

Classification des délais pour les Systèmes Contrôlés en Réseau à ordonnancement WRR

Idriss DIOURI, Vincent BOMBARDIER, Jean-Philippe GEORGES, Eric RONDEAU

Centre de Recherche en Automatique de Nancy,
UMR 7039 – Nancy-Université, CNRS
2, avenue de la forêt de Haye, 54516 Vandœuvre, France.
Prénom.Nom@cran.uhp-nancy.fr

Résumé— Dans les systèmes distribués temps-réel, on distingue les applications ayant des contraintes souples et celles ayant des contraintes dures. Dans cet article, on utilise une architecture Ethernet commutée implémentant le protocole d'ordonnancement WRR pour interconnecter des applications temps-réel souple. L'objectif de cette étude est de contrôler les délais de bout en bout pour éviter qu'ils soient au-delà d'un certain seuil. Dans ce sens, un mécanisme de classification par règles linguistiques floues a été utilisé pour identifier les délais de l'application temps-réel en fonction des poids de l'ordonnancement WRR. Le modèle du comportement ainsi obtenu a servi à mettre en place une méthode de reconfiguration assurant des délais acceptables pour l'application temps-réel tout en maximisant la bande passante garantie aux autres flux.

Mots-clés— SCR, systèmes temps-réel, commande du réseau, ordonnancement, logique floue, identification, règles linguistiques, classification.

I. INTRODUCTION

Un système distribué se compose de plusieurs sous-systèmes autonomes communiquant entre eux via des réseaux. Ces systèmes distribués peuvent à leurs tours être interconnectés entre eux voire même au réseau Internet pour permettre des applications telles que la E-maintenance, etc. Comparés aux systèmes centralisés, les systèmes distribués ont permis d'améliorer les systèmes de contrôle/commande industriels en termes de réduction des coûts de câblage, de flexibilité et aussi en termes d'aide au diagnostic et à la maintenance. Les réseaux de communication qu'utilisent les systèmes distribués génèrent des retards et des pertes de données. Ces perturbations dépendent fortement du réseau utilisé (type du réseau, topologie, protocole de communication et politique d'ordonnancement, etc) et de la charge du réseau. Ces perturbations doivent être prises en compte dans la conception des systèmes et nécessitent d'être maîtrisées de telle sorte à stabiliser le système.

Le taux de pertes des paquets ainsi que les délais induits par les réseaux varient aléatoirement au cours du temps en fonction de la complexité de l'architecture du système distribué et à cause de la nature de la charge traversant le réseau. Cette variation imprévisible des délais complique leur prise en compte dans la conception des systèmes. Ainsi, dans le contexte des Systèmes Contrôlés en Réseau SCRs [2], quatre types d'approches sont investiguées. La première approche consiste à adapter la commande relativement aux performances du système « approche contrôle sur le réseau ». La deuxième, consiste à adapter le réseau pour répondre aux besoins applicatifs « approche commande du réseau »

[3][4][5]. Une troisième approche consiste à concevoir le système en intégrant les deux approches précédentes, c'est l'approche de co-conception [6][7][8]. Dans [6] et [8], l'accent notamment a été mis sur le rôle déterminant des outils de co-simulation pour l'étude de cette approche. La quatrième approche considère le système distribué comme étant un système multi-agents [9][10]. Dans cette dernière approche il s'agit d'utiliser des contrôleurs distribués, ce qui permet de réduire à la fois la charge du réseau et l'impact de la perte de paquets sur le système global. L'intérêt dans [10] s'est porté sur l'étude des architectures commutées ayant des délais variables bornés. Dans cet article, les travaux se situent au niveau de l'approche « commande du réseau ».

Dans les SCRs, on peut distinguer entre deux types d'applications, celles ayant des contraintes temps-réel dures et celles ayant des contraintes temps-réel souples. Dans [11] et [12], l'étude s'est portée sur les applications temps-réel dur. Le calcul réseau [13] a alors été appliqué à une architecture Ethernet commutée implémentant le protocole d'ordonnancement Weighted Round Robin (WRR) [14][15] pour identifier le délai maximal de bout en bout. Le délai maximal identifié dans [11] tient compte des poids de l'ordonnancement WRR, et a permis d'identifier le réglage des poids pour garantir les contraintes temporelles dures tout en maximisant la bande passante du trafic de fond. Tant que les délais subis par le trafic temps-réel respectent les exigences temporelles, le poids du trafic de fond, et implicitement la bande passante allouée, augmenteront. Néanmoins, cette majoration n'est représentative que du pire cas et peut engendrer un certain pessimisme (l'augmentation du poids du trafic de fond restera limitée compte tenu de la sur-majoration).

Dans cet article, l'étude portera sur les applications temps-réel souple pour lesquelles l'utilisation de méthodes probabilistes pour l'identification du délai s'avère difficile à cause des nombreuses non linéarités du système. Ainsi l'identification sera effectuée en se basant sur une définition qualitative du délai séparée en trois tendances principales: Délai Parfait, Délai Intermédiaire et Délai Non-Toléré. Cette définition qualitative repose sur la modélisation du délai par des variables linguistiques floues [32].

