topologies

Principalement deux types de topologies sont utilisés dans les réseaux Ethernet commutés (Rüping *et al.*, 1999). La première est la topologie hiérarchique, dont l'aspect est donné en figure 5.2. La seconde, appelée topologie linéaire, consiste en une cascade de commutateurs reliés deux à deux comme illustré sur la figure 5.5.

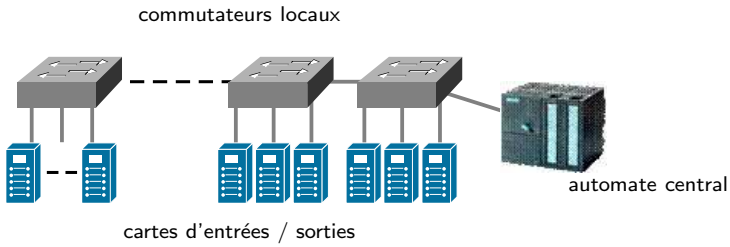


FIG. 5.5: Réseau d'étude à topologie linéaire

Par rapport à l'organisation hiérarchique du réseau de la figure 5.2, la distribution des cartes d'entrées / sorties reste la même en topologie linéaire. Seul l'automate est déplacé sur un des commutateurs locaux en bout de ligne.

De la même manière que précédemment, le calcul réseau nous donne la caractérisation du passage à l'échelle du réseau de la figure 5.5 en topologie linéaire (figure 5.6).

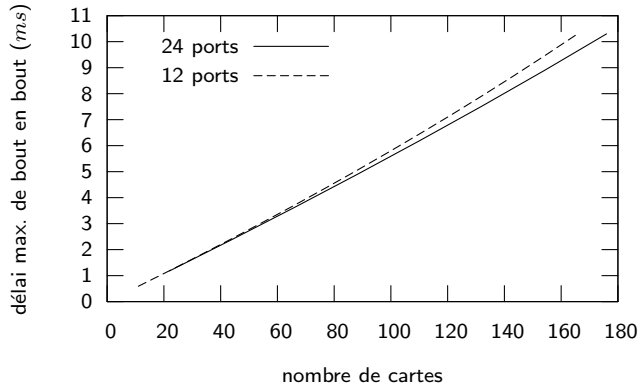


FIG. 5.6: Evolution de l'acceptation de charge du réseau de la figure 5.5

Dans ce cas, seulement 160 cartes d'entrées sorties pour des commutateurs 24 ports peuvent être supportées par le réseau contre 3000 en topologie hiérarchique. Cela signifie que le passage à l'échelle de la topologie linéaire s'avère être relativement moins bon que pour le hiérarchique. Ce résultat était toutefois attendu compte tenu du fait qu'en topologie hiérarchique, le nombre maximal de commutateurs à traverser est de trois, alors qu'en linéaire, il peut aller jusqu'à 16 pour 160 cartes avec des commutateurs 12 ports.

A l'inverse, le nombre de ports des commutateurs n'a pas de réelle incidence en configuration linéaire.

Enfin, en ce qui concerne la quantité de trafic apériodique acceptable par le réseau (figure 5.7), précisons simplement que pour un réseau présentant 16 cartes d'entrées sorties, 763 ko/s de trafic peut être ajouté au trafic périodique critique déjà existant.

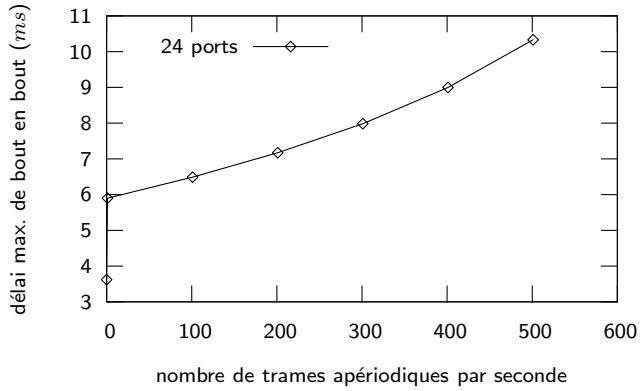


FIG. 5.7: Influence du trafic apériodique sur le réseau de la figure 5.5

5.2 Efficacité du plan de câblage d'architectures commutées pour des applications à temps-réel

L'architecture d'un réseau Ethernet commuté n'est pas sans conséquence sur les performances temps-réel du réseau. Différents travaux menés au CRAN se sont ainsi intéressés à l'optimisation du plan de câblage en vue de minimiser la charge des commutateurs et d'améliorer ainsi les délais de traversée. Nous présentons dans cette partie nos travaux (Krommenacker *et al.*, 2002*b*)(Georges *et al.*, 2004*b*)(Georges *et al.*, 2005*d*) dont le but est d'améliorer le critère d'évaluation du réseau en se basant directement sur les majorants des délais de bout en bout.

L'objectif est donc d'optimiser l'organisation du réseau au niveau de la couche physique en distribuant 'intelligemment' les équipements sur les commutateurs Ethernet afin de réduire leur charge. Comme l'indiquent (Krommenacker *et al.*, 2002*a*; Krommenacker *et al.*, 2001), cette optimisation correspond en fait à un problème mathématique de partitionnement de graphes d'échanges de ces équipements. Le lecteur intéressé par cette problématique pourra se référer aux travaux de (Krommenacker, 2002). Dans ce document, nous nous concentrons sur l'apport du calcul réseau à l'optimisation de la topologie.

La figure 5.8 résume les différentes étapes du processus de conception de réseaux Ethernet commuté. Elle illustre bien le mécanisme d'optimisation d'un plan de câblage réseau. Dans l'étape 1, les stations A et B s'échangent des informations et sont connectées sur deux commutateurs différents. On remarque que les commutateurs sont alors fortement chargés.

Dans l'étape 2, l'ensemble des trafics entre les équipements est représenté à l'aide d'un graphe. Les arcs du graphe traversant la ligne en pointillé montre le trafic inter-commutateur (passant par le commutateur fédérateur). L'étape 3 indique la partition optimale (ligne en pointillé) réduisant le nombre de communication inter-commutateurs. L'étape 4 donne le nouveau plan de câblage et montre une réduction de la charge des commutateurs.

Ainsi, l'objectif est de regrouper sur les mêmes commutateurs les équipements qui s'échangent le plus d'information. Aussi, le critère de minimisation traditionnellement utilisé est le nombre d'arcs du graphe coupé par la partition. L'idée sous-jacente est que les délais de bout en bout seront réduits si les deux acteurs appartiennent à la même partition puisque le nombre de commutateurs à traverser sera réduit. Toutefois aucune garantie concernant les valeurs des délais n'est formulée. D'où l'originalité de

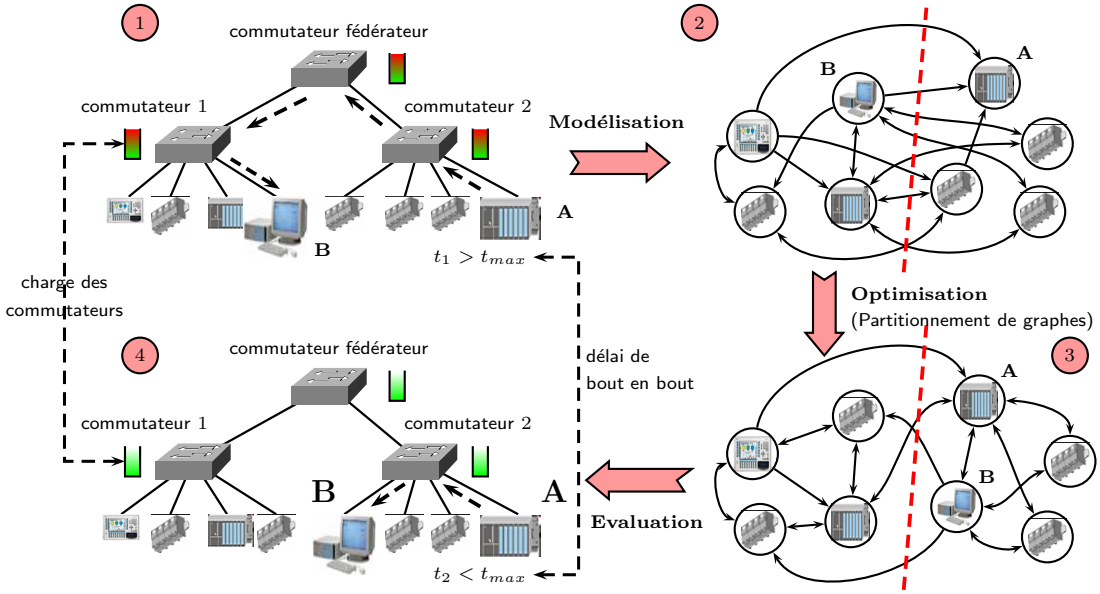


FIG. 5.8: Etapes de la conception d'une architecture Ethernet commutée

nos travaux présentés dans (Georges *et al.*, 2004*b*) qui visent à incorporer le calcul réseau pour que le plan de câblage réseau corresponde à une solution satisfaisante au vu des critères temporels de l'application.

5.2.1 Problème de partitionnement

La théorie des graphes a souvent été utilisée pour étudier la topologie réseau. La modélisation de l'architecture Ethernet commuté s'appuie sur un graphe d'échange qui représente les communications entre les équipements (figure 5.9). Ce graphe $G = (V, E)$ est défini par un ensemble V de sommets (les automates, capteurs, actionneurs ...) et un ensemble E d'arcs (les échanges). Des poids peuvent alors être appliqués sur les arcs pour intégrer les quantités d'information échangées.

