



**HAL**  
open science

## Fair queueing pour les systèmes contrôlés en réseau

Jean-Philippe Georges

► **To cite this version:**

Jean-Philippe Georges. Fair queueing pour les systèmes contrôlés en réseau. École d'été Temps Réel, ETR'09, Aug 2009, Paris, France. pp.157-171. <hal-00413726>

**HAL Id: hal-00413726**

**<https://hal.science/hal-00413726v1>**

Submitted on 5 Sep 2009

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

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



HAL Authorization

# Fair queueing pour les systèmes contrôlés en réseau

Jean-Philippe Georges  
CRAN, Nancy-Université, CNRS  
Boulevard des Aiguillettes, BP 70239  
F-54506 Vandœuvre-lès-Nancy, France  
jean-philippe.georges@cran.uhp-nancy.fr

## Abstract

Les nouveaux systèmes de contrôle-commande industriels et embarqués se caractérisent par l'interconnexion en réseau de multiples équipements via des réseaux non plus dédiés, mais partagés avec d'autres applications. Dans ces systèmes contrôlés en réseau, il est nécessaire que la Qualité de Service du réseau ait un impact limité sur la Qualité de Performance (QdP) du système. Cet article s'appuie sur l'utilisation d'un réseau non déterministe, Ethernet, où seule l'utilisation de techniques de Classification de Service va permettre de poursuivre cet objectif. Différentes stratégies de configuration d'ordonnement équitable de type weight round robin sont mises en œuvre à la recherche de deux objectifs : garantir la QdP tout en maximisant la bande passante fournie au trafic non temps-réel.

## 1. Systèmes contrôlés en réseau (SCR)

### 1.1. Introduction

Un système contrôlé en réseau (SCR) (ou *networked control systems* dans la littérature anglophone) correspond à un système de contrôle/commande distribué via un réseau pouvant être partagé avec d'autres applications non impliquées dans la commande du système. Plus précisément, un SCR est un système de contrôle à asservissement dans lequel les boucles de régulation sont fermées au moyen d'un réseau de communication [39, 33] comme le montre la figure 1.

Le domaine des systèmes contrôlés en réseau 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 notamment dans les systèmes de production, mais aussi dans les systèmes embarqués comme les avions ou les automobiles. Historiquement les SCR s'appuient sur l'implémentation des contrôles distribués définis au travers du Distributed Control System (DCS) de Honeywell dans les années 70. L'introduction d'un réseau dans la boucle de commande vise notamment à la réduction des coûts de câblage, l'aide au diagnostic et à la main-

tenance des systèmes, l'amélioration de la modularité et de la flexibilité dans la conception des systèmes. Aujourd'hui, plusieurs compagnies comme Rockwell Automation, ABB, Siemens proposent des équipements intégrant leur propre coupleur de communication réseau, si bien que plusieurs réseaux sont des composants naturels des SCR actuellement en service comme DeviceNet, Fip, CAN, Profibus, Modbus, etc.

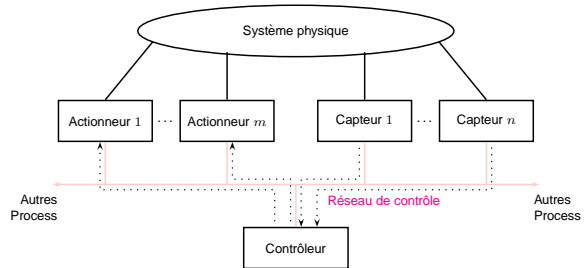


FIGURE 1. Vue traditionnelle des systèmes contrôlés en réseau [39]

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 [39] 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 le rendre instable,
- le réseau fournit un ensemble de chemins non fiables. Il faut donc prendre en compte le fait que des paquets peuvent être perdus, dupliqués, désordonnés,
- l'information peut être contenue dans plusieurs messages. Les chances que la totalité, une partie ou au-

cun des paquets arrivent doivent alors également être étudiées.

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).

Compte tenu des particularités de la ressource « réseau » (délais, giges, 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. Et comme chaque type de réseau répond à des critères de performances différents tels la vitesse de transfert, la taille des messages, des majorants du délai et la disponibilité, l'analyse d'un SCR est implicitement liée au protocole de communication mis en œuvre.

## 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 menée dans le projet européen STREP NeCST<sup>1</sup>. 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 2. Un rouleau central motorisé commande la traction de la bande et lui assure une tension adéquate.

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

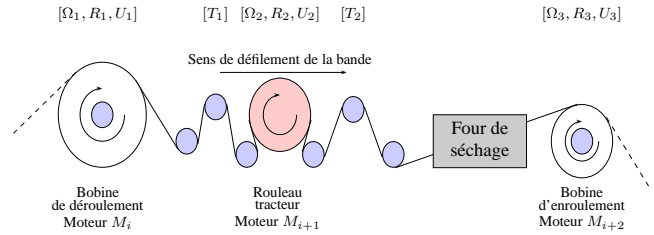


FIGURE 2. Éléments d'un système d'enroulement de bande

$U_i$ ) 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  $\Omega_2$  mesure la vitesse angulaire de  $R_2$ .

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

La plateforme de test est schématisée sur la figure 3.

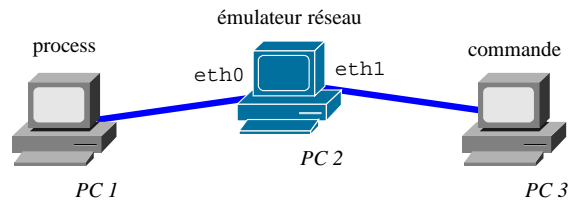


FIGURE 3. Simulation/émulation de SCR

L'expérimentation consiste en une modélisation de la commande et du système sous MATLAB/Simulink et à une émulation du réseau (l'émulateur permet d'introduire des délais et des pertes sur le réseau). Deux PC distincts sont utilisés pour exécuter en temps-réel la simulation de la commande et du système. Les PC 1 et 3 exécutent respectivement les modèles Simulink du process et de la commande. Ces deux PC sont interconnectés via un réseau ethernet. Le troisième PC sera utilisé pour émuler les performances du réseau à l'aide de l'émulateur *netem*. Les états observés correspondent aux trois mesures ( $T_1$ ,  $\Omega_2$  et  $T_2$ ). À ce stade, nous présentons ici directement les valeurs obtenues pour trois valeurs de délai. Pour plus d'information, le lecteur pourra se reporter au livrable [2] du projet européen NeCST.

La figure 4 montre bien qu'un retard trop élevé (c'est-à-dire une faible qualité de service) 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 retard, l'apparition d'un délai de transmission provoque un dépassement de la consigne (figure 4(b)) et peut dans le cas extrême rendre le système instable (figure 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

1. <http://www.strep-necst.org/>

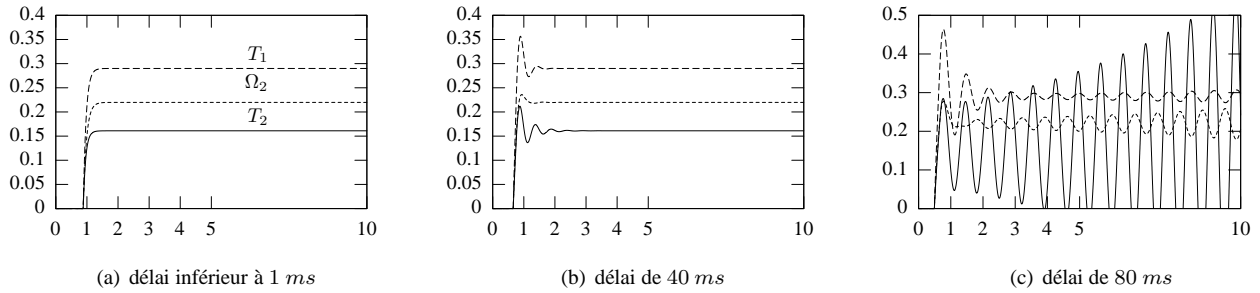


FIGURE 4. Dépassement en sortie d'un SCR dû aux délais introduit par le réseau (la grandeur en abscisse est le temps en secondes)

délaï 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.