II. CONTEXTE DE L'ETUDE

A. Commande du réseau

Dans les SCRs, le contrôle du réseau consiste à adapter le réseau pour satisfaire les contraintes applicatives. Afin de régler les paramètres, il est important de connaître les

caractéristiques du trafic temps-réel (par exemple : le temps d'inter-arrivée des trames, la taille de la trame, etc.).

Dans l'approche "commande du réseau", certaines études sont basées sur le contrôle de la congestion pour adapter le débit d'entrée à la bande passante disponible [3][16]. Dans [3], une méthode basée sur le contrôle de congestion a été proposée dans le but de répartir la bande passante disponible dans un réseau à grande échelle entre différents flux concurrents. [3] s'est basé sur la modélisation du retour du signal de congestion aux sources et a montré que le problème de contrôle de congestion dans ces réseaux à grande échelle peut être traité en décomposant le problème de telle manière à ce qu'il soit géré de façon décentralisée par chacune de ces sources. Toutefois, [3] a considéré que les sources réagissent instantanément aux débits qui leur sont alloués par le réseau. Or il existe un décalage qui peut s'avérer critique quant à la stabilité de tout le système. Un des problèmes majeurs dans l'étude du contrôle de congestion est la conception de contrôleurs qui sont stables même en présence de délais variables. Certains auteurs ont étudiés la stabilité des algorithmes de congestion en présence de délais [17][18].

Dans [19], une technique de lissage de trafic qui permet de fournir des garanties de délai statistiques sur Ethernet a été proposée. Un mécanisme d'adaptation nommé « Harmonic-Increase and Multiplicative-Decrease » (HIMD) qui réagit avec la détection de collision et cela sur une période de temps spécifique a été proposé [20]. Dans [21], une méthode de lissage basée sur la logique floue a été proposée et il a été montré que les délais aller-retour obtenus avec l'approche floue sont inférieurs à ceux obtenus avec la méthode HIMD [20]. Afin d'optimiser leur méthode de lissage floue les algorithmes génétiques ont été utilisés [22]. La méthode de lissage de trafic floue proposé par [22] a permis à Ethernet de fournir une borne statistique du délai. Dans [4], la loi de contrôle de congestion a été réalisée en se basant sur le principe de Smith, celle-ci garantit la stabilité des files du réseau et une utilisation complète des liens du réseau. Les contrôles du flux et de la congestion TCP peuvent être modélisés comme un système à retard et il a été montré que le principe de « self-clocking » (quand un paquet sort, un nouveau entre dans le réseau), correspond à l'implémentation d'un prédicteur de Smith plus un proportionnel simple [23]. Sur le sans fil, le contrôle de congestion et le problème d'allocation de ressources sont de plus en plus considérés [24][25].

L'adaptation de la Qualité de Service a aussi été un axe d'étude dans l'approche « commande du réseau ». L'impact de la QoS sur la stabilité de la commande du process dans les SCRs a été montré [5]. L'influence de la QoS a été étudiée en analysant l'ordonnancement (avec et sans priorité) [5]. La priorité stricte et le Earliest Deadline First (EDF) [26] sont aussi des politiques d'ordonnancement classiques utilisées dans l'approche « contrôle du réseau ». Le calcul réseau a permis d'identifier le délai maximal de bout en bout dans le cas d'une architecture Ethernet commutée implémentant le protocole d'ordonnancement WRR [11]. Cette identification du délai maximal dépend des poids du commutateur [11], ainsi pour un seuil défini il est possible de définir le couple de poids qui garantit que le délai maximal soit inférieur à ce seuil tout en maximisant la bande passante allouée au trafic de fond. L'identification basée sur le calcul réseau donne de bons résultats dans le cas des applications temps-réel dur mais

celle-ci s'avère pessimiste dans le cas d'applications temps-réel souple. Dans ce papier, l'intérêt se porte sur les applications qui ne sont pas fortement contraintes temporellement et l'identification des délais est faite en se basant sur la logique floue. Le modèle ainsi établi pourra servir ultérieurement à la synthèse d'une régulation du délai.

B. Le réseau Ethernet commuté

Le réseau Ethernet est de plus en plus utilisé dans les applications industrielles. Le protocole CSMA/CD utilisé par Ethernet (standard IEEE 802.3) introduit un accès non déterministe dû aux collisions. Pour éviter ce problème, le réseau Ethernet commuté combiné avec le mode full-duplex tel qu'il est défini dans le standard 802.1D [28] est utilisé. De plus, les commutateurs Ethernet donnent la possibilité d'implémenter la classification de service CoS. Les protocoles d'ordonnancement habituellement implémentés dans les commutateurs Ethernet sont la priorité stricte et le WRR [14][15]. Le choix de la politique d'ordonnancement adéquate est fondamental pour l'optimisation des ressources du réseau et pour la garantie des contraintes temps-réel. Dans le cas de cet article, l'étude porte sur le WRR. L'ordonnanceur WRR donne une priorité à chacun des flux. Ces priorités serviront à définir l'ordre du cycle (du flux ayant la plus haute priorité à celui ayant la plus faible priorité). Le nombre de trames qui seront transmises pour une file dans un cycle est borné par son poids. Lorsque l'ordonnanceur s'aperçoit qu'une file est vide, celui-ci passe immédiatement à la file suivante.