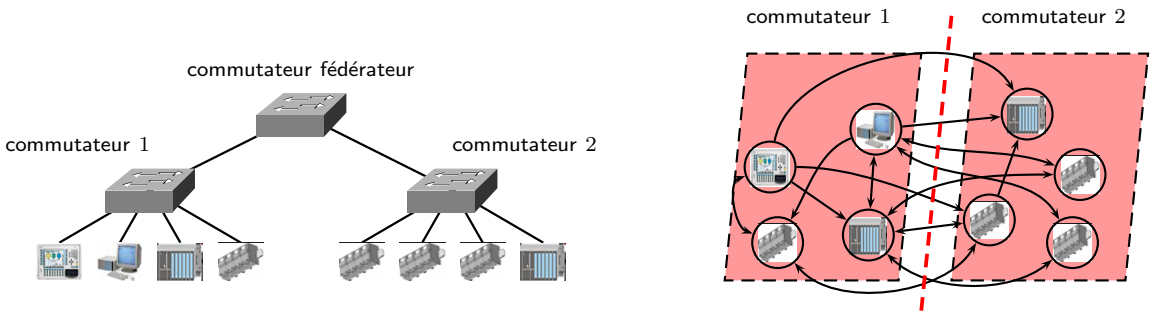


FIG. 5.9: Représentation d'un réseau Ethernet commuté par un graphe

Le regroupement de plusieurs équipements sur un même commutateur correspond alors à la définition d'un sous-ensemble de sommets correspondant à ces équipements. Ce sous-ensemble est appelé partition (P). Comme le montre la figure 5.9, la qualité de la distribution (ou partitionnement) est alors fonction du nombre d'arcs entre les partitions.

Le k partitionnement consiste à diviser les sommets d'un graphe donné en k sous-ensembles disjoints appelés partitions (P) qui sont sujettes à différentes contraintes. Le problème du partitionnement peut alors être exprimé comme l'optimisation d'un objectif unique comme la taille de coupure qui correspond au nombre d'arcs inter-partitions (d'échanges entre deux équipements interconnectés par deux commutateurs différents). (Krommenacker, 2002) montre alors que la complexité du problème d'optimisation du partitionnement nécessite d'utiliser une heuristique et propose pour cela une adaptation des algorithmes génétiques au problème de conception de réseaux Ethernet commuté. Nous proposons alors une fonction objectif chargée d'évaluer les populations de solutions parcourues par les algorithmes génétiques en fonction du degré de satisfaction des exigences temporelles de l'application.

5.2.2 Fonction objectif

La qualité d'une solution sera donc définie par rapport au respect des contraintes temporelles. Aussi, la fonction d'évaluation doit prendre en compte les délais de bout en bout. Dans de précédents travaux ((Kamen *et al.*, 1999), (Torab, 2000)), l'objectif s'est porté sur la réduction des délais moyens de bout en bout. La nécessité d'une théorie déterministe ayant déjà été discutée, nous avons proposé la fonction objectif suivante :

$$f = \max_i \left\{ \overline{D_i} - T_i \right\}$$

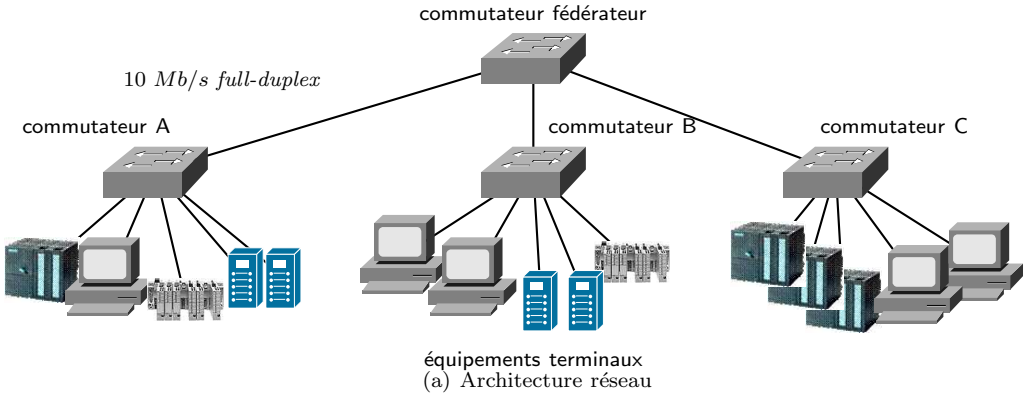
avec i l'identifiant du message, $\overline{D_i}$ le majorant du délai de traversée de bout en bout de ce message et T_i la borne temporelle liée à ce message. La topologie devra donc être optimisée jusqu'à ce que cette fonction devienne négative.

5.2.3 Application

La première étape consiste à déterminer le nombre de commutateurs nécessaire à la distribution de l'application. Cette information peut être imposée par l'architecte à l'aide de techniques données dans (Adoud *et al.*, 2001). Les bénéfices de la méthode précédemment proposée sont alors mis en avant au travers du scénario suivant.

15 équipements industriels doivent être interconnectés à l'aide de trois commutateurs. Les échanges sont représentés sur la figure 5.10(b). Les poids définis par la matrice d'échange représentent la quantité de données émise par un équipement à un autre. Par exemple, l'équipement 3 émet à l'équipement 11, $4 *$ (longueur minimale des données) par temps de cycle. La longueur minimale des données est de 72 octets et il est supposé qu'un équipement émet des trames toutes les 2 ms (ce sera le temps de cycle des automates). Par conséquent, le taux d'arrivée des données émises par l'équipement 3 à destination de l'équipement 11 est de $17,57\ ko/s$.

Afin de vérifier qu'une architecture Ethernet commuté hiérarchique peut être utilisée pour cette application (avec des délais de bout en bout inférieurs au temps de cycle), les algorithmes génétiques sont exécutés avec la fonction objectif établie au paragraphe précédent. Une solution arbitraire et une solution optimisée sont comparées. Pour la première, les équipements sont distribués de la sorte : $\{1, 2, ...5\}$ sur le premier commutateur, $\{6, ...10\}$ sur le second et $\{11, ...15\}$ sur le troisième. L'architecture 'optimisée' obtenue par les algorithmes génétiques regroupe les équipements comme suit : $\{1, 4, 6, 7, 14\}$ sur le premier commutateur, $\{8, 9, 11, 12, 15\}$ sur le second et $\{2, 3, 5, 10, 13\}$ sur le troisième. Les résultats sont alors confrontés avec ceux fournis par l'outil de simulation réseau *Comnet III* (figure 4.3).



(a) Architecture réseau

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1						2									
2	1											3			
3					1						4		1		
4							1		1						
5				1								1			
6	1			2	1					2	4				
7															
8			3												4
9															
10															
11					1			1				1			2
12	3			1											
13		4								3					
14		1		2											2
15					1				1		1	3			

(b) Matrice d'échange

FIG. 5.10: Cas d'étude de la méthode d'optimisation de la topologie.

De plus, des mesures expérimentales ont été menées afin de valider l'évaluation des résultats pour les deux topologies retenues. Le principe de ces mesures est le même que celui présenté au chapitre précédent. Le calcul réseau permet alors d'identifier les communications présentant les plus grands délais. Il s'agit pour la topologie arbitraire de la communication entre les équipements 12 et 1, et 6 et 11 pour la solution optimisée. Ces résultats obtenus *via* le logiciel Cerise sont présentés dans le tableau 5.1.

Notons par ailleurs que le nombre d'inconnues pris en compte par le calcul réseau passe de 188 pour la solution arbitraire à 160 pour la solution optimisée. Cela est évidemment dû au fait que les chemins sont plus courts avec la solution optimisée. Ceci peut servir de premier indicateur de l'efficacité de la solution et de l'intérêt de notre proposition.

TAB. 5.1: Performance des architectures (valeurs en millisecondes)

architecture	calcul réseau	comnet	mesures
arbitraire	2,55	1,68	2,38
optimisée	1,96	1,53	1,94

Le tableau 5.1 montre que les délais sont réduits si la méthodologie de distribution des équipements industriels sur les commutateurs est appliquée. En effet, la théorie du calcul réseau garantit pour la solution optimisée que les délais seront inférieurs au temps de cycle (2 ms), alors que pour la solution non optimisé, ceci n'est pas respecté.

Les résultats de la simulation montrent également que les délais de bout en bout sont réduits. Toutefois, ces résultats n'indiquent pas un dépassement de la contrainte temporelle dans le cas de la solution arbitraire, si bien que cette topologie pourrait être considérée comme satisfaisante.

Les mesures expérimentales confirment alors les indications portées par le calcul réseau et montrent bien que l'étude du pire cas est nécessaire pour les applications à temps critique. Elles montrent également que la réduction des délais n'est pas suffisante et qu'il convient de comparer ces délais aux exigences de l'application.

5.3 Configuration de la CdS par rapport aux niveaux de QdS exigées par l'application

Considérons maintenant le réseau représenté dans la figure 5.11, qui regroupe trois commutateurs gérant les priorités reliés avec des liens Ethernet à 10 Mb/s.

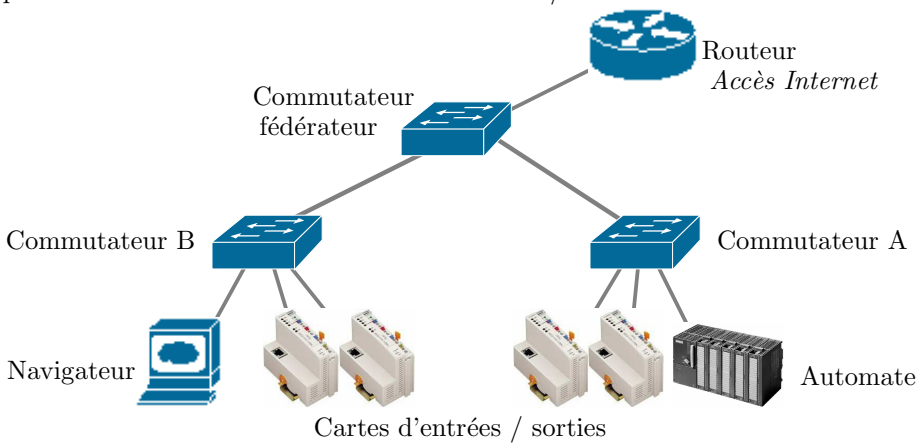


FIG. 5.11: Cas d'étude

Cette application supporte un automate, quatre cartes d'entrées/sorties et un navigateur web qui génèrent trois types de trafic :

- le trafic de contrôle : l'automate envoie tous les 10 ms (temps de cycle) une trame sur chaque carte d'entrées/sorties. Et inversement, ces cartes émettent à la même fréquence une trame vers l'automate. Ces messages supportent une quantité d'informations relativement faible et tiennent donc dans une trame Ethernet de 72 *octets* correspondant à la taille minimale.
- les alarmes : les deux cartes connectées sur le commutateur B peuvent générer des alarmes vers l'automate. Ces alarmes correspondent aussi à une trame Ethernet de 72 *octets*.
- le trafic apériodique utilisé pour des opérations de maintenance : le navigateur web est mis à disposition des opérateurs qui peuvent faire par exemple des opérations de téléchargement de nouveaux pilotes, de documents, etc. La bande passante pour le trafic Internet entrant dans le réseau local est limitée par le routeur à 1 Mo/s. Pour ce flux, des trames Ethernet de 1526 octets sont utilisées

(taille maximale).