### 1.3. Présentation générale des travaux de recherche

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. Un tutoriel des travaux et des problématiques est donné dans [32]. 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, 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 utilisées pour assurer cette hypothèse. Ainsi que ce soit dans le cadre de la téléopération d'un robot [24, 26] ou pour les applications distribuées [28], 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 5.

Ce modèle est établi dans le cadre d'une architecture producteur / consommateur semblable à la figure 1, mais

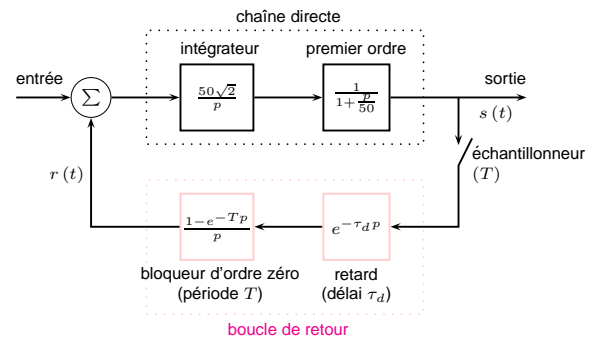


FIGURE 5. Asservissement distribué [19]

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 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 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 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

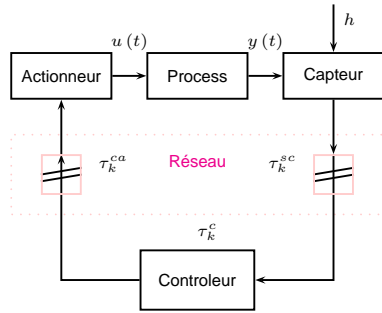


FIGURE 6. **Modèle d'asservissement distribué [29]**

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 7, ce modèle doit être complété lorsque l'application intègre plusieurs entrées et plusieurs sorties.

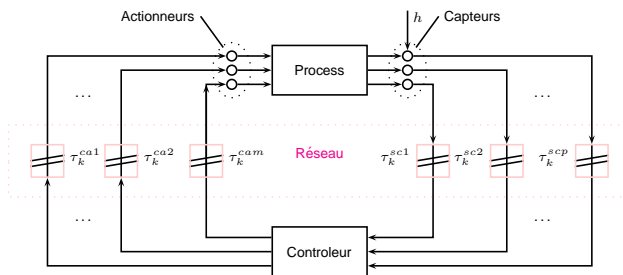


FIGURE 7. **Modèle d'asservissement distribué à multiples entrées et sorties [29]**

La figure 7 introduit un aspect supplémentaire de la difficulté d'analyse des SCR. 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, 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 que la commandabilité et la stabilité du contrôle ainsi que l'intégration des niveaux de performance du réseau dans le contrôle de l'application.

Le lecteur intéressé par la recherche dans le domaine des SCR pourra consulter les papiers introductifs [13, 27, 34, 33] et le dernier état de l'art publié dans [38]. Plus globalement, deux grandes directions peuvent être distinguées selon que le SCR est étudié du point de vue de la commande du système ou alors du point de vue réseau.

Dans la première approche généralement appelée « commande en réseau », la commande s'adapte aux perturbations induites par le réseau. Dans ce contexte,

le réseau est une ressource passive apportant différentes contraintes exprimées en termes de Qualité de Service. Le réseau est ainsi vu comme une « boîte noire » associée à un système à retard variable. En fonction des informations disponibles concernant la QoS offerte par le réseau, plusieurs stratégies de compensation ont été proposées : cela va de l'adaptation traditionnelle du gain PID, la synthèse de commande prédictive ou encore de commande robuste. L'enjeu majeur porte généralement sur la stabilité du système. Ces résultats relevant de la problématique des systèmes à retard, [32] présente un état de l'art des derniers développements dans ce domaine.

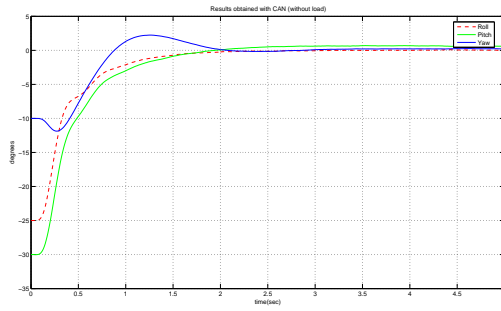
Dans le second cas, le réseau est désormais une ressource « active » et commandable. Cette approche qui sera détaillée dans la suite, est appelée « contrôle de réseau » [38]. L'adaptation du réseau aux exigences de la commande du système vise à optimiser les performances que ce soit de manière endogène (paramètres de QoS) ou exogène, c'est-à-dire relativement à la Qualité de Performance (état courant du système comme sa stabilité ou l'erreur). La caractéristique fondamentale des travaux relatifs à cette approche est d'être profondément marqués par le type de réseau étudié puisque celui-ci restreint les possibilités d'optimisation. Ainsi lorsque [20] relie différents panels de priorité sur CAN à la marge de phase, [37] présente une fonction de dégradation des performances pour différentes architectures. De même lorsque [36] propose un nouveau protocole *try-once-discard*, [23] propose une nouvelle politique d'ordonnancement type TDMA pour Profibus.

Dans cet article, l'étude des SCR se focalisera sur le réseau Ethernet dont l'indéterminisme, et donc son incapacité à satisfaire des exigences temps-réel, le rend a priori inadapté comme support aux SCR. L'utilisation de la Classification de Service et de politiques d'ordonnancement équitables seront alors utilisées en vue de répondre à ces exigences.

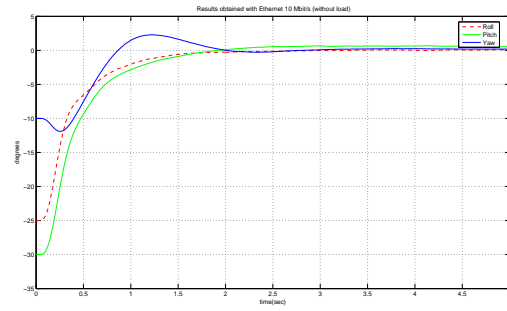
## 2. Ethernet commuté et classification de service pour les SCR

### 2.1. Ethernet, un réseau non déterministe

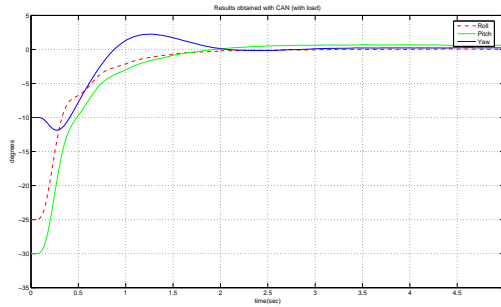
Le recours à l'utilisation de réseaux dans la commande en boucle fermée de systèmes s'est initialement tournée vers les réseaux locaux industriels classiques, comme le réseau CAN. La raison principale s'explique naturellement par le comportement déterministe de ces réseaux, les rendant adaptés aux SCR. Cependant, de nouvelles exigences sont apparues comme de nouvelles fonctionnalités (e-maintenance, télé-opération), mais aussi la volonté de partager le médium avec d'autres applications. Face à cette intensification de l'utilisation de la bande passante et la multitude des messages supportés, le protocole Ethernet apparaît alors comme une solution pour les SCR. Les intérêts majeurs sont donc liés au support de débits largement supérieurs (jusqu'au Gb/s) aux réseaux locaux traditionnels et à la standardisation effective d'Ethernet [15].



