

Contrôle de congestion d'un routeur par retour d'état

Yassine ARIBA, Frédéric GOUAISBAUT, Yann LABIT

Université de Toulouse, LAAS-CNRS,
7 avenue du Colonel Roche, 31077 Toulouse, France

{yariba, fgouaisb, ylabit}@laas.fr

Résumé— De nombreux travaux de recherche ont montré les liens existants entre le contrôle de congestion dans les réseaux de communication et la théorie de la commande en Automatique. Nous présentons dans cet article une méthode pour la synthèse d'un AQM (*Active Queue Management*) assurant le contrôle de congestion d'un routeur. Le modèle utilisé est une représentation fluide linéarisée du comportement de TCP (*Transmission Control Protocol*). La synthèse de l'AQM est alors transformée en un problème de commande par retour d'état pour les systèmes à retards. Enfin, un exemple numérique extrait de la littérature et des simulations via le simulateur de réseaux NS-2 [1] complètent notre étude.

Mots-clés— Systèmes à retards, contrôle de congestion, AQM, Lyapunov-Krasovskii, TCP.

I. INTRODUCTION

TCP est le protocole de contrôle de congestion le plus répandu dans les réseaux IP. Il s'agit d'un protocole dit de « bout en bout » et son principe est de contrôler les taux d'envois de chaque source par rapport au trafic courant [2]. Son fonctionnement est simple : à chaque instant, les sources connectées sont en compétition pour l'accès aux ressources et augmentent progressivement la taille des flux émis jusqu'à ce que le réseau entre en congestion (mécanisme du *Drop Tail*). Cependant, cette stratégie de contrôle peut provoquer de larges oscillations des files induisant alors de fortes variations du RTT (*Round Trip Time*) et alternant sous-utilisation et saturation des liens. Par conséquent, la qualité de service (*QoS*) du réseau est considérablement dégradée.

C'est pourquoi des recherches récentes [3] ont exploré la possibilité de faire intervenir les noeuds intermédiaires afin de mieux contrôler le phénomène de congestion. Ainsi, dans le but de compléter TCP, le mécanisme d'AQM situé au niveau des routeurs a été développé. Ce dernier contribue au contrôle du trafic en éjectant prématurément et aléatoirement des paquets de la file d'attente (impliquant alors indirectement la réduction du taux d'envoi d'une source) afin d'anticiper la saturation du buffer. Divers algorithmes sont alors apparus dans la littérature tels que Random Early Detection (RED) [4] et ses variantes [5], Random Early Marking (REM) [6], BLUE [7]. Leurs performances ont été évaluées [8] et des études empiriques ont prouvé leur efficacité [9].

Par ailleurs, il a été montré que le problème de contrôle de congestion et le développement d'AQM pouvaient être vus comme un problème de synthèse de correcteur régulant la file d'attente du routeur. Dans [10], à partir de la théorie de la commande, des AQMs de types *P*, *PI* ont été construits afin de contrôler le phénomène de congestion. A partir du modèle dynamique de TCP/AQM développé par [11], de

nombreux travaux ont été menés pour résoudre le problème de contrôle de congestion dans le cadre de l'Automatique (par exemples [12], [13], [14] et [15] ainsi que ses références). Néanmoins, la plupart de ces études n'ont pas pris en compte le retard. En effet, basée plutôt sur des méthodes fréquentielles, la stabilité du système est assurée pour tous les retards possibles, conduisant alors à des résultats très pessimistes en pratique.

La modélisation du système TCP/AQM par un système à retards n'est pas nouvelle et a déjà été exploitée avec succès. Ainsi la stabilité globale de l'ensemble TCP/AQM a été analysée dans [16] via la théorie de Lyapunov-Krasovskii. Dans le même cadre théorique, l'analyse de la stabilité locale du système bouclé par un retour proportionnel a été réalisé dans [17]. Enfin, dans [18], un retour d'état dépendant du retard est synthétisé à partir d'une méthode de prédiction. Cette dernière est intéressante en théorie mais difficile à mettre en oeuvre en pratique.

Dans cet article, basé sur une fonctionnelle de Lyapunov-Krasovskii récemment développée, un AQM stabilisant localement la file d'attente est construit. Cet objectif est réalisé par la synthèse d'un retour d'état pour les systèmes à retards. Ensuite, cette méthode est appliquée à un système augmenté (ajout d'un effet intégral) afin de rejeter l'erreur induite par des flux non TCP (tels que UDP, *User Datagram Protocol*) assimilés alors à un signal de perturbation. D'autre part, l'implémentation d'un retour d'état sur le modèle proposé permet de prendre en compte deux types d'informations : la taille de la file et le débit agrégé au routeur. Ces dernières données caractérisent généralement deux catégories distinctes d'AQM [19] basés soit sur la file d'attente tels que RED et PI soit sur le débit tels que AVQ et REM (*queue-based AQM* et *rate-based AQM*, respectivement). Ainsi, l'approche adoptée dans cet article permet de tirer profit des deux classes d'AQM et ceci de façon décentralisée (les données mesurées étant locales au routeur). La prise en compte des deux types d'information permet l'application d'une commande efficace pour le contrôle de congestion du routeur.