C. Système étudié

L'objectif est d'identifier le comportement de l'ordonnanceur WRR implanté dans une architecture Ethernet commutée. Pour cela, on cherchera à définir un modèle du délai en fonction des caractéristiques du trafic et du réseau : débit, charge des liens, longueur et temps d'inter-arrivée des trames, et ordonnanceur. Dans notre étude, le seul paramètre commandable correspond au poids alloué à chaque classe pour la politique WRR. Cette identification de délai est basée sur une série de simulations numériques. Cette démarche qui s'appuie sur une identification hors-ligne se justifie par le fait qu'il est difficile de réaliser des mesures de délais unidirectionnels en exploitation. Les simulations ont été effectuées en utilisant le logiciel OPNET [29] qui permet de simuler le comportement des réseaux. Le scénario de simulation correspond à l'observation du délai entre deux stations A et B interconnectées à un commutateur WRR. Une troisième station C raccordée à ce même commutateur permet ensuite d'introduire une charge variable à destination de B. On notera w_1 le poids alloué au flux AB et w_2 le poids alloué au flux CB.

Les liens sont à 10 Mbit/s. La taille des paquets du trafic temps-réel (flux AB) est 125 octets et un paquet est envoyé chaque 10 ms. L'inter-arrivée des paquets du trafic de fond (flux CB) suit une loi exponentielle. Le comportement du système a été étudié avec des charges dont les moyennes sont les suivantes : 5 Mbit/s, 6 Mbit/s et 7 Mbit/s basées sur des trames pleines. Pour chaque configuration de poids du commutateur (entrées qui permettent d'agir en temps-réel en termes de commande du réseau), trente simulations (avec à chaque fois un tirage de variables aléatoires « seed » différent) ont été effectuées afin que notre estimateur soit fiable [29]. Et pour chaque moyenne de charge (5 Mbit/s, 6 Mbit/s et

7 Mbit/s) 266 configurations de poids ont été simulées. La durée de chaque simulation est de 5 minutes. Notre étude concerne les applications temps-réel souple. L'identification du délai sera effectuée en utilisant la borne supérieure des maximas moyens obtenus pour un intervalle de confiance de 90 %. Une étape préalable à la fuzzification des données a consisté à définir 3 classes de délais imposées par le système de contrôle/commande utilisé dans notre exemple. Les 3 classes adaptées à l'application étudiée sont les suivantes :

- 1ère classe : Délai Parfait (DP) pour des $D(w_1, w_2, \text{moyenne de charge})$ inférieurs ou égaux à 0,7 ms.
- 2ème classe : Délai Intermédiaire (DI) pour des $D(w_1, w_2, \text{moyenne de charge})$ supérieurs strictement à 0,7 ms et inférieurs ou égaux à 2,5 ms.
- 3ème classe : Délai Non Tolérés (DNT) pour des $D(w_1, w_2, \text{moyenne de charge})$ supérieurs strictement à 2,5 ms.

En fonction de ces 3 classes et des résultats des simulations, il est possible de déterminer les règles d'inférence qui régissent notre système. La méthodologie suivie pour déterminer ces règles d'inférence ainsi que le résultat de l'identification sont présentés dans la section suivante.

III. IDENTIFICATION DU JEU DE REGLES FLOUES

Pour identifier le jeu de règles floues synthétisant le comportement du modèle, les travaux s'appuient sur des techniques issues du domaine de la reconnaissance de formes en traitement d'images. Les poids w_1 et w_2 sont assimilés à des primitives en entrée d'un classificateur (FRC : Fuzzy Rule Classifier) [30], basé sur un mécanisme de règles linguistiques floues. Ce classificateur est adapté au contexte applicatif. En effet, il possède une bonne capacité de généralisation comme l'ont montré les comparaisons effectuées avec d'autres classificateurs (K-NN, RdN, SVM) [1]. Ce point est important au vu du nombre de simulations nécessaires pour obtenir un échantillon pour l'identification. Il rend compte également de la gradualité d'appartenance aux différentes catégories de délais en sortie, compatible avec le concept de contrainte de temps « souple ». Enfin, le principal avantage de ce classificateur dans le cas d'étude proposé concerne la possibilité d'interprétation du mécanisme de raisonnement sous la forme de règles linguistiques.

Cette méthode de reconnaissance floue est un mécanisme par apprentissage supervisé qui se décompose en trois parties : la fuzzification des paramètres du vecteurs caractéristiques, la génération des règles et l'ajustement du modèle. Seule la partie apprentissage est réellement utilisée ici, puisque le jeu de règles obtenu ne sert pas à créer un moteur d'inférences floues pour une reconnaissance, mais est exploité pour réaliser la commande du réseau.