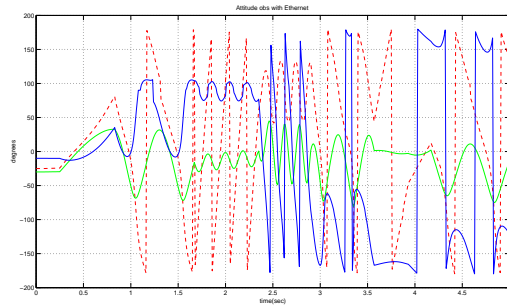
(a) CAN 1 Mb/s sans charge



(b) Ethernet partagé 10 Mb/s, sans charge



(c) CAN 1 Mb/s avec charge



(d) Ethernet partagé 10 Mb/s, avec charge

FIGURE 8. Problématique d’Ethernet dans les réseaux partagés (les graphes représentent l’évolution des angles au cours du temps en seconde)

C’est ainsi qu’Ethernet est aujourd’hui de plus en plus implanté dans les systèmes temporellement contraints (citons par exemple l’Airbus A380 d’EADS).

Néanmoins, l’utilisation d’Ethernet comme support de SCR pose différentes problématiques. Dans un premier temps, en termes de sûreté de fonctionnement, différentes précautions doivent être prise afin de faire face notamment à la défaillance d’un lien, ou même du bus. Il s’agit donc de construire des topologies redondantes (par exemple, redondance de liens, d’équipements d’interconnexion, etc). La gestion de cette redondance physique doit également être accompagné d’une topologie logique (algorithmes d’arbre couvrant comme STP ou RSTP) optimisant les temps de détection de défaillance et de reconfiguration de la topologie. Cette problématique est notamment abordée dans [21].

Dans un second temps, l’augmentation de la bande passante ne solutionne pas tous les problèmes. En effet, le protocole CSMA/CD utilisé par Ethernet conduit à un accès au médium non déterministe du fait de la probabilité d’occurrence de collisions [1]. L’occurrence de ces collisions sera d’autant plus grande lorsque le réseau est partagé par différentes applications (qu’elles soient ou non impliquées dans la commande du système). Ce phénomène est ainsi mis en exergue dans [9] pour la commande en réseau d’un drone hélicoptère à quatres rotors développé par le GIPSA-lab dans le cadre du projet

ANR SafeNecs<sup>2</sup>. Pour ce système, le réseau embarqué doit supporter le trafic généré par 17 flux périodiques, où la période d’échantillonnage est fixée à 10 ms. La figure 8 présente les résultats de simulation conjointe (système, commande et réseau) du SCR réalisé via la librairie True-Time sous Matlab/Simulink. L’étude considère tour à tour le réseau Ethernet et le réseau CAN (comme référence).

La figure 8(b) montre qu’Ethernet peut réussir à fournir des performances du même niveau que CAN pour la commande du système, dans le contexte d’un réseau dédié (sans trafic annexe). Néanmoins, la figure 8(d) montre ensuite que si le réseau est partagé avec d’autres applications, Ethernet, malgré sa bande passante de 10 Mb/s, impacte sensiblement la commande puisque cette fois le système est instable. Le problème pour la commande de systèmes industriels ou embarqués est lié au fait qu’aucune garantie de performance n’est fournie aux messages à fortes contraintes temporelles.

## 2.2. Ethernet commuté, un réseau *best-effort*

Le problème des collisions préalablement étudié peut être résolu par l’utilisation d’architectures Ethernet commuté à liens *full-duplex*. En effet, l’utilisation de commutateurs [14] permet de réduire le domaine de collisions au lien, et l’utilisation de liens *full-duplex* point à point résout définitivement ce problème. L’utilisation de telles architectures va donc permettre d’éliminer l’indéterminisme de

2. <http://safe-necs.cran.uhp-nancy.fr/>

l'accès au médium. Néanmoins, la figure 9 montre que de telles architectures peuvent encore impacter dramatiquement la commande, puisque le système est ici toujours instable.

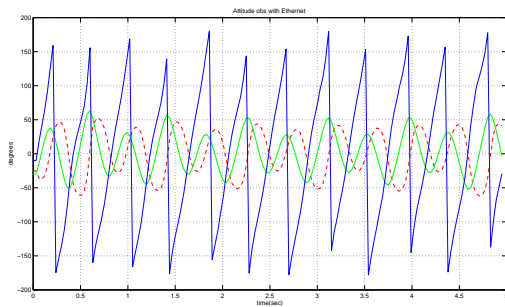


FIGURE 9. Problématique d'Ethernet commuté dans les réseaux partagés

L'instabilité observée à la figure 9 s'explique par le fait que le trafic réseau va induire une charge (congestion) du commutateur et en conséquence une augmentation des délais. L'ordonnancement utilisé par le standard [14] étant de type FIFO, il n'existe aucune différenciation de service entre le trafic de commande et le trafic non temps-réel. À l'inverse, le réseau CAN de la figure 8 intégrant un ordonnancement de type priorité stricte, reste capable de fournir un service suffisant bien que le débit soit plus faible. L'utilisation d'Ethernet comme support au SCR passe donc dans le cas d'un réseau partagé, par l'utilisation de mécanismes d'ordonnancement plus avancés comme décrit ci-dessous.

### 2.3. 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 sans pour autant garantir un niveau de service. Dans ce cadre, cela sous entend également une première étape qu'est la priorisation 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 (QoS). La QoS se définit par la garantie que donne le réseau à une application en terme de 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 QoS 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 [17].

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 QoS.

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 priorisation 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

Un projet nommé P802.1p fut lancé en 1995 afin de traiter initialement les problèmes de priorisation 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 priorisation pour Ethernet amenèrent alors à intégrer le projet P802.1p au standard existant IEEE 802.1D [14] 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 régénération 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 (figure 10).

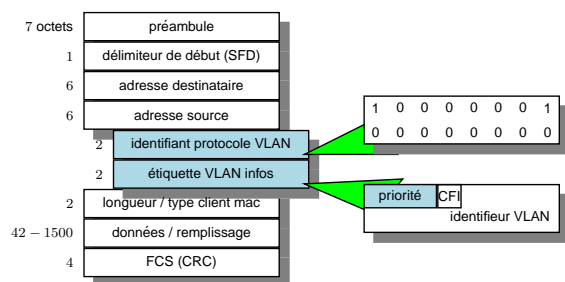


FIGURE 10. Format de la trame Ethernet étiquetée

### 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 définie à la table 1.

TABLE 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

La plupart des implémentations utilisent deux 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 pour chaque port de sortie comme le montre la figure 11. Le standard 802.1p prévoit enfin que le nombre de classes de service puisse être différent du nombre de niveaux de priorité.