L'article est organisé en 5 parties. La seconde partie présente le modèle mathématique de la communication entre ordinateurs à travers un routeur. La partie III est dédiée à la synthèse d'un AQM assurant la régulation de la file d'attente au niveau du routeur. Dans la partie IV, un exemple numérique et des simulations à l'aide de NS-2 illustrent la technique développée.

Notations : Soit deux matrices symétriques, A et B , $A > (\geq) B$ signifie que $A - B$ est (semi-) définie positif. A^T est le transposé de A . $\mathbf{1}_n$ et $\mathbf{0}_{m \times n}$ représentent respectivement

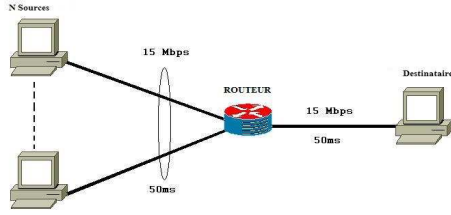


Fig. 1. Configuration du réseau

la matrice identité de dimension n et la matrice nulle de dimension $m \times n$. Si le contexte le permet, les dimensions seront omises. Enfin, pour une matrice $B \in \mathbb{R}^{m \times n}$ telle que $\text{rank}(B) = r$, on définit $B^\perp \in \mathbb{R}^{n \times (n-r)}$ une matrice orthogonale de B par $BB^\perp = 0$.

II. MODÉLISATION

A. Le modèle linéarisé et fluide de TCP

Considérons un réseau de communication constitué de N sources TCP homogènes (c'est-à-dire avec le même temps de propagation) connectées au destinataire à travers un routeur (figure 1). Le goulet d'étranglement est partagé entre les N flux et TCP applique l'algorithme du « congestion avoidance » pour faire face au phénomène de saturation du réseau [2]. Le comportement dynamique de TCP (du type AIMD, *Additive-Increase Multiplicative-Deaccrease*) a été l'objet de nombreuses études, voir à ce sujet la monographie [20] et ses références internes ou encore [21], [15]. Nous utilisons ainsi la modélisation (1) développée par [11], largement utilisée pour la commande de réseaux TCP dans le cadre de la théorie de la commande des systèmes dynamiques [21] :

$$\begin{cases} \dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))}p(t-R(t)) \\ \dot{q}(t) = \frac{W(t)}{R(t)}N - C + d(t) \end{cases} \quad (1)$$

où W est la fenêtre de congestion de TCP, q est la taille de la file d'attente du buffer au niveau du routeur, R est le temps d'aller-retour (RTT) et peut s'exprimer par $R = q/C + T_p$. C , T_p et N sont des paramètres liés à la configuration du réseau et représentent respectivement la capacité du routeur, le temps de propagation et le nombre de source TCP (homogènes). Le signal p correspond à la probabilité de marquage/éjection d'un paquet (il y a marquage si l'option ECN, *Explicit Congestion Notification*, est activée, voir [22]). A la différence de [10], nous avons introduit un signal supplémentaire $d(t)$ modélisant d'autres types de trafic traversant également le routeur et remplissant le buffer. Ces trafics ne sont pas des flux TCP (ils ne respectent donc pas le modèle dynamique de TCP (1)) et peuvent être assimilés à des perturbations comme par exemples les flux UDP qui ne sont pas réactifs aux pertes de paquets.

Suivant l'idée proposée par [10] ou [18], en considérant le RTT $R(t) = h$ constant lorsqu'il est en argument d'une variable et pour un nombre de sources N constant, le modèle (1) peut être linéarisé autour d'un point d'équilibre (voir [10]). Cette première hypothèse est valide si l'écart de la taille de file autour du point d'équilibre reste faible et si le temps de propagation est relativement important par rapport au temps de stockage dans le buffer. Le modèle