A. Fuzzification

L'étape de fuzzification vise à traduire une variable numérique en une variable linguistique. Une variable linguistique [32] est définie par un triplet (V, X, Tv) où :

- V est une variable (w_1, w_2) définie sur un référentiel X
- X est l'univers de discours (domaine de variation de V)
- Tv est le vocabulaire choisi pour définir linguistiquement les valeurs de V (Petit, Moyen, Grand, etc.).

L'ensemble $Tv = \{A1, A2, \dots\}$, fini ou infini, contient les sous-ensembles flous normalisés de X qui sont utilisables pour caractériser V . Chaque sous-ensemble flou, A_i , est défini par un degré d'appartenance $\mu_{A_i}(x)$. Chaque partie de cette décomposition est associée à un mot du langage naturel. Par exemple, le poids w_1 de l'ordonnanceur (WRR) peut être « Grand », « Moyen » ou « Petit » dans une décomposition en 3 termes. Cette variable est caractérisée par un vecteur composé de trois degrés d'appartenance: $[\mu_{Faible}(x), \mu_{Moyenne}(x), \mu_{Forte}(x)]$. La figure 1 donne un exemple de décomposition de la variable w_1 en 3 termes équi-répartis.

Cette étape de fuzzification détermine le nombre de termes de la variable X à fournir aux prémisses des règles floues. Ces différents termes sont choisis en lien avec le vocabulaire de l'expert du domaine et sont généralement définis de façon empirique. Pour simplifier le réglage, les utilisateurs industriels de la méthode préfèrent une partition égale des termes ce qui conduit généralement à utiliser plus de termes qu'il n'en faut. Cependant, quand le nombre de termes augmente, le nombre de règles augmente également et donc la complexité du système. C'est pourquoi, pour pouvoir interpréter le système de règles obtenu, les termes doivent être adaptés aux données d'entrée. Des méthodes peuvent être utilisées pour effectuer une adaptation automatique de la fuzzification. Elles sont classiquement basées sur l'utilisation d'algorithmes génétiques [33] ou d'algorithmes de clustering [34]. L'inconvénient de ces méthodes réside dans le besoin de lots d'apprentissage conséquents.

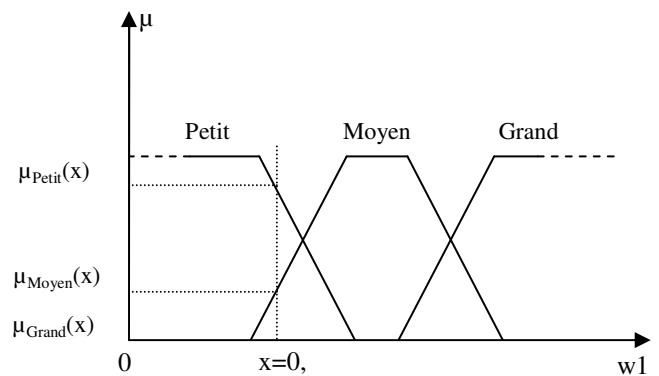


Figure 1 : Fuzzification du poids w_1 en trois termes avec des courbes trapézoïdales

La méthode d'auto-fuzzification utilisée pour les essais exposés dans le paragraphe E repose sur l'étude de la typicalité du vecteur caractéristique en fonction des classes de sortie [35]. A partir de $T(V)$, un coefficient de corrélation (Corr) et de cross-corrélation (Xcorr) est calculé pour chaque classe de sortie. Ensuite, à partir du rapport $Corr/Xcorr$, qui caractérise la ressemblance intra-classe, on détermine le nombre de termes ainsi que leur position [31].

B. Génération des règles floues

La deuxième partie définit les règles floues de type "SI... ALORS...". Chaque règle décrit la façon dont les défauts sont perçus par le système. Il existe deux grandes catégories de règles floues. Les règles conjonctives regroupent les règles à possibilité et les règles anti-graduelles. Les règles implicatives regroupent quant à elles, les règles à certitude et les règles

graduelles. Les règles conjonctives peuvent être obtenues à partir de lots de données numériques tandis que les règles implicatives sont généralement utilisées dans les sciences cognitives [36]. C'est pourquoi, dans la présente étude, le choix des règles conjonctives a été fait [37]. Chaque règle est activée en parallèle et un opérateur de disjonction est utilisé pour agréger les conclusions partielles dans le but de fournir la décision finale.

Le classificateur que nous avons utilisé est basé sur l'algorithme d'Ishibuchi qui propose une étape automatique de génération de règles [38]. Il existe de nombreuses méthodes permettant d'obtenir automatiquement le jeu de règles à partir du lot de données [39], mais l'algorithme de Ishibuchi a l'avantage d'être simple et très efficace [30]. En outre, il correspond à un mécanisme d'inférence de type Larsen, qui est préférable à celui de Mamdani dans le cas où il y a plusieurs prémisses [40]. La version itérative de l'algorithme est utilisée pour notre étude [27], les règles sont obtenues en favorisant la règle de réponse maximum, à partir d'un lot de défauts préparé par un expert du domaine.