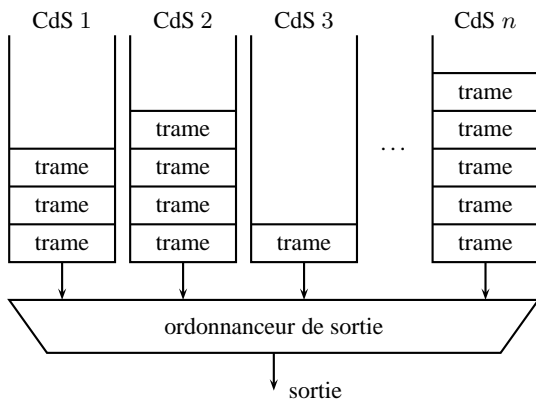


FIGURE 11. **Instanciation d'une file d'attente pour chaque classe de service au niveau d'un port de sortie d'un commutateur**

L'instanciation de plusieurs files en sortie d'un commutateur nécessite d'adopter une stratégie d'ordonnement. Les standards [16, 14] précisent alors que la discipline de service définie par défaut est la politique à priorité stricte. Néanmoins, ils laissent la possibilité à l'utilisation d'autres politiques, sous réserve que la discipline de service d'une seule file doit rester FIFO. Toutefois, une étude des produits actuellement commercialisés montre que seule la discipline WFQ (*weighted fair queueing*) est implantée. Aussi seules ces 2 politiques seront considérées.

Finalement, le recours à la Classification de Service 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.

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 [18], [11] ou [12] qui présente une optimisation des différents paramètres de la CdS.

### 3. Analyse de performances

L'étude suivante vise à la modélisation du service supporté par un commutateur implantant les politiques à priorité stricte et de type WFQ. Les développements suivants se basent sur la théorie du calcul réseau développée dans [5, 6, 7, 22]. L'arrivée de trafic sera supposée comme contrainte par une courbe affine (régulation type seuil percé).

#### 3.1. Modélisation du commutateur à priorité

Comme le montre la figure 12, 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 12, 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'ordonnement entre les différentes files. La politique de retransmission utilisée par ces multiplexeurs de sortie sera donc fixée par l'algorithme d'ordonnement à priorité stricte ou *Weighted Fair Queueing* (WFQ). 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 12 représente le modèle d'un commutateur deux ports gérant deux niveaux de priorité.

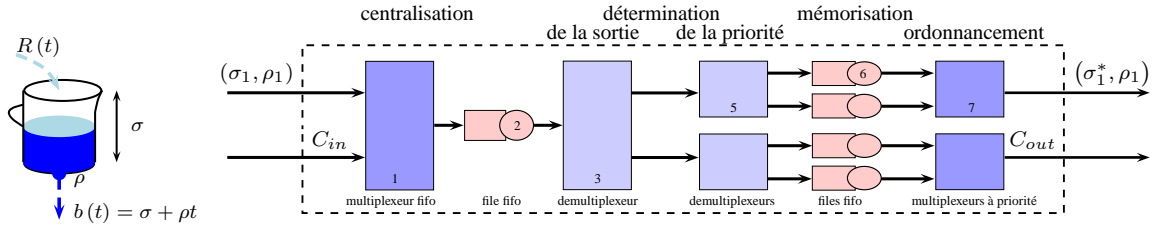


FIGURE 12. Le mécanisme de priorisation modifie la mise en attente au niveau des ports de sortie

L'étape suivante consiste alors à exprimer des délais de traversée de chacun des composants élémentaires introduits par le modèle de la figure 12. Nous nous intéressons simplement au port de sortie, les expressions définies au chapitre précédent pour les composants en amont restent vraies. 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  $\sum_j \rho_j < C$  comme le montre la figure 12. De plus, un poids  $\phi_i$  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 la quantité de données cumulée retransmise en sortie  $R_i^*(t)$  est contrainte par  $R_i^*(t) \leq Ct$ . 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.

### 3.2. 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 13.

#### 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. Si l'on note  $L_i$  la longueur maximale des

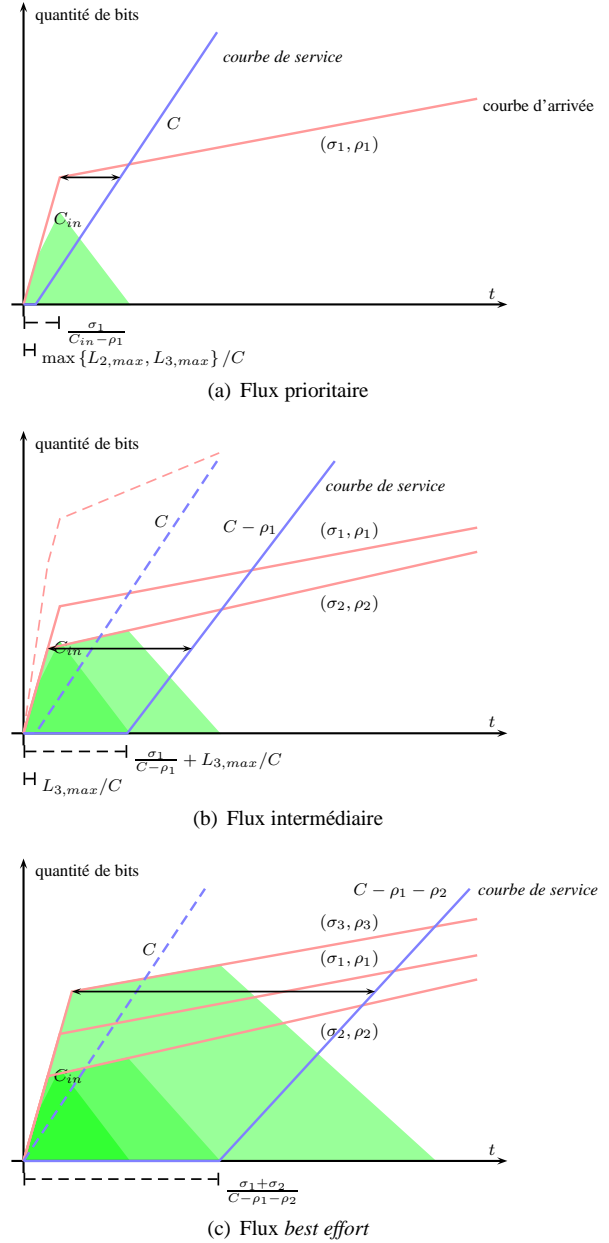


FIGURE 13. Courbe de service & arriéré de traitement pour un ordonnancement statique à priorité stricte.

trames du flux  $i$ , on obtient la courbe de service :

$$\beta_1(t) = R(t - T)^+ \quad (1)$$

$$T = \max \{L_{2,max}, L_{3,max}\} / C, R = C$$

### 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 :

$$\beta_2(t) = R(t - T)^+ \quad (2)$$

$$T = \frac{\sigma_1}{C - \rho_1} + \frac{L_{3,max}}{C}, \quad R = C - \rho_1$$

### 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 :

$$\beta_3(t) = R(t - T)^+ \quad (3)$$

$$T = \frac{\sigma_1 + \sigma_2}{C - \rho_1 - \rho_2}, \quad R = C - \rho_1 - \rho_2$$

Les équations (1), (2) et (3) montrent qu'avec la politique à priorité stricte, le service offert à un flux dépend des autres flux. De surcroît, l'équation (3) démontre que le service offert au flux de priorité basse peut tendre vers zéro (c'est l'effet "famine" engendré par cet algorithme d'ordonnement).