Nous détaillons alors les différentes étapes du calcul des majorants du délai de traversée de bout en bout pour cet exemple.

Détermination des courbes d'arrivées : tout d'abord, le trafic de contrôle (de nature périodique) est modélisé par une contrainte d'avalanche des données dans laquelle l'avalanche maximale est initialisée à la taille minimale de la trame Ethernet, c'est-à-dire 72 octets et le débit moyen d'arrivée est de 7200 octets/s. Cela donne $b^0_{aut}(t) = b^0_{carte}(t) = 72 + 7200t$. Ensuite, puisque le routeur régule les informations extérieures entrant dans le réseau local, le trafic web est modélisé par une contrainte d'avalanche des données dont la rafale correspond à la taille maximale d'une trame Ethernet (1526 octets) et le débit moyen est égal à 1 Mo/s. D'où $b^0_{web}(t) = 1526 + 1048576t$. Enfin, pour les alarmes, il n'est pas possible de le représenter directement par un seau percé car leurs occurrences sont inconnues. Pour s'assurer que le transfert des alarmes se fasse dans un temps borné et « suffisamment » court, une fonction échelon $ku_T(t) = k_{\{t>T\}}$ est utilisée pour représenter une alarme avec notamment $k = 72$ octets. Pour obtenir les limites temporelles d'un envoi d'alarme, nous nous plaçons dans le pire des cas en supposant que toutes les alarmes peuvent être émises simultanément à $T = 0$. Aussi $b^0_{alarme}(t) = 72$.

Détermination des chemins et des priorités : dans les réseaux Ethernet commutés, les chemins sont fixes et déterminés par le protocole Spanning Tree. Ensuite l'affectation des priorités est la suivante :

- les messages prioritaires, c'est-à-dire avec des dates d'échéances strictes sont configurés en priorité haute. Cela concerne les trafics périodiques et les alarmes.
- les messages de maintenance ont une priorité faible.

Formulation des équations de rafales de sortie : sur chaque commutateur, nous posons les équations (4.3) d'avalanches en sortie des composants élémentaires. Afin de simplifier notre présentation, nous décrivons uniquement les calculs les plus significatifs associés à l'ordonnancement des trafics sur le commutateur fédérateur. Tout d'abord, nous devons considérer le trafic émis par le commutateur fédérateur à destination du commutateur B regroupant les communications entre l'automate et les cartes d'entrées/sorties et les communications entre le routeur et le navigateur. Cela signifie que le multiplexeur du commutateur fédérateur (composant 7 sur la figure 4.8(b)) doit retransmettre en prenant en compte les deux classes de service. Aussi l'union des équations établies pour un flux prioritaire et (4.3) donne pour le trafic de contrôle :

$$\sigma^{out}_{aut} = \sigma^{in}_{aut} + \rho^{in}_{aut}\frac{L_{web}}{C_{out}}$$

On voit que l'augmentation des rafales du trafic de contrôle ne dépend que de la longueur d'une trame « web ». Pour déterminer les rafales du trafic de maintenance, l'équation du délai de traversée d'un multiplexeur en best-effort est utilisée et donne :

$$\sigma^{out}_{web} = \sigma^{in}_{web} + 2\frac{\rho^{in}_{web}}{C_{out}}\left[\sigma^{in}_{aut} + \rho^{in}_{aut}\frac{\sigma^{in}_{web}}{C_{out} - \rho^{in}_{web}} + \rho^{in}_{aut}\frac{L_{aut}}{C_{out}}\right]$$

Le facteur 2 provient du fait que l'automate échange avec les deux cartes d'entrées connectées sur le commutateur B. Contrairement au trafic de contrôle, on s'aperçoit que le délai du trafic de maintenance dépend du nombre des communications de haute priorité.

Enfin, le trafic à destination du commutateur B rassemble les communications entre les cartes d'entrées/sorties connectées au commutateur B et l'automate qui sont de deux types : périodique et alarme. Ces communications sont de même priorité. Donc le retard n'est fonction que du temps nécessaire pour retransmettre

la trame la plus longue. Donc, la rafale de sortie s'exprime pour le flux périodique par :

$$\sigma^{out}_{cartes} = \sigma^{in}_{cartes} + \rho^{in}_{cartes}\frac{L_{cartes}}{C_{out}}$$

Une formalisation similaire est établie pour les alarmes.

Calcul des rafales de sortie : toutes les équations sur les rafales de sortie précédentes donnent un système d'équations défini par l'équation matricielle de l'étape 5 de la méthode de calcul. La résolution de ce système d'équations permet de déterminer les valeurs de rafale tout au long du réseau.
Le nombre d'inconnues dépend du nombre de communications ainsi que du nombre de composants élémentaires traversés par chacun de ces flux (soit encore du modèle de commutateur utilisé). Si l'on considère que le modèle n°3 est utilisé pour décrire un commutateur et qu'il fait apparaître deux inconnues supplémentaires (les valeurs de l'avalanche de données suite à la traversée du multiplexeur et du port de sortie), le nombre d'inconnues augmente de 2 par commutateur traversé. Le nombre de saut est alors défini ainsi :

- 1 saut de l'automate vers les cartes connectées au commutateur (2 flux)
- 3 de l'automate vers les cartes connectées au commutateur B (2 flux)
- 1 des cartes connectées au commutateur A vers l'automate (2 flux)
- 3 des cartes connectées au commutateur B vers l'automate (2 flux)
- 2 du routeur au PC (1 flux)
- 3 pour les alarmes (2 flux)

Dans la mesure où les alarmes ne correspondent qu'à une seule trame, elles n'apportent pas de volumes d'avalanches de données inconnues. On obtient alors 36 inconnues, auquel il convient d'ajouter les valeurs initiales σ^0, soit 45 inconnues pour ce simple exemple !

Calcul des délais de bout en bout : Dans cette dernière partie, l'augmentation des rafales tout au long du réseau est traduite en délai maximum en utilisant l'équation (4.3). Ainsi, avec $(h+1)$ le nombre de composants, le délai de bout en bout pour chaque flux i est obtenu à partir de :

$$\overline{D_i} = \frac{\sigma_i^h - \sigma_i^0}{\rho_i}$$

Les calculs pour le scénario de communication sur l'architecture réseau de la figure 5.11 sont présentés dans le tableau 5.2. Nous les comparons avec ceux obtenus par l'outil de simulation réseau *ComNet III*.

TAB. 5.2: Délais des communications de la figure 5.11 en ms

trafic	criticité	calcul réseau	comnet
trafic web	*best effort*	10,80	7,08
trafic de contrôle			
vers les cartes B		1,28	1,08
des cartes B	prioritaire	0,67	0,48
de/vers les cartes A		0,38	0,12
alarmes		0,60	0,46

Les résultats présentés dans la table 5.2 montrent les effets des priorités sur les messages puisque le trafic web dépasse dans le pire des cas 10 ms alors que les délais du trafic prioritaire ne dépassent pas les

$1,3\ ms$. Les performances du trafic prioritaire sont donc en adéquation avec le temps cycle automate de $10\ ms$. Nous constatons aussi que les résultats du simulateur sont toujours en déçà de nos calculs mais se comportent de la même manière. Le calcul réseau est évidemment pessimiste car il considère toujours le pire des cas, au niveau de chacun des commutateurs, ce que nous pouvons faire uniquement sur les stations sources sur Comnet.

Considérons maintenant le cas où la politique du multiplexeur de sortie du premier commutateur est de type Weighted Fair Queueing. Compte tenu des expressions développées au chapitre précédent, il est attendu que sous WFQ, le majorant du délai de bout en bout supporté par les trames du flux de contrôle sera plus grand que dans le cadre de la politique à priorité stricte. Il est toutefois possible de limiter cette augmentation de sorte que ce majorant reste inférieur aux besoins de l'application, c'est-à-dire au temps de cycle de l'automate. En effet, dans WFQ, chaque délai dépend des poids attribués aux niveaux de priorité. L'objectif sera alors de déterminer les valeurs des poids pour lesquels les délais du trafic web seront les plus faibles possibles, tout en veillant à ce que le trafic de contrôle respecte les valeurs prédéterminées par l'application.

5.4 Conclusion

La principale caractéristique de ce chapitre est qu'il donne un aperçu des possibilités offertes par l'évaluation de majorant des délais de traversée de bout en bout. Outre le fait que ces majorants peuvent être directement utilisés pour déterminer si le réseau Ethernet commuté va satisfaire les exigences de l'application distribuée, ils peuvent être utilisés au dimensionnement du réseau.

L'intérêt de la méthode du calcul réseau est alors multiple. Pour une configuration donnée, le calcul réseau permet d'évaluer les délais de bout en bout et de déterminer ainsi si les contraintes temporelles seront satisfaites ou non (en d'autre termes, il permet de rejeter ou non la configuration). Si cette configuration apparait comme respectueuse des contraintes temporelles, le calcul réseau permet également d'identifier les ressources nécessaires (taille des buffers dans les cartes réseaux et commutateurs par exemple). Si la configuration apparait non valide, il peut alors être utilisé pour évaluer d'autres configurations et identifier ainsi la meilleure. Dans ce cas, il est joint à une heuristique d'optimisation et sert de point de comparaison pour le type de topologie, la topologie logique, l'interconnexion des équipements, les priorités, les ressources nécessaires.... Enfin, il permet d'évaluer le degré d'évolutivité de l'architecture, comme la capacité d'accueil de nouveaux nœuds et communications.