C. Ajustement du modèle

L'étape d'ajustement correspond à la partie itérative de l'algorithme. En fait, cela consiste à ajuster le découpage de l'espace de représentation des attributs d'entrée. Cet ajustement est effectué à partir des résultats donnés par le jeu d'échantillons d'apprentissage. L'algorithme génère un premier modèle et un coefficient de confiance CF est calculé à partir du degré de vérité de chaque règle.

Si le taux de classification est inférieur à un seuil défini par l'utilisateur, la partie itérative ajuste le modèle en modifiant le coefficient de confiance CF affecté à chaque règle. Il est augmenté si l'échantillon confirme la règle, diminué dans le cas contraire.

En sortie, le système d'inférences floues délivre autant de degrés de possibilité que de classes de sortie, signifiant la possibilité que les poids donnés conduisent à un délai de type « Parfait », « Intermédiaire » ou « Non Toléré ». Ainsi, trois degrés de possibilité sont associés au délai concerné. La décision finale est prise à l'aide d'un opérateur de disjonction flou (Max). Il est à noter que ces possibilités ne sont pas complémentaires.

D. Résultats de classification

La méthode a été appliquée sur l'ensemble de 798 mesures prises pour une charge de 5 Mbit/s, 6 Mbit/s et 7 Mbit/s et un modèle général a été généré, indépendant de la charge. La table 1 donne les taux de reconnaissance obtenus à l'aide du jeu de règles après apprentissage. Bien que les taux soient significatifs du bon comportement du modèle obtenu (87.5%), il est à noter que les réglages du FRC n'ont pas été choisis pour fournir des taux optimaux. Notre objectif a été d'obtenir les règles principales du système. Le manque d'échantillons pour certaines classes de délais explique les faibles taux obtenus pour les délais intermédiaires et non-tolérés

TABLE 1
TAUX DE RECONNAISSANCE

| Lot de données | 5M | 6M | 7M | Total |
|----------------|--------------|--------------|--------------|--------------|
| DP | 218 99.5% | 218 99.5% | 218 99.1% | 653 99.4% |
| DI | 36 22,2% | 17 23.5% | 14 14,3% | 67 28.4% |
| DNT | 12 58.3% | 31 35.5% | 35 34.3% | 78 38.5% |
| Total | 89.1% | 87.2% | 86.1% | 87.5% |

E. Synthèse des règles floues - Modèle

L'obtention du jeu de règles se fait en analysant la matrice C obtenue lors de la phase itérative. La table 2 montre le jeu de règles obtenues après apprentissage sur le lot de données. Les règles associées à un CF proche de 1 sont les règles « sûres », confirmées par une grande majorité des échantillons du lot d'apprentissage. A l'inverse, les règles associées à un CF faible sont des règles correspondant à des zones d'ambiguïté du modèle.

TABLE 2
REGLES GENEREES

| Règles n° | Prémisses $w_2 - w_1$ | Sortie | CF |
|-----------|--------------------------|--------|--------|
| Rule_0 | P - P | DP | 0,5760 |
| Rule_1 | P - M | DP | 0.9819 |
| Rule_2 | P - G | DP | 0,9999 |
| Rule_3 | M - P | DNT | 0.7789 |
| Rule_4 | M - M | DP | 0.7149 |
| Rule_5 | M - G | DP | 0.9999 |
| Rule_6 | G - P | DNT | 0.8896 |
| Rule_7 | G - M | DI | 0.4479 |
| Rule_8 | G - G | DP | 0.9169 |

Le FRC a été réglé de façon à ce que les règles générées aient le CF le plus élevé possible, au détriment de taux de reconnaissance dans certain cas (voir table 1). Ainsi, les règles identifiées correspondent aux règles principales de comportement du système.

La règle 2 s'interprète par exemple de la façon suivante :

SI l'entrée w_2 est PETIT (P) ET l'entrée w_1 est GRAND (G) ALORS la sortie Délai est PARFAIT (DP) avec un coefficient de confiance égal à 0,9999.

On peut donc en déduire que si l'on veut un délai « Parfait » en sortie du réseau, il convient de régler les poids w_1 et w_2 de l'ordonnanceur comme, respectivement, Petit et Grand.

IV. RECONFIGURATION BASEE SUR LES REGLES FLOUES

A. Mise en place de la stratégie de reconfiguration des poids

Dans cette partie, l'objectif est de mettre en place une reconfiguration basée sur le modèle flou obtenu dans la partie précédente. Notre objectif étant essentiellement de montrer le bien fondé de notre méthode, nous avons choisi un mécanisme de reconfiguration simple. Cette reconfiguration se fera en ayant comme entrée le délai du trafic temps-réel et comme sortie les poids du WRR. Ainsi, à l'obtention d'un Délai

Parfait, la combinaison de poids qui sera affectée est ($w_1=16$, $w_2=119$). A l'obtention d'un Délai Non Toléré la combinaison de poids qui sera affectée est ($w_1=141$, $w_2=78$). La règle d'inférence utilisée dans ce cas (w_1 Grand ET w_2 Moyen Alors Délai Parfait) a un taux de confiance approchant 1. Si le Délai est Intermédiaire, l'ordonnanceur WRR ne réagit pas. Dans cette étude, la classification des délais est utilisée pour la conception d'un commutateur intelligent. Il est ainsi proposé d'implanter au sein d'un commutateur un module d'observation des délais de traversée de chaque file WRR d'un port de sortie. Un algorithme permettant de faire varier les poids dynamiquement en fonction des délais, a alors été codé et mis en œuvre en simulation sous OPNET.