### 3.3. Ordonnement WFQ

#### Algorithmes équitables

Le *Weighted Fair Queueing*, initialement proposé dans [8], est également connu sous la dénomination *Packetized Generalized Processor Sharing* (PGPS) [30]. Il est basé sur l'algorithme conceptuel appelé *Generalized Processor Sharing* (GPS) [31]. 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  $\phi_i$  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  $(\sigma, \rho)$ , [5] 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'ordonnement idéaliste qui suppose que les paquets sont infinitivement divisibles. Aussi, [31] présentent un schéma de transmission paquet par paquet qui approxime GPS : PGPS. Lorsqu'il est prêt, un nœud PGPS sélectionne le paquet qui serait le premier retransmis dans la simulation GPS comme le montre la figure 14

La figure 14 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 [31]) 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'ordonneur choisit le paquet présentant l'étiquette temporelle la plus petite parmi l'ensemble des paquets disponibles.

Comme le montre la figure 14, [31] 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  $\max\{x, 0\}$ , [5] 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 [3], 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

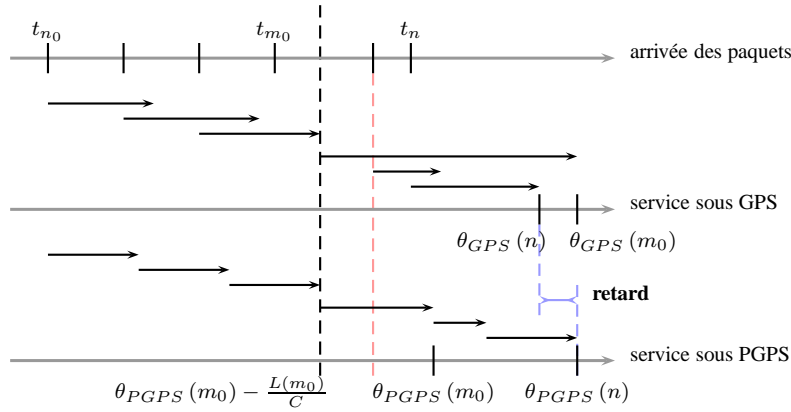


FIGURE 14. Contexte de retardement du service offert à un paquet sous PGPS par rapport à GPS.

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  $(\sigma, \rho)$  des autres flux. Si l'on souhaite améliorer le service offert à un trafic à temps-critique, les poids (noté  $\phi_i$  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  $\phi_i$  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  $\phi_i$  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.

La définition de la courbe de service minimale offerte à un flux va donc utiliser les propriétés mises en avant ci-

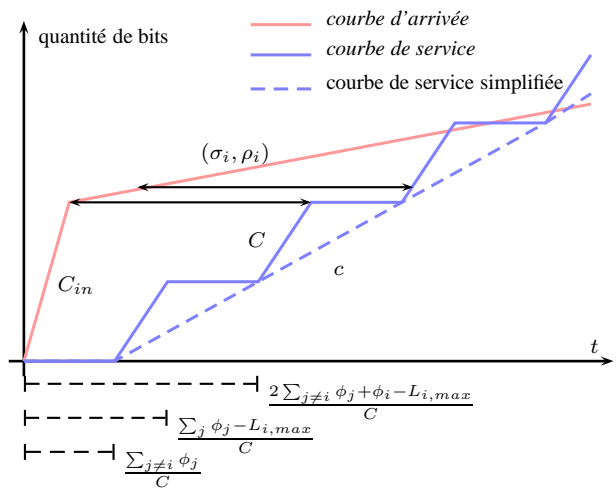


FIGURE 15. Courbe de service weighted round robin.

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 15 illustre la courbe de service qui en est déduite et l'équation (4) formule le service offert à un flux.

$$\beta(t) = \sup \left( C \left( t - \left\lceil \frac{t}{x} \right\rceil \frac{\sum_{j \neq i} \phi_j}{C} \right)^+, \phi_i \left\lfloor \frac{t}{x} \right\rfloor \right) \quad (4)$$

avec  $x = \frac{\sum_j \phi_j - L_{i,max}}{C}$  la durée du cycle pour  $i$ . Comme l'équation (4) peut engendrer une complexité de calcul, notons qu'une autre courbe de service peut également être proposée.

$$\beta_{R,T}(t) = R(t - T)^+ \quad (5)$$

$$R = C \frac{\phi_i - L_{i,max}}{\sum_j \phi_j - L_{i,max}}, T = \frac{\sum_{j \neq i} \phi_j}{C}$$

La courbe de service proposée à l'équation (5) 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 défini par la courbe de service donnée par l'équation (4). L'intérêt dégagé par ce type de courbe de service est sa simplicité par rapport au pseudo-escalier introduit dans l'équation (4).

Ces deux équations (4) et (5) 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 (5) 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.

### 3.4. Majorant des délais pour SP et WFQ

Comme nous l'avons vu au chapitre précédent, la figure 15 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 :

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

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  $R(t + \Delta - T)^+ = R(t + \Delta - T)$

$$\begin{aligned} \overline{D}_i &= T \vee \max_{0 \leq t < \tau_i} \left\{ \inf_{\Delta \geq 0} \{R\Delta = (C_{in} - R)t + RT\} \right\} \\ &\quad \vee \max_{t \geq \tau_i} \left\{ \inf_{\Delta \geq 0} \{R\Delta = (\rho_i - R)t + \sigma_i + RT\} \right\} \\ &= T \vee \max_{0 \leq t < \tau_i} \frac{(C_{in} - R)t + RT}{R} \\ &\quad \vee \max_{t \geq \tau_i} \frac{(\rho_i - R)t + \sigma_i + RT}{R} \end{aligned}$$

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  $\tau_i$  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 :

$$\overline{D}_i = (T - \tau_i) + \frac{\sigma_i + \rho_i\tau_i}{R} \quad (6)$$

Dans le cas d'une architecture à  $n$  ordonnanceurs, la méthode développée dans le cadre du calcul réseau consiste à calculer au préalable la composition des serveurs définie suivant le principe *pay bursts only once* [22]. Pour les architectures plus compliquées à agrégation de plusieurs flux de même classe de service (ou priorité), les techniques de service résiduel combinées au principe *pay multiplexing only once* doivent être instanciées [4, 25].

### 3.5. Conclusion

L'équation (6) 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*. Par souci de clarté, 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 :

$$\begin{aligned} \overline{D}_{1,PQ} &= \left( \frac{\max(L_{2,max} + L_{3,max})}{C} - \tau_1 \right) + \frac{\sigma_1 + \rho_1\tau_1}{C} \\ \overline{D}_{1,WRR} &= \left( \frac{\phi_2 + \phi_3}{C} - \tau_1 \right) + \frac{\sigma_1 + \rho_1\tau_1}{C \frac{\phi_1 - L_{1,max}}{\phi_1 + \phi_2 + \phi_3 - L_{1,max}}} \end{aligned}$$

Il est alors intéressant de remarquer ici que si  $\phi_2 \rightarrow 0$  et  $\phi_3 \rightarrow 0$ , 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 :

$$\begin{aligned} \overline{D}_{3,PQ} &= \left( \frac{\sigma_1 + \sigma_2}{C - \rho_1 - \rho_2} - \tau_3 \right) + \frac{\sigma_3 + \rho_3\tau_3}{C - \rho_1 - \rho_2} \\ \overline{D}_{3,WRR} &= \left( \frac{\phi_1 + \phi_2}{C} - \tau_3 \right) + \frac{\sigma_3 + \rho_3\tau_3}{C \frac{\phi_3 - L_{3,max}}{\phi_1 + \phi_2 + \phi_3 - L_{3,max}}} \end{aligned}$$

Dans ce cas, si  $\rho_1 + \rho_2 \rightarrow C$ , 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.