Dans des architectures distribuées temps-réel qui intègrent de plus en plus de trafic supplémentaire de maintenance ou même directement d'entreprise (trafic vertical issu des ERP et MES), les possibilités mises en avant dans ce chapitre d'évaluation de la quantité maximale de ce type de données admissibles sont relativement intéressantes. Elles permettent notamment d'identifier l'enveloppe (σ, ρ) qui sera utilisée par les régulateurs de trafic pour limiter l'arrivée des données non utiles à la conduite du système. De la même façon, la possibilité de prendre en compte la classification de service et d'évaluer l'impact des paramètres associés, fournit un moyen de contrôler les impacts de ce nouveau trafic. Enfin, comme le montrent les deux premiers paragraphes, cette évaluation *a priori* du réseau peut être utilisée dans le cadre de la conception du réseau. Ainsi, si dans le premier exemple, cette évaluation sert simplement à identifier la meilleure architecture commutée, elle peut participer dans un second temps à l'optimisation de la distribution des nœuds terminaux comme le montre le second cas d'application. Dans ce cas, la méthode

proposée dans cette thèse est utilisée comme fonction d'évaluation au sein de techniques d'optimisation.

Conclusions

Les travaux abordés dans cette thèse présentent la particularité de se situer à la rencontre des deux domaines des sciences de l'information que sont l'automatique (ou plus précisément le contrôle / commande) et les réseaux. Plus précisément, le contexte de cette étude porte sur les systèmes contrôlés en réseau qui correspondent à des applications distribuées temps-réel. Pour le contrôle de l'application, la problématique revient à prendre en compte les paramètres intrinsèques du réseau (retard, pertes ...) et leurs effets implicites sur la performance de la commande. Du point de vue du réseau, l'enjeu que nous abordons ici est l'utilisation du réseau Ethernet pour ce type d'applications à contraintes temps-réel. Bien que ce réseau est aujourd'hui le plus utilisé dans les réseaux d'entreprise, il n'intègre pas de base aucun mécanisme adapté à une telle criticité temporelle. C'est ainsi que le premier chapitre a montré que si Ethernet est de plus en plus retenu, aucune étude ne permettait d'avoir une connaissance analytique de type déterministe de son comportement temporel alors que les premiers résultats obtenus dans le cadre des systèmes contrôlés en réseau insistent sur la nécessité d'être capable de borner le comportement du réseau, et plus particulièrement d'avoir des bornes sur les délais d'acheminement des messages.

Au deuxième chapitre, l'état de l'art des solutions visant à améliorer les performances d'Ethernet, voire à lui conférer un certain déterminisme n'a pas permis d'identifier une solution qui serait capable de fournir le déterminisme nécessaire aux études de stabilité de la commande sans pour autant remettre en cause les qualités d'Ethernet à l'origine de son succès (simplicité, faible coût, standardisation effective, flexibilité). C'est sur ce constat que nous nous sommes intéressés à une évaluation des performances de l'Ethernet standardisé. Dans ce cadre, ces travaux se sont concentrés sur les architectures commutées full-segmentées définies dans les standards successifs (802.1D, 802.1p, 802.1Q). Au sein du laboratoire, cette étude d'évaluation de performances des réseaux Ethernet commutés intervient après d'autres travaux concernant notamment l'optimisation des plans de câblage réseau Ethernet commuté.

L'évaluation de performances présentée dans les chapitres 3 & 4 repose sur la théorie du calcul réseau qui a initialement été développée pour les architectures à Qualité de Service. Cette théorie permet notamment le calcul de majorants déterministes caractérisant les systèmes à file d'attente comme les délais ou les arriérés de traitement. Elle fait intervenir différentes notions comme la courbe d'arrivée et la courbe de service en lieu et place des traditionnels modèles de trafic et de système. Nos travaux ne sont pas les seuls abordant l'utilisation de cette théorie aux réseaux Ethernet commutés. Toutefois, ils présentent une double originalité. Dans un premier temps, nous avons proposé un modèle de commutateur issu d'une analyse fonctionnelle détaillée de la commutation. De ce modèle, l'utilisation de la théorie du calcul réseau nous a alors permis de développer différents majorants de traversée de composants élémentaires sous l'hypothèse que le trafic soit limité par une contrainte d'avalanche maximale des données. La seconde originalité de notre approche présentée au chapitre 4 repose sur la conservation des travaux originaux

de Cruz qui conduisent à évaluer l'augmentation du délai au fur et à mesure de la traversée du réseau à partir de l'augmentation de l'avalanche maximale de données. L'ensemble de la méthodologie de calcul de majorants des délais de bout en bout a donné lieu à un algorithme à 7 étapes. Différentes expérimentations réelles ont alors permis de constater la validité des résultats obtenus par cette méthode. Finalement, l'intérêt de ces travaux est double : d'une part, ils contribuent à estimer l'applicabilité des réseaux Ethernet commutés comme support d'applications distribuées et d'autre part, ils fournissent aux études de commandabilité et de stabilité des systèmes contrôlés en réseau une information déterministe sur l'évolution des délais.

La portée de ces travaux de thèse a alors été étendue aux réseaux Ethernet commutés implémentant la Classification de Service définie dans la norme et divers cas d'utilisation de cette méthode ont été présentés au chapitre 5. Dans ce chapitre, nous avons pu voir que les résultats de ces travaux peuvent tout aussi bien être utilisé comme aide au dimensionnement et à l'optimisation de l'architecture du réseau.

Perspectives

L'étude menée tout au long de cette thèse ainsi que les résultats obtenus permettent alors de dégager plusieurs perspectives de recherche.

Dans les travaux présentés, le calcul réseau est utilisé pour évaluer les performances temporelles du réseau et fournir ainsi au contrôle / commande une information utile à des études de stabilité du contrôle : les pires délais de bout en bout. Dans le cadre du projet européen NeCST, nous avons commencé à nous intéresser à étudier les effets du réseau pour le diagnostic. La plateforme développée dans ce cadre consiste à la simulation des contrôleurs et du process *via* des outils comme Matlab ou de modélisation à bases de systèmes à événements discrets sur des machines distinctes. Ces stations sont ensuite interconnectées *via* un réseau Ethernet réel dans lequel on place un routeur logiciel qui nous permet de jouer / émuler différents niveaux de délai et de perte. L'objectif à terme est de pouvoir corréler les niveaux de performance du réseau et avec les niveaux de qualité de service exigés par la commande ou encore le diagnostic des applications distribuées. Dans ce cadre, il serait également envisageable que l'évaluation de performances présenté dans cette thèse intègre également la modélisation des nœuds terminaux comme le process et les stations. L'utilisation de techniques de représentation formelles comme les réseaux de Petri pourrait être aussi utilisée dans la mesure où les propriétés mises en avant dans cette formalisation peuvent être transposées dans la théorie Min-Plus (Boimond, 1999).

Jusqu'ici les travaux présentés se sont intéressés au réseau Ethernet commuté filaire. Si il est vrai que ces réseaux bénéficient aujourd'hui d'un grand attrait, la technologie sans-fil IEEE 802.11 présente quant à elle des qualités complémentaires comme la mobilité ou la simplification et la flexibilité du câblage. Cet engouement se caractérise notamment par des versions sans-fil des réseaux de terrain traditionnels, par la parution de nouveaux périodiques comme *The Industrial Wireless book* mais aussi par des premiers travaux de recherche basés sur ce type de réseau (Colandairaj *et al.*, 2005). Par rapport à l'Ethernet commuté, la principale différence est liée au mécanisme d'accès CSMA/CA ainsi que la nécessité d'acquittement des messages. Bien que les réseaux sans-fil 802.11 n'offrent pas plus de déterminisme, nous pensons néanmoins qu'une modélisation des points d'accès basée sur une analyse fonctionnelle des étapes clés doit nous permettre de développer une analyse d'évaluation de performances similaire. Cette évaluation serait

d'autant plus intéressante dans le cas d'un réseau mêlant sous-réseau sans-fil et sous-réseau Ethernet commuté. Cela nous permettrait également d'étudier l'impact de chaque technologie. De la même façon que pour Ethernet commuté où le calcul réseau peut être utilisé comme évaluation de la distribution des nœuds terminaux sur les commutateurs, il pourrait être utilisé pour l'optimisation de la charge des points d'accès qui est fonction de la distribution des acteurs sur ces bornes.

L'application au contexte des réseaux sans-fils 802.11 reste toutefois un problème ouvert compte tenu de l'indéterminisme de ce protocole : collisions, problème des stations cachées ... De plus, le medium utilisé est beaucoup plus sujet aux perturbations physiques liés à l'environnement extérieur. Cela nécessitera par conséquent d'intégrer dans la théorie utilisée la notion de pertes (de nature stochastique !). L'étude pourra alors se baser sur les travaux développés dans (Chang, 2000) et dans (Vojnovic et Boudec, 2002) qui présentent tout deux un calcul réseau stochastique pour étudier les performances du réseau dans le cas d'environnements à probabilités de pertes non nulles. Ces derniers travaux ont ainsi déjà été abordés dans le cadre des réseaux Ethernet commutés embarqués dans les avions (Grieu, 2004).
Par ailleurs, la question de l'optimisation des paramètres inhérents à la Classification de Service devra être traitée. (Grieu, 2004) présente une optimisation des niveaux de priorités affectés aux communications pour différentes politiques d'ordonnancement en se basant sur différentes heuristiques comme les algorithmes génétiques multicritères. Nous pensons qu'il serait ainsi intéressant de proposer une méthode d'optimisation de la mise en œuvre de la classification de service dans le cadre des applications distribuées temps-réel. Le cas envisagé fait intervenir 3 niveaux de priorité et utilise à la fois les politiques d'ordonnancement à priorité stricte et WFQ.