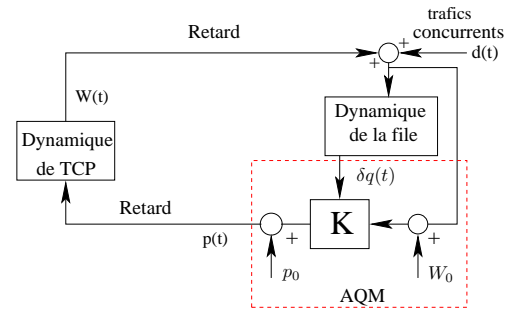


Fig. 2. Synthèse d'un AQM en tant que retour d'état

linearisé est donc de la forme,

$$\begin{cases} \delta \dot{W}(t) = -\frac{N}{R_0^2 C} (\delta W(t) + \delta W(t-h)) \\ -\frac{1}{R_0^2 C} (\delta q(t) - \delta q(t-h)) - \frac{R_0 C^2}{2N^2} \delta p(t-h) \\ \delta \dot{q}(t) = \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) + d(t) \end{cases} \quad (2)$$

où $\delta W \doteq W - W_0$, $\delta q \doteq q - q_0$ et $\delta p \doteq p - p_0$ sont les variations de l'état autour du point d'équilibre. Ce dernier (W_0, q_0, p_0) est défini par

$$\begin{cases} \dot{W} = 0 \Rightarrow W_0^2 p_0 = 2 \\ \dot{q} = 0 \Rightarrow W_0 = \frac{R_0 C}{N}, R_0 = \frac{q_0}{C} + T_p \end{cases}$$

L'entrée du modèle (2) correspond à la probabilité δp d'éjection d'un paquet. Cette probabilité est calculée par l'AQM. Ce dernier a pour objectif de contrôler la taille de la file d'attente du routeur. Dans cet article, ce problème de régulation est traité dans la partie III avec la synthèse d'un retour d'état stabilisant pour les systèmes à retard. En effet, un AQM agit comme un compensateur (voir figure 2) et considérant un retour d'état, la probabilité d'éjection sera exprimée par

$$p(t) = p_0 + k_1 \delta W(t) + k_2 \delta q(t). \quad (3)$$

où k_1 et k_2 sont les composantes de la matrice de gain K qui doit être construite. Il est à noter que l'entrée $p(t) = u(t) + p_0$ est également retardée.

Remarque 1 : Ainsi, le routeur informe implicitement les sources de l'occurrence d'une congestion en provoquant la perte d'un paquet. Cependant, cette méthode a pour conséquence la perte de données puis leur retransmission par les sources. Afin d'éviter cette perte de temps inutile, le mécanisme d'ECN, *Explicit Congestion Notification*, a été proposé par [22]. Comme son nom l'indique, il a pour objectif de signaler explicitement aux sources le phénomène de congestion. En effet, lorsqu'il y a une congestion le routeur met à "1" le bit ECN situé dans l'en-tête IP du paquet. Sur réception de l'acquittement (à son tour marqué par le destinataire) la source réagira comme si ce dernier avait été perdu.

B. Approche par système à retards

Dans cet article, les dynamiques des sources TCP et de la file d'attente sont modélisées par un système à retards. En effet, le retard est un phénomène inhérent aux réseaux et la prise en compte de cette caractéristique a pour but d'améliorer la précision du modèle et la pertinence de l'étude.

On observe que le modèle linearisé de TCP (2) peut

aisément être réécrit sous la forme d'un système à retards où

$$\begin{cases} \dot{x}(t) = Ax(t) + A_d x(t-h) + Bu(t-h) + B_d d(t) \\ x_0(\theta) = \phi(\theta), \text{ with } \theta \in [-h, 0] \end{cases} \quad (4)$$

avec

$$A = \begin{bmatrix} -\frac{N}{R_0^2 C} & -\frac{1}{CR_0^2} \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}, A_d = \begin{bmatrix} -\frac{N}{R_0^2 C} & \frac{1}{R_0^2 C} \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -\frac{C^2 R_0}{2N^2} \\ 0 \end{bmatrix} \quad (5)$$

$B_d = [0 \ 1]^T$, $x(t) = [\delta W(t) \ \delta q(t)]^T$ est le vecteur d'état et $u(t) = \delta p(t)$ l'entrée. $\phi(\theta)$ est la condition initiale.

Considérant l'équation (4), il nous reste à étudier sa stabilité. Dans le cadre des systèmes à retards, il existe principalement trois méthodes pour étudier la stabilité des systèmes à retards : l'analyse directe des racines caractéristiques, l'approche robuste et la théorie de Lyapunov. Cette dernière sera considérée car il s'agit d'une méthode efficace et pratique apportant des conditions LMI et BMI (*Linear/Bilinear Matrix Inequalities*, [23]). Afin d'analyser et de commander le système (4), la théorie de Lyapunov-Krasovskii (voir [24]) qui est une extension de la théorie traditionnelle de Lyapunov, est utilisée.