## 4. Contrôle équitable de réseau pour SCR

L'objectif de cette section est d'illustrer l'apport de la Classification de Service pour Ethernet, et plus particulièrement de l'ordonnancement WRR dans le contexte de SCR reposant sur un réseau partagé. Par rapport aux deux approches d'étude d'un SCR identifié dans [38], cet article détaille la stratégie « commande de réseau » permettant de répondre aux exigences requises par l'application (ici, la commande du système). De manière générale, il s'agit, relativement à une QoS souhaitée, de gérer au mieux les ressources de communication (réservation de ressources, contrôle dynamique de leur allocation, flexibilisation de l'offre de service, etc.). Dans le cadre d'un SCR à base d'Ethernet, les solutions suivantes visent à agir explicitement sur l'affectation des poids de l'ordonnancement WRR, soit implicitement sur la bande passante offerte à chaque trafic, et *in fine* sur les délais de bout en bout. À titre de comparaison, [35] présente une approche de commande en réseau basée sur la synthèse d'une commande robuste à partir de majorants des délais de bout en bout.

La particularité des approches suivantes est de développer un *co-design*, où les paramètres de Qualité de Service (ici, les délais de bout en bout) exigés du réseau sont exprimés en fonction d'indicateurs de Qualité de Performance (QdP) du SCR. La QdP d'un SCR peut être exprimé de différentes manières : stabilité du système, temps de réponse, dépassement maximal, marge de phase ou de retard, ou encore la position des pôles. Il s'agit ensuite de traduire cette QdP en QoS. Dans la suite, nous considérerons que la QoS exigé du réseau correspond à des retards maxima.

Enfin, dans la mesure où le réseau support au SCR est un réseau partagé avec d'autres applications non temps-réel, le but est aussi de maximiser le service offert au trafic non temps-réel. Ce dernier point reflète l'originalité des ordonnancements de type *fair queueing* pour les SCR : assurer une bande passante au trafic de fond, et éviter ainsi de se retrouver dans la configuration d'un réseau dédié. Deux approches d'optimisation des poids  $\phi_j$  sont alors proposées : soit de manière dynamique avec un objectif de flexibilité maximale, soit de manière statique avec une contrainte de robustesse.

### 4.1. Approche adaptative

Les résultats suivants s'appuient sur une plateforme expérimentale présentée dans [10]. Sur cette plateforme, 4 stations sont interconnectées *via* un commutateur à politique d'ordonnancement WRR (figure 16). Les deux premières stations exécutent des codes C/Posix simulant respectivement la partie process et commande du SCR. Le système étudié est le suivant :

$$P(s) = \frac{2}{(s+5)(s+0,2)}, \quad C(s) = \frac{K_p s + K_i}{s}$$

avec  $K_p = 0,5508$  et  $K_i = 0,4529$  les paramètres de commande minimisant l'erreur quadratique.

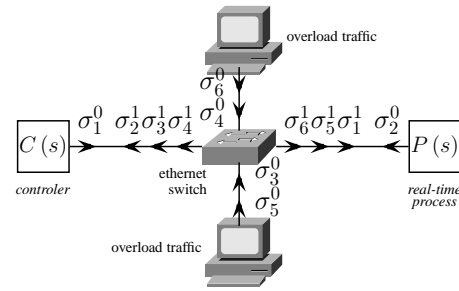


FIGURE 16. Plateforme expérimentale

Les deux autres stations sont responsables d'un flux de charge. Enfin, la plateforme est enrichie d'agents de mesure des délais uni-directionnels basés sur la synchronisation d'horloge selon le standard IEEE PTP 1588 et d'un agent de reconfiguration en-ligne des poids attribués à chaque classe du WRR. Le débit du réseau est fixé à 10 Mb/s et la charge de fond introduite par les deux stations est approximativement de 8 Mb/s. Comme précédemment, la figure 17(a) montre alors que si la bande passante peut apparaître en moyenne suffisante, de grands retards peuvent apparaître et se traduire en phase transitoire par des dépassements, voire engendrer une instabilité du système.

Compte tenu des indications de la figure 17(a), une première utilisation de l'ordonnancement équitable du WRR est proposée dans le but d'adapter les poids de chaque classe de trafic conformément à l'évolution du délai de commande. L'adaptation expérimentale proposée est la suivante : lorsque le délai des messages temps-réel augmente, le poids (allant de 1 à 255) de la classe temps réel est augmenté. La valeur des poids est alors gérée de deux manières. Tout d'abord, comme les paramètres du système impliquent un retard de 2 ms (plus petite constante de temps de la fonction de transfert  $P(s)$ ), si le retard observé est supérieur à ce seuil, alors le poids de la classe temps réel est incrémenté de 10 et celui de la classe basse priorité décrementé d'autant. Ensuite, dans le cas où les délais mesurés sont inférieurs à ce seuil, le processus est inversé de manière à offrir le service maximum au flux de fond. La figure 17(b) présente les résultats obtenus. Elle montre alors clairement que cette stratégie a permis de maîtriser les délais, et ainsi les dépassements.

Pour éviter les oscillations observées en fin de figure 17(b), une seconde stratégie basée sur le signal d'erreur de commande a été étudiée. La figure 17(c) donne les résultats. La procédure est la même que précédemment, excepté l'identification du seuil. Dans cette approche, le seuil est déterminé suivant que l'erreur est supérieure à 5 % du signal de référence. L'originalité ici consiste donc à consolider la commande du système et celle du réseau, ce qui permet comme observé à la figure 17(c), de limiter les dépassements et les oscillations. Au final, ces deux stratégies peuvent alors être rassemblées en une seule, c'est-à-dire où l'adaptation est basée à la fois sur l'erreur