Les travaux liés à la métrologie réseau devront également être prolongés dans le contexte des systèmes contrôlés en réseau. Dans cette thèse, nous avons mis en évidence l'importance cruciale de connaître les délais d'acheminement pour la commande du système distribué. Les points à développer touchent alors la généralisation des mesures sur un réseau en fonctionnement, le degré de précision des mesures, la prise en compte des incertitudes de mesure ainsi que l'extension des architectures étudiées (Aitali *et al.*, 2005). Le dernier point fait référence à la nécessité d'entreprendre des expérimentations complémentaires pour valider le modèle à Classification de Service. Pour cela, l'architecture mesurée devra intégrer des commutateurs implémentant les priorités et la synchronisation d'horloge pour faciliter les mesures.

La poursuite de nos travaux la plus proche concerne l'implémentation des modèles de commutateur (définis pour le calcul réseau) pour vérifier le bon fonctionnement de ceux-ci en utilisant le concept de filtre de comportement. La thèse de Belynda Brahimi étudie ainsi les possibilités pour embarquer différents modèles dans les commutateurs, les premiers basés sur le calcul réseau pour détecter les fautes et les seconds pour les isoler en utilisant par exemple les réseaux de Petri ou les automates temporisés.

* * *

Majorants du Network Calculus (Le Boudec et Thiran, 2001)

L'objectif de cette annexe est de fournir les démonstrations des théorèmes à la base du calcul réseau. Les pages suivantes sont extraites de (Le Boudec et Thiran, 2001).

Theorem 1.5.1 (Backlog Bound). *Assume a flow, constrained by arrival curve α, traverses a system that offers a service curve β. The backlog $R(t) - R^*(t)$ for all t satisfies :*

$$R(t) - R^*(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\}$$

Proof : The proof is a straightforward application of the definitions of service curve and arrival curves :

$$R(t) - R^*(t) \leq R(t) - \inf_{0 \leq s \leq t} [R(t-s) + \beta(s)]$$

Thus

$$R(t) - R^*(t) \leq \sup_{0 \leq s \leq t} [R(t) - R(t-s) + \beta(s)] \leq \sup_{0 \leq s \leq t} [\alpha(s) + \beta(t-s)]$$

□

Call $\delta(s)$ the virtual delay for a hypothetical system that would have α as input and β as output, assuming that such a system exists (in other words, assuming that $(\alpha \leq \beta)$).

$$\delta(s) = \inf \{\tau \geq 0 : \alpha(s) \leq \beta(s + \tau)\}$$

Then, $h(\alpha, \beta)$ is the supremum of all values of $\delta(s)$.

Theorem 1.5.2 (Delay Bound). *Assume a flow, constrained by arrival curve α, traverses a system that offers a service curve β. The virtual delay $d(t)$ for all t satisfies : $d(t) \leq h(\alpha, \beta)$.*

Proof : Consider some fixed $t \geq 0$; for all $\tau < d(t)$, we have, from the definition of virtual delay, $R(t) > R^*(t + \tau)$. Now the service curve property at time $t + \tau$ implies that there is some s_0 such that

$$R(t) > R(t + \tau - s_0) + \beta(s_0)$$

It follows from this latter equation that $t + \tau - s_0 < t$. Thus

$$\alpha(\tau - s_0) \geq [R(t) - R(t + \tau - s_0)] > \beta(s_0)$$

Thus $\tau \leq \delta(\tau - s_0) \leq h(\alpha, \beta)$. This is true for all $\tau < d(t)$ thus $d(t) \leq h(\alpha, \beta)$. □

Theorem 1.5.3 (Output Flow). *Assume a flow, constrained by arrival curve α, traverses a system that offers a service curve β. The output flow is constrained by the arrival curve $\alpha^* = \alpha \oslash \beta$.*

Proof : With the same notation as above, consider $R^*(t) - R^*(t - s)$, for $0 \leq t - s \leq t$. Consider the definition of the service curve, applied at time $t - s$. Assume for a second that the inf in the definition of $R \otimes \beta$ is a min, that is to say, there is some $u \geq 0$ such taht $0 \leq t - s - u$ and

$$(R \otimes \beta)(t - s) = R(t - s - u) + \beta(u)$$

Thus

$$R^*(t - s) - R(t - s - u) \geq \beta(u)$$

and thus

$$R^*(t) - R^*(t - s) \leq R^*(t) - \beta(u) - R(t - s - u)$$

Now $R^*(t) \leq R(t)$, therefore

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

and the latter term is bounded by $(\alpha \oslash \beta)(s)$ by the definition of the $\oslash$ operator.

Now relax the assumption that the inf in the definition of $R \otimes \beta$ is a min. In this case, the proof is essentially the same with a minor complication. For all $\epsilon > 0$ there is some $u \geq 0$ such that $0 \leq t - s - u$ and

$$(R \otimes \beta)(t - s) \geq R(t - s - u) + \beta(u) - \epsilon$$

and the proof continues along the same line, leading to :

$$R^*(t) - R^*(t - s) \leq (\alpha \oslash \beta)(s) + \epsilon$$

This is true for all $\epsilon > 0$, which proves the result. □

Liste des publications

Revues internationales avec comité de lecture

J.P. Georges, T. Divoux, E. Rondeau, *Confronting the performances of a switched Ethernet network with industrial constraints by using the Network Calculus. Accepté, en cours de publication* IJCS, International Journal of Communication Systems, John Wiley & Sons, `http://www3.interscience.wiley.com/cgi-bin/abstract/110570793/ABSTRACT`.

J.P. Georges, N. Krommenacker, T. Divoux, E. Rondeau, *Designing suitable switched Ethernet architectures reagarding real-time application constraints. Accepté, en cours de publication* EAAI, Engineering Applications of Artifical Intelligence, Elsevier, IFAC.

J.P. Georges, E. Rondeau, T. Divoux, *Evaluation de performances d'architectures Ethernet commuté.* RSTI - Technique et Science Informatique, Revue thématique "Temps Réel", Hermès Lavoisier, volume 22, numéro 5, pages 621-649, 2003.

Conférences internationales avec comité de lecture

J.P. Georges*, T. Divoux, E. Rondeau, *Validation of the Network Calculus approach for the performance evaluation of switched Ethernet based on industrial communications.* WORLD IFAC'05, 16ème IFAC World congress. juillet 2005.

J.P. Georges, T. Divoux, E. Rondeau, *Strict priority versus Weighted Fair Queueing in switched Ethernet networks for time-critical applications.* WPDRTS'05, Workshop on Parallel and Distributed Real-Time Systems, Denver, Colorado. avril 2005.

J.P. Georges, T. Divoux, E. Rondeau, *A formal method to guarantee a deterministic behaviour of switched Ethernet networks for time-critical applications.* CACSD'04, IEEE International Symposium on Computer Aided Control Systems Design, Taipei, Taiwan, septembre 2004.

J.P. Georges, N. Krommenacker, T. Divoux, E. Rondeau, *Designing suitable switched Ethernet architectures regarding real-time application constraints.* INCOM'04, 11th IFAC Symposium on Information Control Problems in Manufacturing, Salvador Bahia, Brésil, avril 2004.

J.P. Georges*, T. Divoux, E. Rondeau, *Comparison of switched Ethernet architectures models.* ETFA'03, 9th IEEE International Conference on Emerging Technologies and Factory Automation, Lisbonne, Portugal, pages 375-382, vol. 1, septembre 2003.

J.P. Georges, E. Rondeau, T. Divoux, *Evaluation of switched Ethernet in an industrial context by*

using the Network Calculus. WFCS'02, 4th IEEE International Workshop on Factory Communication Systems, Västerås, Suède, pages 19-26, août 2002.

J.P. Georges*, E. Rondeau, T. Divoux, *How to be sure that Ethernet networks will satisfy the real time requirements ?*. ISIE'02, IEEE International Symposium on Industrial Electronics, L'Aquila, Italie, juillet 2002.

N. Krommenacker, J.P. Georges, E. Rondeau, T. Divoux, *Designing, modelling and evaluating switched Ethernet networks in factory communication systems*. RTLIA'02, 14th IEEE Euromicro Conference on Real Time Systems, 1st Worshop on Real Time LAN's in the Internet Age, Vienne, Austriche, juin 2002.

Conférences nationales avec comité de lecture

J.P. Georges*, T. Divoux, E. Rondeau, *Une méthode formelle pour garantir le comportement déterministe de réseaux Ethernet commutés pour des applications temps-réel*. CIFA'04, IEEE Conférence Internationale Francophone d'Automatique, Douz, Tunise, novembre 2004.

J.P. Georges*, E. Rondeau, T. Divoux, *Evaluation d'architectures Ethernet commuté par calcul et simulation*. MSR'03, 4ème Colloque Francophone sur la Modélisation des Systèmes Réactifs, Metz, France, octobre 2003.

J.P. Georges*, E. Rondeau, T. Divoux, *Evaluation des performances de réseaux Ethernet industriels : étude et comparaison de différents modèles*. JDA'03, "Journées Doctorales d'Automatique", Valenciennes, France, juin 2003.

Groupes de travail nationaux

J.P. Georges, *Evaluation de l'Ethernet Commuté dans un Contexte Industriel par le Calcul Réseau*, réunion AS01, Nantes, 2002.

J.P. Georges, *Evaluation d'Ethernet Commuté dans un Contexte Industriel par le Calcul Réseau*, groupe de travail Réseaux Grand Est, *RGE*, Nancy, juin 2002.

Rapports de contrats

C. Aubrun, E. Rondeau, J.P. Georges, N. Krommenacker, V. Lecuire, B. Brahimi, F. Michaut, *Deliverable 5-1 : Simulation of optimized network architecture*. Projet STREP NeCST, mai 2005.

J.P. Georges*, E. Rondeau, C. Aubrun, *A short tutorial to distribute a Matlab / Simulink model on the network*. Rapport technique du projet européen de type STREP intitulé NeCST, mars 2005.

E. Rondeau, J.P. Georges. *Rapport final sur le projet « Intégration des mémoires embarquées et d'une caméra télépilotable dans une application de commerce électronique »*. Convention de sous-traitance entre l'ILGU et le CRAN sur le contrat Européen : Programme TELEMATICS projet MEDIASITE IA 1012UR, janvier 2002.

Bibliographie

Adoud, H. (2002). Méthodes de partitionnement de graphes pour la conception d'architectures de réseaux locaux. PhD thesis. Université Henri Poincaré Nancy 1, Centre de Recherche en Automatique de Nancy.