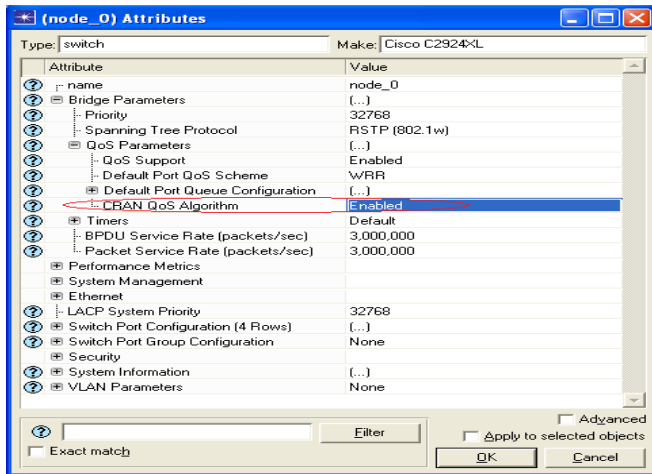


Figure 2 : Commutateur permettant de mettre en place un WRR dynamique sur OPNET

Dans la figure 2, la partie entourée correspond à l'interface OPNET qui a été ajoutée au commutateur pour activer notre algorithme.

B. Résultats obtenus par la reconfiguration des poids

Les figures 3 et 4 permettent de comparer les résultats obtenus avec la reconfiguration avec ceux obtenus sans classification de service des trafics en utilisant la même base de comparaison (mêmes paramètres du trafic de fond et du trafic temps-réel). La figure 3.b. montre l'évolution de la charge de fond.

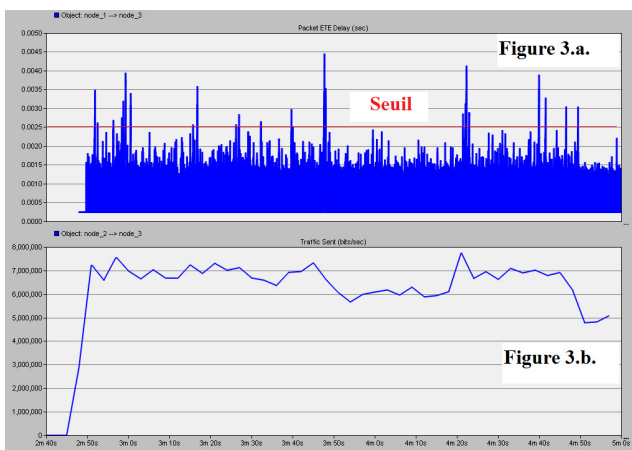


Figure 3 : Délais du trafic temps-réel (sans WRR)

Ce type de charge s'applique dans les deux cas avec et sans adaptation WRR, et varie autour de 5 et de 7 Mbit/s. La charge de fond est en compétition avec le trafic temps-réel, ce qui induit des dépassements de seuil. La figure 3.a. montre que l'on obtient 95 dépassements (souvent par rafale) quand on est en configuration FIFO, c'est-à-dire sans WRR.

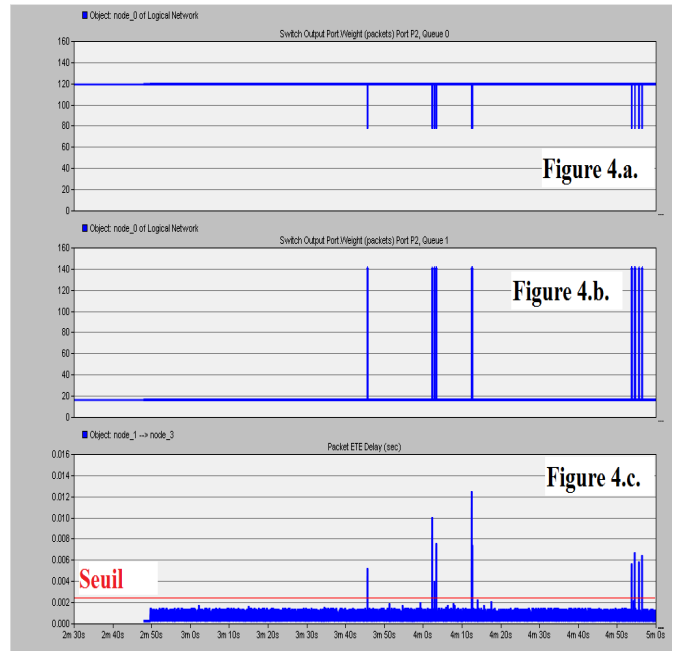


Figure 4 : Délais du trafic temps-réel (avec WRR dynamique)