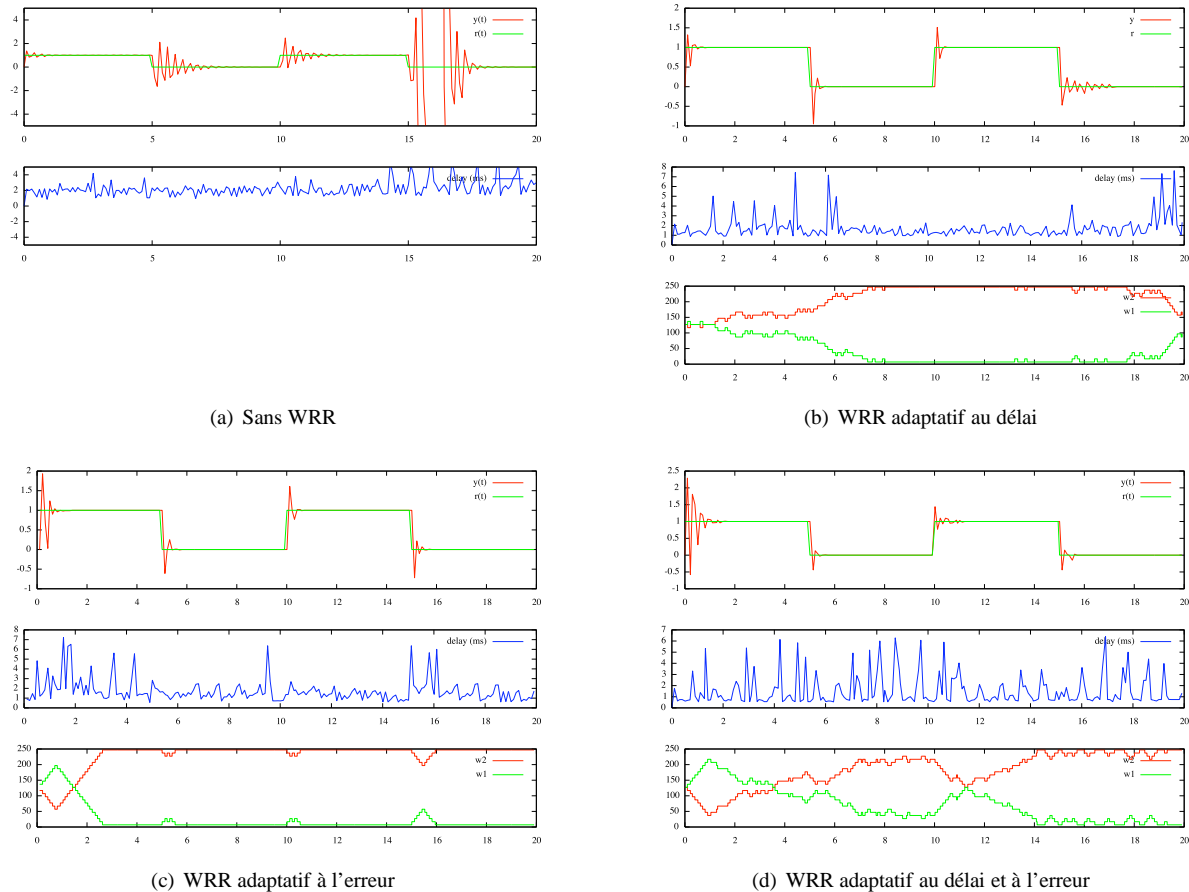


FIGURE 17. Adaptation en ligne des poids d'un WRR (le graphe du haut représente la consigne et la sortie du système, l'intermédiaire le délai et celui du bas, l'évolution des poids  $\omega_1$  et  $\omega_2$  en fonction du temps en s)

et le délai. Le délai apporte ici une indication de prédiction de l'état du réseau, permettant d'anticiper des phases de congestions progressives. Dans ce cas, la procédure est redéfinie telle que l'augmentation des poids du flux temporel intervient dès que l'un des deux seuils définis est atteint. La figure 17(d) présente les résultats.

Au final, l'adaptation en ligne des poids du WRR (c'est-à-dire la maîtrise de la bande passante) a ici permis :

- de réduire le dépassement maximal puisque l'on passe de près de 800 % dans le cas sans WRR à 94, 72 et finalement 44 % pour l'approche combinée
- de réduire les délais moyens (1,52 contre 2,27 ms initialement)
- tout en augmentant et garantissant la bande passante offerte aux autres flux circulant sur le réseau partagé support du SCR.

#### 4.2. Approche robuste

Dans le cas où le réseau ne permet pas de mettre en œuvre de procédure de reconfiguration en ligne des paramètres d'un ordonnanceur WRR, la stratégie de contrôle d'un SCR consiste à identifier une commande robuste, c'est-à-dire une configuration garantissant que la QoS

exigée par la commande du système sera satisfaite. Si l'on considère alors que cette QoS est qualifiée en tant que retard maximal, l'analyse basée sur le calcul réseau présentée au paragraphe 3, va alors permettre d'identifier la configuration des poids maximisant la bande passante offerte au trafic de fond sous la contrainte que les délais de bout en bout subis par le trafic de commande restent inférieurs à un seuil préalablement défini. L'étude suivante reprend alors la simulation du drone hélicoptère à quatre rotors [9] introduit au paragraphe 2 .

Une analyse de la commande a permis de déterminer un seuil maximum pour les retards de 5 ms. En utilisant la courbe de service identifiée par l'équation (5), et en appliquant le résultat concernant le majorant du délai de l'équation (6), il devient alors possible d'exprimer  $\omega_1$  et  $\omega_2$  en fonction de ce seuil de 5 ms. La détermination finale des poids passe alors par une recherche itérative des couples maximisant la bande passante  $\frac{\omega_2}{\omega_1 + \omega_2}$  pour un retard inférieur au seuil de 5 ms. Pour ce système, [9] identifie l'optimum ( $\omega_1 = 9, \omega_2 = 3$ ). Cette configuration tient compte de la concurrence d'accès des différents échanges induits par la commande du système avec le trafic de fond.

Afin d'illustrer l'impact de cette configuration, une simulation du SCR sous Matlab/Simulink – TrueTime a été

réalisée. Dans la mesure où la librairie TrueTime n’inclus pas le support de base des architectures Ethernet commuté à classification de service, un *patch* de la librairie a été développée (voir [9]). La figure 18 présente la réponse du drone pour cette configuration de poids WRR.

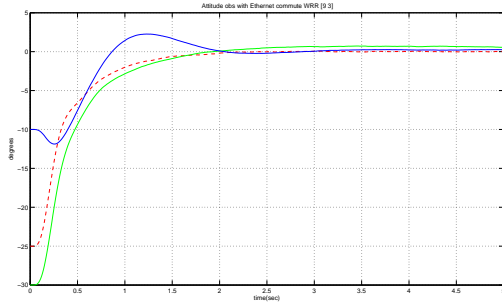


FIGURE 18. Stabilité recouvrée via WRR

Contrairement aux précédentes simulations de réseau partagé type Ethernet ou Ethernet commuté, la figure 18 montre que cette configuration présente un impact limité le plus faible possible sur la commande. En fait, on retrouve ici la même réponse qu’à la figure 8(b) dans le cas d’un réseau Ethernet 10 Mb/s dédiée ou encore à la figure 8(c) dans le cas d’un réseau CAN partagé. Par contre, le tableau 2 souligne l’intérêt de l’utilisation d’un ordonnancement équitable de type WRR par rapport à des politiques type priorité stricte en terme d’augmentation du service offert au trafic de fond. Dans le cas d’Ethernet, ceci est d’autant plus remarquable qu’il permet de profiter pleinement des larges débits en comparaison de ceux supportés par les réseaux locaux industriels traditionnels comme CAN.

TABLE 2. Comparaison de performances

Réseau	Temps de réponse (s)			Bande passante (Mb/s)
	roulis	tangage	lacet	
CAN	1.22	1.3	1.84	0.923
WRR	1.23	1.3	1.84	8.76

Il importe également de souligner que ces résultats pour WRR sont obtenus sans connaissance particulière concernant le trafic de fond dans la mesure où cette discipline de service agit comme un régulateur de trafic. Le trafic de fond pourra donc consommer l’intégralité de la bande passante qui lui est garanti. De plus, si la configuration ( $\omega_1 = 9, \omega_2 = 3$ ) peut sembler défavorable au trafic de fond, il faut rappeler que si l’ordonnanceur accède durant un cycle à une classe dont la file est vide, il passe automatiquement à la classe suivante. Cette configuration de poids ne vaut donc que pour le cas particulier où du trafic temps-réel est en attente.