Adoud, H., E. Rondeau et T. Divoux (2001). Configuration of communication networks by analysing co-operation graphs. *Computer Communications*, **24**(15-16), 1568–1577.

AFNOR (1990). Bus FIP pour échange d'informations entre transmetteurs, actionneurs et automates - Couche liaison de données. NF C46-603.

Aitali, A., F. Michaut et F. Lepage (2005). Analyse et étude comparative des techniques et outils de mesure de la bande passante disponible. In : *11ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'05)*. Bordeaux.

Almeida, L., P. Pedreiras et J.A. Fonseca (2002). FTT-CAN : why and how. *IEEE Transactions on Industrial Electronics*, **49**(6), 1189–1201.

Alves, M., E. Tovar et F. Vasques (2000). Ethernet goes real-time : a survey on research and technological developments. Technical Report HURRAY-TR-2K01. Groupe de recherche IPP-HURRAY, Polytechnic Institute of Porto (ISEP-IPP).

Aubrun, C., E. Rondeau, J.-P. Georges, N. Krommenacker, V. Lecuire, B. Brahimi et F. Michaut (2005). Deliverable 5-1 : Simulation of optimized network architecture. Projet STREP NeCST.

Baccelli, F. et D. Hong (2000). TCP is max-plus linear : and what tells us on its throughput. In : *ACM SIGCOMM Computer Communication Review*. Vol. 30. Stockholm, Suède. pp. 219–230.

Bennett, J.C. et H. Zhang (1997). Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking*, **5**(5), 675–689.

Boimond, J.-L. (1999). Sur l'étude des systèmes à événements discrets dans l'algèbre des dioïdes : identification, commande des graphes d'événements temporisés, représentation des graphes d'événements temporisés à paramètres variables. Habilitation à diriger des recherches, Université d'Angers, LISA.

Branicky, M.S., V. Liberatore et S.M. Phillips (2003). Networked control system co-simulation for co-design. In : *American Control Conference*. Vol. 4. Denver, Etats Unis. pp. 3341–3346.

Brooks, P. (2001). EtherNet/IP : Industrial protocol white paper. In : *8th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'01)*. Vol. 2. Rockwell Automation. Antibes - Juan lès Pins, France. pp. 505–514.

CACI Products Company (1998). Comnet III reference guide. Release 2.0.

Caponetto, R., L. Lo Bello et O. Mirabella (2002). Fuzzy traffic smoothing : Another step towards statistical real-time communication over Ethernet networks. In : *1st International Workshop on Real-Time LANs in the Internet Age (RTLIA'02)*. Vienne, Autriche. pp. 45–48.

CENELEC (1996). Fieldbus ; vol.1 P-net, vol.2 Profibus, vol. 3 WorlFip. European Standard EN50170.

Cervin, A., D. Henriksson, B. Lincoln, J. Eker et K.-E. Arzen (2003). How does control timing affect performance ?. *IEEE Control Systems magazine*, **23**(3), 16–30.

Chang, C.S (1997). A filtering theory for deterministic traffic regulation. In : *IEEE Conference on Computer Communications (INFOCOM'97)*. Kobe, Japon. pp. 436–443.

Chang, C.S. (2000). *Performance Guarantees in Communication Networks*. Telecommunication Networks and Computer Systems. Springer Verlag.

Chang, C.S., R. Cruz, J.-Y. Le Boudec et P. Thiran (2002). A min-plus system theory for constrained traffic regulation and dynamic service guarantees. In : *IEEE/ACM Transactions on Networking*. Vol. 10. pp. 805–817.

Chiueh, T. et Venkatramani C. (1994). Supporting real-time traffic on Ethernet. In : *15th IEEE Real-Time Systems Symposium (RTSS'94)*. San Juan, Puerto Rico. pp. 282–286.

Cohen, G., S. Gaubert, R. Nikoukhah et J.-P. Quadrat (1991). A linear system theory for systems subject to synchonization and saturation constraints. In : *1st European Control Conference*. Grenoble.

Colandairaj, J., G.W. Irwin et W.G. Scanlon (2005). Analysis and co-simulation of an IEEE 802.1b wireless networked control system. In : *16th IFAC World Congress*. Prague, République Tchèque.

Cruz, R. (1991*a*). A calculus for network delay, part I : network elements in isolation. *IEEE Transactions on Information Theory*, **37**(1), 114–141.

Cruz, R. (1991*b*). A calculus for network delay, part II : Network analysis. *IEEE Transactions on Information Theory*, **37**(1), 132–141.

Cruz, R. (1995). Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications*, **13**(6), 1048–1056.

Cruz, R.L. et C.M. Okino (1996). Service guarantees for window flow control. In : *34th Allerton Conference on Communication, Control, & Computing*. Monticello, Etats-Unis.

Daoudi, C., F. Michaut et H. Panetto (2001). A dioids algebra model of congestion control of ABR traffic on ATM networks. In : *IAR*. Strasbourg.

Darouach, M. (2001). Linear functional observers for systems with delays in State variables. *IEEE Transactions on Automatic Control*, **46**(3), 491–496.

Decotignie, J.D. (2001). A perspective on Ethernet-TCP/IP as a fielbus. In : *4th IFAC International Conference on Fieldbus Systems and their Applications (FET'2001)*. Nancy, France. pp. 138–143.

Demers, A., S. Keshav et S. Shenker (1989). Analysis and simulation of a fair queueing algorithm. In : *ACM SIGCOMM Computer Communication Review*. Vol. 19. pp. 1–12.

Digital Equipment, Intel et Xerox (1980). The Ethernet : a local area network - data link layer and physical layer specifications. Technical report. Digital Equipment Corporation and Intel Corporation and Xerox Corporation. version 1.0.

Dolejš, Ondřej, Petr Smolík et Zdeněk Hanzálek (2004). On the Ethernet use for real-time publish-subscribe based applications. In : *IEEE International Workshop on Factory Communication Systems (WFCS'04)*. Vienne, Autriche. pp. 39–44.

ESPRIT consortium CCE-CNMA (1995). CCE : an integration platform for distributed manufacturing applications. Technical report. Springer Verlag.

Gaderer, G., P. Loschmidt et T. Sauter (2005). IEEE 1588 Real-Time Networks with Hybrid Master Group Enhancements. In : *4th International Workshop on Real-Time Networks (RTN'05)*. Palma de Mallorca, Espagne.

Gallager, R.G. (1978). Conflict resolution in random access broadcast networks. In : *AFOSR Workshop in Communication Theory and Applications*. Provincetown. pp. 74–76.

Gallara, D. et J.-D. Decotignie (1984). Groupe de réflexion FIP : proposition d'un système de transmission série multiplexée pour les échanges entre des capteurs, des actionneurs et des automates réflexes. Livre blanc, Ministère de l'Industrie.

Georges, J.-P., E. Rondeau et T. Divoux (2003*a*). Evaluation de performances d'architectures Ethernet commuté. *RSTI - Technique et Science Informatique, Revue thématique "Temps Réel"*, **22**(5), 621–649.

Georges, J.-P., T. Divoux et E. Rondeau (2003*b*). Comparison of switched Ethernet architectures models. In : *IEEE Conference on Emerging Technologies and Factory Automation (ETFA'03)*. Vol. 1. Lisbone, Portugal. pp. 375–382.

Georges, J.-P., T. Divoux et E. Rondeau (2004*a*). Une méthode formelle pour garantir le comportement déterministe de réseaux Ethernet commutés pour des applications temps-réel. In : *Conférence Internationale Francophone d'Automatique (CIFA'04)*. Douz, Tunisie.

Georges, J.-P., T. Divoux et E. Rondeau (2005*a*). Confronting the performances of a switched Ethernet network with industrial constraints by using the Network Calculus. *International Journal of Communication Systems (IJCS)*. Accepté, en cours de publication.

Georges, J.-P., T. Divoux et E. Rondeau (2005*b*). Strict priority versus Weighted Fair Queueing in switched Ethernet networks for time-critical applications. In : *Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'O5 - IPDPS'05)*. Denver, Colorado.

Georges, J-P., T. Divoux et E. Rondeau (2005*c*). Validation of the Network Calculus approach for the performance evaluation of switched Ethernet based on industrial communications. In : *16ème IFAC World congress*. Prague, République Tchèque.

Georges, J.P., E. Rondeau et T. Divoux (2002). Evaluation of switched Ethernet in an industrial context by using the network calculus. In : *4th IEEE International Workshop on Factory Communication Systems (WFCS'02)*. Västerås, Suède. pp. 19–26.

Georges, J.P., E. Rondeau et T. Divoux (2003*c*). Evaluation des performances de réseaux Ethernet industriels : étude et comparaison de différents modèles. In : *JDA'03 "Journées Doctorales d'Automatique"*. Valenciennes.

Georges, J.P., N. Krommenacker, T. Divoux et E. Rondeau (2004*b*). Designing suitable switched Ethernet architectures regarding real-time application constraints. In : *11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'04)*. Salvador Bahia, Brésil.

Georges, J.P., N. Krommenacker, T. Divoux et E. Rondeau (2005*d*). A design process of switched Ethernet architectures according to real-time application constraints. *Engineering Applications of Artificial Intelligence*. Accepté, en cours de publication.

Georges, JP., T. Divoux et E. Rondeau (2004*c*). A formal method to guarantee a deterministic behaviour of switched Ethernet networks for time-critical applications. In : *IEEE Conference on Computer Aided Control Systems Design (CACSD'04)*. Taipei, Taiwan.

Grieu, J. (2004). Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques. PhD thesis. Institut National Polytechnique de Toulouse, Ecole doctorale informatique et télécommunications.

Grieu, J., F. Frances et C. Fraboul (2003). Preuve de déterminisme d'un réseau embarqué avionique. In : *10ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'03)*. Hermes Lavoisier. Paris. pp. 287–303.

Hanssen, F. et P.G. Jansen (2003). Real-time communication protocols : an overview. Technical Report TR-CTIT-03-49. Centre for Telematics and Information Technology, University of Twente, Enschede.