La figure 4.a. donne l'évolution de w_2 en fonction du comportement du délai et la figure 4.b. donne celle de w_1 . On remarque que la configuration initiale est $w_2=119$ et $w_1=16$ qui permet d'obtenir la classe de Délais Intermédiaires. La figure 4.c. montre les délais du trafic temps-réel. On observe que lorsque ce délai dépasse le seuil admissible (2,5 ms), les poids w_1 et w_2 changent de façon à faire repasser les délais en dessous du seuil. On voit bien que notre proposition s'adapte bien à des applications temps-réel souple car les délais peuvent dépasser ponctuellement le seuil critique, mais que le contrôleur réseau agit pour les faire redescendre en dessous de celui-ci le plus rapidement possible. Par rapport à une solution sans WRR, on réduit le nombre de dépassement de seuil à 11. On remarque aussi que l'adaptation WRR réduit le nombre de dépassements successifs qui est le facteur le plus pénalisant pour maintenir la stabilité du système de contrôle/commande. Ceci est à mettre en relation avec la qualité de la classification des délais, c'est-à-dire avec la qualité de l'identification basée sur la logique floue.

V. CONCLUSION

Ce papier présente une méthodologie d'identification des délais dans les architectures Ethernet commutées. L'identification est basée sur la logique floue et permet de formuler le délai de traversée en fonction de l'ordonnancement WRR et des poids associés à chaque classe. Dans cet article, un simple mécanisme de reconfiguration a été utilisé pour illustrer l'intérêt de cette approche. Cette reconfiguration doit permettre de satisfaire les contraintes des applications temps-réel souple tout en maximisant la bande

passante garantie aux flux non contraints temporellement. En effet, la bande passante offerte à un flux par un ordonnancement WRR correspond simplement au rapport entre le poids alloué au flux et la somme des poids des différentes classes. Au final, cet article présente une méthodologie qui est intéressante dans la mesure où elle fournit de bons résultats et aussi car elle peut être généralisée pour d'autres types d'applications temps-réel. Dans de futurs travaux, nous envisageons de mettre en place un mécanisme de régulation flou afin d'exploiter au mieux le degré de possibilité relatif au délai.

REMERCIEMENTS

Ce travail a pu être réalisé en partie grâce au soutien d'OPNET Technologies Inc. à travers le programme « Teaching with OPNET ».