## 5. Conclusion

Parmi les différentes stratégies d’adaptation des SCR aux influences de la Qualité de Service du réseau, la

« commande de réseau » est une approche pouvant permettre de prévenir toute dégradation de performances du système. Elle a pour objectif, relativement à une QdP souhaitée, de gérer au mieux les ressources de communication (réservation de ressources, contrôle dynamique de leur allocation, flexibilisation de l’offre de service, etc.). Chaque réseau apportant ses propres caractéristiques de contrôle d’admission, d’accès et de régulation, cela implique la nécessité de développer des approches spécifiques au réseau utilisé par le SCR.

Dans cet article, des stratégies de reconfiguration d’un ordonnancement de type *weighted round robin* sont proposés pour des SCR à base d’architectures Ethernet commutées. L’apport de cette technique d’ordonnancement équitable est de valider l’utilisation d’un réseau non déterministe comme Ethernet tout en favorisant le service rendu aux trafics non impliqués dans la commande du SCR. Cette configuration des poids, qu’elle soit adaptative ou robuste, passe par un *co-design* entre les disciplines de la commande et du réseau.

## Références

- [1] M. Alves, E. Tovar, and F. Vasques. 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), Jan. 2000.
- [2] C. Aubrun, E. Rondeau, J.-P. Georges, B. Brahimi, and F. Michaut. Deliverable 5-1 : Simulation of optimized network architecture. Projet STREP NeCST, May 2005.
- [3] J. Bennett and H. Zhang. Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking*, 5(5) :675–689, Oct. 1997.
- [4] A. Bouillard, B. Gaujal, S. Lagrange, and E. Thierry. Optimal routing for end-to-end guarantees using network calculus. *Performance Evaluation*, 65(11-12) :883–906, Nov. 2008.
- [5] C.-S. Chang. *Performance Guarantees in Communication Networks*. Telecommunication Networks and Computer Systems. Springer Verlag, 2000.
- [6] R. Cruz. A calculus for network delay, part i : network elements in isolation. *IEEE Transactions on Information Theory*, 37(1) :114–141, Jan. 1991.
- [7] R. Cruz. A calculus for network delay, part II : Network analysis. *IEEE Transactions on Information Theory*, 37(1) :132–141, Jan. 1991.
- [8] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *ACM SIGCOMM Computer Communication Review*, 19(4) :1–12, Sept. 1989.
- [9] I. Diouri, C. Berbra, J.-P. Georges, S. Gentil, and E. Rondeau. Evaluation of a switched ethernet network for the control of a quadrotor. In *16th Mediterranean Conference on Control and Automation (MED’08)*, pages 1112–1117, Ajaccio, France, June 2008.
- [10] I. Diouri, J.-P. Georges, and E. Rondeau. Adaptation of scheduling policy parameters to control networked systems. In *3rd International NeCST IFAC Workshop on Networked Control Systems : Tolerant to Faults (NECST’07)*, Nancy, June 2007.
- [11] J.-P. Georges, T. Divoux, and E. Rondeau. 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)*, pages 255–260, Taipei, Taiwan, Sept. 2004.
- [12] J. Grieu. *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, Sept. 2004.
- [13] Z. Huo, H. Fang, and C. Ma. Networked control system : state of the art. In *Fifth World Congress on Intelligent Control and Automation (WCICA'04)*, volume 2, pages 1319–1322, Hangzhou, China, June 2004.
- [14] IEEE Computer Society. 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, Dec. 1998.
- [15] IEEE Computer Society. Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 3 : Carrier sense multiple access with collision detection (csma/cd) access method and physical layer specifications. IEEE standard 802.3, Edition 2002, 2002.
- [16] IEEE Computer Society. Ieee standards for local and metropolitan area networks - virtual bridged local area networks. IEEE standard 802.1Q, Edition 2003, 2003.
- [17] IETF Intserv working group. Network element service specification template. Request For Comments RFC 2216, 1997.
- [18] J. Jasperneite, P. Neumann, M. Theis, and K. Watson. Deterministic real-time communication with switched ethernet. In *4th IEEE International Workshop on Factory Communication Systems (WFCS'02)*, pages 11–18, Västerås, Suède, Aug. 2002.
- [19] G. Juanole and I. Blum. 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)*, pages 217–232, Nancy, France, Apr. 1999.
- [20] G. Juanole and G. Mouney. Real time distributed systems : Qos and impact on the performances of process control applications. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, pages 1739–1746, Kyoto, Japan, July 2006.
- [21] S. Kubler. Continuité de service sur ethernet industriel. Master's thesis, Université Henri Poincaré, Nancy 1, Centre de Recherche en Automatique de Nancy, 2009.
- [22] J.-Y. Le Boudec and P. Thiran. *Network calculus, a theory of deterministic queueing systems for the Internet*. Lecture Notes in Computer Science. Springer Verlag, 2001.
- [23] K. Lee, S. Lee, and M. Lee. Qos-based remote control of networked control systems via profibus token passing protocol. *IEEE Transactions on Industrial Informatics*, 1(3) :183–191, Aug. 2005.
- [24] A. Lelevé and P. Fraisse. Teleoperation over ip network : Network delay regulation and adaptive control. *Journal of Autonomous Robots, special issue "Internet and online robots"*, 15(3) :225–235, Nov. 2003.
- [25] L. Lenzi, E. Mingozzi, and G. Stea. A methodology for computing end-to-end delay bounds in fifo-multiplexing tandems. *Performance Evaluation*, 65(11-12) :922–943, Nov. 2008.
- [26] F. Lepage, T. Divoux, and F. Michaut. Adaptation of control parameters based on qos monitoring. In *17th International Symposium on Mathematical Theory of Networks and Systems (MTNS'06)*, pages 1753–1758, Kyoto, Japan, July 2006.
- [27] S. Li, Z. Wang, and Y. Sun. Fundamental problems of networked control system from the view of control and scheduling. In *28th IEEE Conference of the Industrial Electronics Society (IECON'02)*, volume 3, pages 2503–2508, Séville, Espagne, Nov. 2002.
- [28] R. Luck and A. Ray. An observer-based compensator for distributed delays. *Automatica*, 26(5) :903–908, 1990.
- [29] J. Nilsson. *Real-time control systems with delays*. PhD thesis, Lund Institute of Technology, Department of Automatic Control, Feb. 1998.
- [30] A. Parekh. *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, Feb. 1992.
- [31] A. Parekh and R. Gallager. 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, June 1993.
- [32] J.-P. Richard. Time-delay systems : an overview of some recent advances and open problems. *Automatica*, 39(10) :1667–1694, Oct. 2003.
- [33] J.-P. Richard and T. Divoux, editors. *Systèmes commandés en réseau*. Traité IC2. Hermès Lavoisier, Feb. 2007.
- [34] Y. Tipsuwan and M.-Y. Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11 :1099–1111, 2003.
- [35] N. Vatanski, J.-P. Georges, C. Aubrun, E. Rondeau, and S.-L. Jämsä-Jounela. Networked control with delay measurement and estimation. *Control Engineering Practice*, 17(2) :231–244, Feb. 2009.
- [36] G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, 10(3) :438–446, May 2002.
- [37] J. Yook, D. Tilbury, and N. Soparkar. A design methodology for distributed control systems to optimize performance in the presence of time delays. *International Journal of Control*, 74(1) :58–76, Jan. 2001.
- [38] S. Zampieri. Trends in networked control systems. In *Proceedings of the 17th IFAC World Congress*, pages 2886–2894, Séoul, Corée du Sud, July 2008.
- [39] W. Zhang, S. Branicky, and S. Phillips. Stability of networked control systems. *IEEE Control Systems magazine*, 21 :84–89, Feb. 2001.