Hartman, J.R. (2004). Networked control system co-simulation for co-design : theory and experiments. PhD thesis. Department of Electrical Engineering and Computer Science, Case Western Reserve University.

Hirschmann GmBH (n.d.). Highlights : automation and networking solutions. Technical report. Edition 10/99.

Hoang, H. (2004). Real-time communication for industrial embedded systems using switched Ethernet. In : *12th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'04 - IPDPS'04)*. Santa Fe, Nouveau Mexique.

Hoang, H. et M. Jonsson (2003). Switched real-time Ethernet in industrial applications - asymmetric deadline partitioning scheme. In : *2nd International Workshop on Real-Time LANs in the Internet Age (RTLIA'03)*. Porto, Portugal.

Hoang, H., M. Jonsson, A. Larsson, R. Olsson et C. Bergenhem (2002). Deadline first scheduling in switched real-time Ethernet - Deadline partitioning issues and software implementation experiments. In : *1st International Workshop on Real-Time LANs in the Internet Age (RTLIA'02)*. Vienne, Autriche. pp. 68–71.

Höller, R., T. Sauter et N. Kerö (2003). Embedded SynUTC and IEEE 1588 clock synchronization for industrial Ethernet. In : *9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'03)*. Vol. 1. Lisbonne, Portugal. pp. 422–426.

IEEE (1998). IEEE standards for information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 3 : Media Access Control (MAC) bridges. ANSI/IEEE Std 802.1D, Edition 1998.

IEEE (2002). IEEE standard for a precision clock synchronization protocol for networked measurement and control systems. ANSI/IEEE standard 1588-2002.

IEEE Computer Society (2002). IEEE standard for information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific require-

ments - Part 3 : Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE standard 802.3, Edition 2002.

IEEE Computer Society (2003). IEEE Standards for local and metropolitan area networks - Virtual bridged local area networks. IEEE standard 802.1Q, Edition 2003.

IEEE, The Open Group Base Specifications (2004). Standard 1003.1, 2004 Edition.

IETF Intserv working group (1997). Network Element Service Specification Template. Request For Comments RFC 2216.

ISO (1989). Information processing systems - Fibre Distributed Data Interface (FDDI) - Part 2 : Token Ring Media Access Control (MAC). ISO international standard 9314-2.

ISO (1993). Road vehicle - Interchange of digital information - Controller Area Network (CAN) for high-speed Communication. ISO 11898.

Iung, B. (2002). Contribution à l'Automatisation des Systèmes Intelligents de Production :Interopérabilité des Processus de Contrôle, Maintenance et Gestion Technique. Habilitation à Diriger des Recherches, Université Henri Poincaré Nancy 1, CRAN.

Jasperneite, J. et P. Neumann (2001). Performance evaluation of switched Ethernet in real-time applications. In : *4th IFAC International conference on Fieldbus Systems and their Applications (FET'01)*. Nancy, France. pp. 144–151.

Jasperneite, J., K. Shehab et K. Weber (2004). Enhancements to the time synchronization standard IEEE-1588 for a system of cascaded bridges. In : *5th IEEE International Workshop on Factory Communication Systems (WFCS'04)*. Vienne, Autriche. pp. 39–44.

Jasperneite, J., P. Neumann, M. Theis et K. Watson (2002). Deterministic real-time communication with switched Ethernet. In : *4th IEEE International Workshop on Factory Communication Systems (WFCS'02)*. Västerås, Suède. pp. 11–18.

Juanole, G. (2005). Systèmes de Contrôle - Commande et Informatique Distribuée Temps Réel. Journée du groupe de travail STRQDS du GDR ARP sur le thème Networked Control Systems.

Juanole, G. et I. Blum (1999). Influence de fonctions de base (communication-ordonnancement) des systèmes distribués temps-réel sur les performances des applications de contrôle-commande. In : *7ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'99)*. Nancy, France. pp. 217–232.

Kamen, E.W., P. Torab, K. Cooper et G. Custodi (1999). Design and analysis of packet-switched networks for control systems. In : *38th IEEE Conference on Decision and Control (CDC'99)*. Vol. 5. Phoenix, Etats-Unis. pp. 4460–4465.

Kermani, P. et L. Kleinrock (1979). Virtual cut-through : A new computer communication switching technique. *Computer networks*, **3**, 267–286.

Krákora, J. et Z. Hanzalek (2004). Timed automata approach to CAN verification. In : *11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'04)*. Salvador Bahia, Brésil.

Krákora, J., L. Waszniowski, P. Písa et Z. Hanzálek (2004). Timed Automata Approach to Real Time Distributed System Verification. In : *5th IEEE International Workshop on Factory Communication Systems (WFCS'04)*. Vienne, Autriche. pp. 407–410.

Krommenacker, N. (2002). Heuristiques de conception de topologies réseaux : application aux réseaux locaux industriels. PhD thesis. Université Henri Poincaré, Nancy 1, Centre de Recherche en Automatique de Nancy.

Krommenacker, N., E. Rondeau et T. Divoux (2001). Study of algorithms to define the cabling plan of switched Ethernet for real-time applications. In : *8th IEEE International conference on Emerging Technologies and Factory Automation (ETFA'01)*. Antibes - Juan lès Pins, France. pp. 223–230.

Krommenacker, N., E. Rondeau et T. Divoux (2002*a*). Genetic algorithms for industrial Ethernet network design. In : *4th IEEE International Workshop on Factory Communication Systems (WFCS'02)*. Västerås, Suède. pp. 149–155.

Krommenacker, N., J.P. Georges, E. Rondeau et T. Divoux (2002*b*). Designing, modelling and evaluating switched Ethernet networks in factory communications systems. In : *1st International Workshop on Real-Time LANs in the Internet Age (RTLIA'02)*. Vienne, Autriche. pp. 72–75. RTLIA-ECTRS'02.

Kweon, S.-K., K.G. Shin et Q. Zheng (1999). Statistical real-time communication over Ethernet for manufacturing automation systems. In : *5th IEEE Real-Time Technology and Applications Symposium (RTAS'99)*. Vancouver, Canada. pp. 192–202.

Kweon, S.K., K.G. Shin et G. Workman (2000). Achieving real-time communication over Ethernet with adaptive traffic smoothing. In : *6th IEEE Real-Time Technology and Applications Symposium (RTAS'00)*. Washington DC. pp. 90–100.

Le Boudec, J.-Y. et P. Thiran (2001). *Network calculus, a theory of deterministic queueing systems for the Internet*. Lecture Notes in Computer Science. Springer Verlag.

Le Boudec, J.Y. (1992). The Asynchronous Transfer Mode : A Tutorial. *Computer Networks and ISDN Systems*.

Le Boudec, J.Y. (1996). Network calculus made easy. Technical Report EPFL-DI 96/218. Ecole Polytechnique Fédérale de Lausanne (EPFL).

Le Boudec, J.Y. (1998). Application of Network Calculus to guarantee service networks. *IEEE Transactions on Information Theory*, **44**, 1087–1096.

Le Boudec, J.Y. et P. Thiran (1998). Network Calculus viewed as a min-plus system theory applied to communication networks. Technical Report SSC/1998/016. Ecole Polytechnique Fédérale de Lausanne (EPFL).

Le Lann, G. et N. Rivierre (1993). Real-time communications over broadcast networks : The CSMA-DCR and DOD-CSMA-CD protocols. Technical Report 1863. INRIA.

Lee, K. (1995). Performance bounds in communication networks with variable-rate links. In : *ACM SIGCOMM Computer Communication Review*. Vol. 25. pp. 126–136.

Lelevé, A. et P. Fraisse (2003). Teleoperation over IP network : Network delay regulation and adaptive control. *Journal of Autonomous Robots, special issue "Internet and online robots"*, **15**(3), 225–235.

Lian, F-L. (2001). Analysis, design, modeling and control of networked control systems. PhD thesis. Department of Mechanical Enginneering, University of Michigan.

Lian, F.-L., J.R. Moyne et D.M. Tilbury (2001). Performance evaluation of control networks. *IEEE Control Systems magazine*, **21**(1), 66–83.

Liberatore, V., M.S. Branicky, S.M. Phillips et P. Arora (2004). Networked control systems repository. Internet - http ://home.cwru.edu/ncs/.

Liu, C.L. et J.W. Layland (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, **20**(1), 46–61.

Luck, R. et A. Ray (1990). An observer-based compensator for distributed delays. *Automatica*, **26**(5), 903–908.

Marsal, G., D. Witsch, B. Denis, J.-M. Faure et G. Frey (2005). Evaluation of real-time capabilities of Ethernet-based automation systems using formal verification and simulation. In : *1ères Recontres Jeunes Chercheurs en Informatique Temps Réel, Ecole d'été Temps Réel (RCJITR'05 - ETR'05).*

Michaut, F. (2003). Adaptation des applications distribuées à la Qualité de Service fournie par le réseau de communication. PhD thesis. Université Henri Poincaré, Nancy 1, Centre de Recherche en Automatique de Nancy.

Michaut, F. et F. Lepage (2003). Un cadre pour la prise en compte du retard dans la téléopération d'un robot mobile. In : *JDA'03 "Journées Doctorales d'Automatique"*. Valenciennes.

Mills, D.M. (1991). Internet Time Synchronization : The Network Time Protocol. *IEEE Transactions on Communications*, **39**(10), 1482–1493.

Modlovansky, A. (1998). Utilization of modern switching technology in EtherNet/IP networks. In : *1st International Workshop on Real-Time LANs in the Internet Age (RTLIA'02).* Vienne, Autriche. pp. 35–37.

Molle, M.L. (1994). A new Binary Logarithmic Arbitration Method for Ethernet. Technical Report CSRI-298. Computers Research Institute, Université de Toronto, Canada.

Naudts, J. (2000). Towards real-time measurements of traffic control parameters. *Computer Networks*, **34**(3), 157–167.

Nilson, J. (1998). Real-time control systems with delays. PhD thesis. Lund Institute of Technology, Department of Automatic Control.

Nutzer Organisation e.v. (1992). Profibus standard DIN 19245 Part I and II. Translation from German.