RÉFÉRENCES

- [1] Bombardier V., Schmitt E., Charpentier P., A fuzzy sensor for color matching vision system, *Measurement*, Elseviers, Vol 42, n° 2, pp 189-201, 2009.
- [2] Richard J-P. et Divoux T. *Systèmes commandés en réseau*, Edition Lavoisier, Hermès sciences publications, 2007.
- [3] Kelly F., Maulloo A. et Tan D. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.
- [4] Mascolo, S. Congestion control in high-speed communication networks using the Smith principle. *Automatica*, 35(12), 1999.
- [5] Juanole G. et Mouney G. Real time distributed systems: QoS and impact on the performances of process control applications. In *Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.
- [6] Branicky M. S., Liberatore V., et Phillips S. M. Networked Control System Co-Simulation for Co-Design. In *Proc. of the American Control Conference*, Denver, USA, vol. 4, pp. 3341--3346, June 2003.
- [7] Sename O., Simon D. et Robert D. Feedback Scheduling for Real-Time Control of Systems with Communication Delays. In *IEEE Conference ETFA'03*, Lisbonne, Portugal, 2003.
- [8] Brahimi B, Rondeau E. et Aubrun C. Integrated Approach based on High Level Petri Nets for Evaluating Networked Control Systems. In *16th Mediterranean Conference on Control and Automation (MED'08)*, Ajaccio, Corsica, France, 25-27 juin 2008.
- [9] Zampieri S. Trends in networked control systems. In *17th IFAC World Congress*, pages 2886-2894, Seoul, South Korea, July 2008.
- [10] Xiao F. et Wang L. State consensus for multi-agent systems with switching topologies and time-varying delays. *International J. Control*, 79(10):1277–1284, 2006.
- [11] Diouri I., Georges J.-P., et Rondeau E. Accommodation of delays for NCS using classification of service. In *IEEE International Conference on Networking, Sensing and Control (ICNSC'07)*, London, United Kingdom, Apr. 2007, pp. 410–415.
- [12] Diouri I., Berbra C., Georges J.-P., Gentil S. et Rondeau E. Evaluation of a switched Ethernet network for the control of a quadrotor. In *16th Mediterranean Conference on Control and Automation (MED'08)*, Ajaccio, Corsica, France, 25-27 juin 2008.
- [13] Le Boudec J.-Y. et Thiran P. *Network calculus, a theory of deterministic queueing systems for the Internet*. Lecture Notes in Computer Science. Springer Verlag, 2001.
- [14] Demers A., Keshav S. et Shenker S. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM Computer Communication Review*, vol. 19, no. 4, Sept. 1989, pp. 1–12.
- [15] Katevenis M., Sidiropoulos C. et Courcoubetis C. Weighted round robin cell multiplexing in a general purpose ATM switch chip. *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 9, pp. 1265–1279, Oct. 1991.
- [16] Floyd S. et Fall K. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4), 458–472, 1999.
- [17] Alpcan T. et Basar T. A globally stable adaptive congestion control scheme for internet-style networks with delay. *IEEE/ACM Trans. on Networking*, 13:1261– 1274, 2005.
- [18] Ying L., Dullerud G. E. et Srikant R. Global stability of internet congestion controllers with heterogeneous delays. *IEEE/ACM Trans. on Networking*, 14:579–590, 2006.
- [19] Kweon S., Shin K. G. et Zheng Q. Statistical real-time communication over ethernet for manufacturing automation systems. In *Proc. 5th IEEE Real-Time Technology and Application Symp.*, Vancouver, BC, Canada, Jun. 1999.
- [20] Kweon S., Shin K. G. et Workman G. Achieving real-time communication over Ethernet with adaptive traffic smoothing. In *Proc. Sixth IEEE Real-Time Technology and Applications Symp.*, RTAS 2000, Washington, DC, Jun. 2000, pp. 90–100.
- [21] Carpenzano A., Caponetto R., Lo Bello L. et Mirabella O. Fuzzy traffic smoothing: An approach for real-time communication over Ethernet networks. In *Proc. 4th IEEE Workshop on Factory Communication Systems, WFCS 2002*, Västerås, Sweden, Aug. 2002, pp. 241–248.
- [22] Lo Bello L., Kaczynski G.A. et Mirabella O. Improving the Real-Time Behaviour of Ethernet Networks Using Traffic Smoothing. *IEEE Transactions On Industrial Informatics*, Vol.1, N° 3, August 2005.
- [23] Mascolo S. Modeling the Internet congestion control using a Smith controller with input shaping. *Control Engineering Practice*, 14(4), pp. 425–435, 2006.
- [24] Yin L., Srikant R., Eryilmaz A., et Dullerud G. E. Distributed fair resource allocation in cellular networks in the presence of heterogeneous delays. *IEEE Transactions on Automatic Control*, 52:129–134, 2007.
- [25] Abate A., Chen M. et Sastry S. Analysis of an implementable application layer scheme for flow control over wireless networks. In *Mathematical Theory of Networks and Systems*, pages 701–708, Kyoto, Japan, July 2006.
- [26] Liu C.L. et Layland J.W. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1), 46–61, 1973.
- [27] Schmitt E., Contribution au Système d'Information d'un Produit Bois. Appariement automatique de pièces de bois selon des critères de couleur et de texture, Thèse de doctorat de l'Université Henri Poincaré, CRAN, CNRS UMR 7039, Nancy, Septembre 2007.
- [28] IEEE, "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.
- [29] OPNET. OPNET, University program [Online], Available at: http://www.opnet.com/university_program/index.html. 2005.
- [30] Schmitt E., Mazaud C., Bombardier V., Lhoste P., A Fuzzy Reasoning Classification Method for Pattern Recognition, In *15th Int. Conf. on Fuzzy Systems, FUZZIEEE'06*, Vancouver, Canada, pp 5998-6005, 2006.
- [31] Schmitt E., Bombardier V., Charpentier P., Self-Fuzzification Method according to Typicality Correlation for Classification on tiny Data Sets, *Proc. 16th Int. Conf. on Fuzzy Systems, FUZZIEEE'07*, London, England, pp 1072-1077, 2007.
- [32] Zadeh L.A., The concept of linguistic variable and its application to approximate reasoning, *Information sciences*, vol 8, pp 199-249 ; pp 301-357, 1975.
- [33] Cordon O., Gomide F., Herrera F., Hoffmann F., Magdalena L., *Ten years of genetic fuzzy systems: current framework and new trends*, *Fuzzy Sets and Systems*, vol 141, pp 5-31, 2004.
- [34] De Carvalho F.A.T., Fuzzy c-means clustering methods for symbolic interval data, *Pattern Recognition Letters*, vol. 28, pp 423-437, 2007.
- [35] Forest J., Rifqi M., Bouchon-Meunier B., Class Segmentation to Improve Fuzzy Prototype Construction: Visualization and Characterization of Non Homogeneous Classes, *Proc. 15th Int. Conf. on Fuzzy Systems, FUZZIEEE'06*, Vancouver, Canada, pp 555-559, 2006.
- [36] Dubois D., Prade H., What are Fuzzy rules and how to use them?, *Fuzzy Sets and Systems*, vol. 84, pp 169-185, 1996.
- [37] Dubois D., Prade H., Ughetto L., Checking the coherence and redundancy of fuzzy knowledge bases, *IEEE Trans. Fuzzy Systems*, vol. 5, pp 398-417, 1997.
- [38] Ishibuchi H., Nozaki K., Tanaka H., Distributed representation of fuzzy rules and its application to pattern classification, *Fuzzy Sets and Systems*, vol. 52, pp 21-32, 1992.
- [39] Berthold M.R., Mixed fuzzy rule formation, *Int. Journal. of Fuzzy Sets and Systems*, vol 32, pp 67-84, 2003.
- [40] Ishibuchi H., Nozaki K., Tanaka H., A Simple but powerful heuristic method for generating fuzzy rules from numeric data, *Fuzzy sets and systems*, vol 86, pp 251-270, 1997.