Parekh, A.K. (1992). A generalized processor sharing approach to flow control in integrated services networks. PhD thesis. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Parekh, A.K. et R.G. Gallager (1993). A generalized processor sharing approach to flow control in integrated services networks : The single node case. *IEEE/ACM Transactions on Networking*, **1**(2), 344–357.

Pasztor, A. et D. Veitch (2001). A precision infrastructure for active probing. In : *Workshop on Passive and Active Measurements (PAM'01).* Amsterdam.

Pedreiras, P. (2003). Supporting flexible real-time communication in distributed systems. PhD thesis. Université d'Aveiro.

Pedreiras, P. et L. Almeida (2002). Flexibility, timeliness and efficiency over Ethernet. In : *1st International Workshop on Real-Time LANs in the Internet Age (RTLIA'02).* Vienne, Autriche. pp. 53–56.

Phipps, M. (1999*a*). A guide to LAN switch architectures and performance. In : *Packet Cisco Systems user magazine.* Vol. 4. pp. 54–57. Cisco Systems.

Phipps, M. (1999*b*). LAN switch architectures and performance. In : *Networkers*. Cisco Systems. Session 603.

Poulard, G., B. Denis et J.-M. Faure (2004). Modélisation par réseau de Petri coloré des architectures de commande distribuées sur réseau de terrain Ethernet et TCP/IP. In : *5ème Conférence Francophone de Modélisation et Simulation (MOSIM'04)*. Nantes. pp. 405–412.

Pritty, D.W., J.R. Malone, D.N. Smeed, S.K. Banerjee et N.L. Lawrie (1995). A real-time upgrade for Ethernet based factory networking. In : *21st IEEE International Conference on Industrial Electronics (IECON'95)*. Vol. 2. Orlando. pp. 1631–1637.

Ramakrishnan, K.K. et H. Yang (1994). The Ethernet capture effect : Analysis and solution. In : *19th IEEE Local Computer Networks conference (LCN'99)*. Minneapolis Minn. pp. 228–240.

Richard, J.P. (2003). Time-delay systems : an overview of some recent advances and open problems. *Automatica*, **39**(10), 1667–1694.

Richard, J.P. (2005). Systèmes à retard. Journée du groupe de travail STRQDS du GDR ARP sur le thème Networked Control Systems.

Rockwell International Corporation (1998). Ethernet for industrial control, An Ethernet white paper. Technical report.

Rondeau, E. (2001). Conception d'architectures de réseaux locaux par analyse du trafic. Habilitation à Diriger des Recherches, Université Henri Poincaré Nancy 1, CRAN.

Rondeau, E., T. Divoux et H. Adoud (2001). Study and method of Ethernet architecture segmentation for Industrial applications. In : *4th IFAC Conference on Fieldbus Systems and Their Applications (FET'2001)*. Nancy, France. pp. 165–172.

Rüping, S., E. Vonnahme et J. Jasperneite (1999). Analysis of switched Ethernet networks with different topologies used in automation systems. In : *3rd IFAC International conference on Fieldbus Systems and Their Applications (FET'99)*. Magdeburg, Allemagne. pp. 351–358.

Seifert, R. (2000). *The switch book*. John Wiley & Sons, Inc.

Song, Y.Q. (2001). Time constrained communication over switched Ethernet. In : *4th IFAC International conference on Fieldbus Systems and Their Applications (FET'01)*. Nancy, France. pp. 152–169.

Song, Y.Q. (2004). Qualité de service dans les réseaux et systèmes temps réel distribués : contributions à l'évaluation de performances temporelles et à l'ordonnancement. Habilitation à diriger des recherches, Université Henri Poincaré Nancy 1, LORIA.

Starobinski, D. et M. Sidi (2000). Stochastically bounded burstiness for communication networks. *IEEE Transactions on Information Theory*, **46**(1), 206–212.

The ATM forum (1996). ATM User Network Interface. Version 4.0.

Thiele, L., S. Chakraborty et M. Naedele (2000). Real-time calculus for scheduling hard real-time systems. In : *IEEE International Symposium on Circuits and Systems (ISCAS'00)*. Vol. 4. Genève. pp. 101–104. ISCAS.

Tobagi, F. (1982). Carrier sense multiple access with message-based priority functions. *IEEE Transactions on Communications*, **30**(1), 185–200.

Torab, P. (2000). Performance analysis of packet switched networks with tree topology. PhD thesis. School of Electrical and Computer Engineering.

Torab, P. et E.W. Kamen (1999). Load analysis of packet switched networks in control systems. In : *25th Annual Conference of the IEEE International Electronics Society (IECON'99)*. Vol. 3. San Jose, California. pp. 1222–1227.

Turiel, J., J. Marinero et J. González (1996). CSMA/PDCR : a random access protocol without priority inversion. In : *22nd International Conference on Industrial Electronics, Control, and Instrumentation (IECON'96)*. Vol. 2. Taipei, Taiwan. pp. 910–915.

Varadarajan, S. et T. Chiueh (1998). EtheReal : a host-transparent real-time Fast Ethernet switch. In : *6th IEEE International Conference on Network Protocols (ICNP'98)*. Austin. pp. 12–21.

Vojnovic, M. et J.Y. Le Boudec (2002). Elements of probabilistic network calculus for packet scale rate guarantee nodes. In : *15th International Symposium on Mathematical Theory of Networks and Systems (MTNS'02)*. University of Notre Dame, Indiana.

Walsh, G.C., H. Ye et L. Bushnell (1999). Stability analysis of networked control systems. In : *American Control Conference*. San Diego, Etats Unis. pp. 2876–2880.

Wang, P. (1996). Class-of-service (CoS) implementation for PACE. Technical report. 3Com Technology Brief. version 0.96b.

Watson, K. (2002). Network calculus in star and line networks with centralized communication. Technical Report 10573, Projet 393444. Fraunhofer Institut Informations und Datenverarbeitung. Karlsruhe, Allemagne.

Watson, K. et J. Jasperneite (2003). Determining end-to-end delays using network calculus. In : *5th IFAC International conference on Fielbus Systems and their Applications (FET'03)*. Aveiro, Portugal. pp. 255–260.

Wollschlaeger, M. (2000). A framework for fieldbus management using XML descriptions. In : *3rd IEEE International Workshop on Factory Communication Systems (WFCS'2000)*. Porto. pp. 3–10.

Yang, Y., Y. Wang et S.H. Yang (2005). A networked control system with stochastically varying transmission delay and uncertain process parameters. In : *16th IFAC World Congress*. Prague, République Tchèque.

Yavatkar, R., P. Pai et R. Finkel (1992). A reservation-based CSMA protocol for integrated manufacturing networks. Technical Report CS-216-92. Department of Computer Science, University of Kentucky.

Zhang, W., S. Branicky et S. Phillips (2001). Stability of networked control systems. *IEEE Control Systems magazine*, **21**, 84–89.

Zhao, W. et K. Ramamritham (1986). A Virtual Time CSMA Protocol for Hard Real Time Communication. In : *7th IEEE Real-Time Systems Symposium (RTSS'86)*. Nouvelle Orléans. pp. 120–127.

Zhao, W. et K. Ramamritham (1987). Virtual time CSMA protocols for hard real-time communication. *IEEE Transactions on Software Engineering*, **13**(8), 938–952.

Résumé

La recherche dans le domaine des systèmes contrôlés en réseau a considérablement évolué depuis qu'Ethernet est de plus en plus utilisé en remplacement des traditionnels réseaux locaux industriels. Si ce choix d'Ethernet se justifie par sa faculté intrinsèque à supporter toutes les communications de l'entreprise (du bureau à l'atelier), il s'accorde difficilement avec les contraintes temporelles des applications de contrôle / commande distribuées. Contrairement aux réseaux de terrain, l'indéterminisme de l'accès à la voie d'Ethernet ne permet pas de garantir le respect des contraintes temporelles strictes.

La contribution de cette thèse est la définition d'une approche analytique permettant de majorer les délais de communication de bout en bout dans les systèmes contrôlés en réseau lorsqu'ils reposent sur une architecture Ethernet commutée. Les travaux se sont focalisés sur l'adaptation de la théorie du calcul réseau à ce type d'environnement. Dans ce cadre, cette thèse développe la modélisation d'un commutateur IEEE 802.1D ainsi qu'une méthode de calcul des délais de bout en bout basée sur l'augmentation du volume des rafales. Plusieurs expérimentations réelles ont permis de valider l'efficacité des bornes obtenues par calcul.

Outre l'évaluation de performances, ces travaux s'intéressent également à la Classification de Service (IEEE 802.1D/p) et à l'optimisation de l'ordonnancement des trames sur Ethernet. Cette thèse montre enfin comment une méthode d'évaluation de performances peut être utilisée pour dimensionner et optimiser la conception d'architectures Ethernet commutées.

Mots clés

Systèmes contrôlés en réseau, systèmes temps-réel, évaluation de performances, Network Calculus, Ethernet commuté, classification de service

Abstract

In the field of networked control systems, researches have considerably evolved since Ethernet is more and more used to substitute the traditional industrial local area networks. Even if this choice of Ethernet is justified regarding its intrinsic ability to support all the communications of the enterprise (from office to workshop), it is not suitable to assume the time constraints of distributed control applications. Contrary to the fieldbuses, the non-determinism of the medium access method used by Ethernet does not enable to guarantee strict time constraints.

The contribution of this thesis is to define an analytical approach to upper-bound the end-to-end delays in networked control systems which are based on switched Ethernet architectures. Work has focused on the adaptation of the network calculus theory to these specifics environments. Within this framework, this thesis presents the modeling of an IEEE 802.1D switch as well as a computation method of the end-to-end delays based on the increasing of the traffic burstiness. Several real experiments validate the tightness of the computed bounds.

In addition to the performance evaluation, the work has also considered the Classification of Service (IEEE 802.1D/p) and the optimization of the frames scheduling on Ethernet. Finally, this thesis shows how such a performances evaluation method can be used to scale and optimize the design of switched Ethernet architectures.

Keywords

Networked control systems, real-time systems, performance evaluation, Network Calculus, switched Ethernet, classification of service

Université Henri Poincaré, Centre de Recherche en Automatique de Nancy

Faculté des Sciences, B.P. 239, 54506 Vandœuvre lès Nancy