III. STABILISATION : SYNTHÈSE D'UN AQM

Dans la partie II, le modèle dynamique de l'ensemble TCP/AQM a été présenté. Ce dernier peut s'écrire sous la forme d'un système à retard. Afin d'asservir le phénomène de congestion, il apparaît alors nécessaire de synthétiser un correcteur qui jouera le rôle de l'AQM. Dans un premier temps, un critère d'analyse de stabilité dépendant du retard (*delay dependent*, DD) pour les systèmes à retard est présenté. Puis, basé sur cette condition de stabilité, un critère de synthèse est alors établi.

A. Analyse de stabilité des systèmes à retards

Dans ce paragraphe, il s'agit de déterminer une condition de stabilité prenant en compte une borne maximale admissible sur le retard. Le cas DD suppose que le système est stable pour un retard nul et cherche la valeur maximale du retard qui préserve la stabilité en boucle fermée.

De manière générale, cette méthode utilise une fonctionnelle de Lyapunov et des techniques pour borner les différents termes de la dérivée de la fonctionnelle [24]. Les choix de la fonctionnelle et des techniques de majoration sont à l'origine du conservatisme. Dans cet article, une fonctionnelle de Lyapunov-Krasovskii (6) qui a été récemment développée dans [25] est considérée :

$$V(x_t) = x^T(t)Px(t) + \int_{t-\frac{h_m}{r}}^t \int_{t-\frac{h_m}{r}\theta}^t \dot{x}^T(s)R\dot{x}(s)dsd\theta + \int_{t-\frac{h_m}{r}}^t \begin{pmatrix} x(s) \\ x(s-\frac{1}{r}h_m) \\ \vdots \\ x(s-\frac{r-1}{r}h_m) \end{pmatrix}^T Q \begin{pmatrix} x(s) \\ x(s-\frac{1}{r}h_m) \\ \vdots \\ x(s-\frac{r-1}{r}h_m) \end{pmatrix} ds \quad (6)$$

où $P \in \mathbb{R}^{n \times n}$, $Q \in \mathbb{R}^{rn \times rn}$ et $R \in \mathbb{R}^{n \times n}$ sont des matrices définies positives. $r \geq 1$ est un entier correspondant au pas de discrétisation du retard. A partir de cette fonctionnelle, la proposition suivante est introduite.

Proposition 1: S'il existe des matrices définies positives $P, R \in \mathbb{R}^{n \times n}$, $Q \in \mathbb{R}^{rn \times rn}$, un scalaire $h_m > 0$ et un entier $r \geq 1$ tels que

$$S^{\perp T} \Gamma S^{\perp} < 0 \quad (7)$$

$$\Gamma = \begin{bmatrix} \frac{h_m}{r} \mathbf{R} & \mathbf{P} & 0 & \dots & 0 \\ \mathbf{P} & -\frac{r}{h_m} \mathbf{R} & \frac{r}{h_m} \mathbf{R} & & \vdots \\ 0 & \frac{r}{h_m} \mathbf{R} & -\frac{r}{h_m} \mathbf{R} & & \vdots \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} + \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \mathbf{Q} & \vdots \\ 0 & \dots & 0 \end{bmatrix} + \begin{bmatrix} 0 & \dots \\ 0 & \dots \\ \vdots & \mathbf{Q} \end{bmatrix} \quad (8)$$

et

$$S = \begin{bmatrix} -1 & A & 0_{n \times (r-1)n} & A_d \end{bmatrix} \quad (9)$$

alors, le système (4) (avec $u(t) = 0$ et $d(t) = 0$) est stable pour tout $h \leq h_m$.

Preuve: Il est toujours possible de réécrire (4) sous la forme $S\xi = 0$ où

$$\xi = \begin{bmatrix} \dot{x}(t) \\ x(t) \\ x(t-\frac{1}{r}h) \\ \vdots \\ x(t-\frac{r-1}{r}h) \\ x(t-\frac{r-1}{r}h) \end{bmatrix} \in \mathbb{R}^{(r+2)n} \quad (10)$$

et S est défini par (9). Soit la variable étendue $\xi(t)$ (10), la dérivée de V le long des trajectoires du système (4) s'écrit :

$$\left\{ \begin{array}{l} \dot{V}(x_t) = \xi^T \begin{bmatrix} \frac{h_m}{r} \mathbf{R} & \mathbf{P} & 0 & \dots & 0 \\ \mathbf{P} & -\frac{r}{h_m} \mathbf{R} & \frac{r}{h_m} \mathbf{R} & & \vdots \\ 0 & \frac{r}{h_m} \mathbf{R} & -\frac{r}{h_m} \mathbf{R} & & \vdots \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} \xi \\ + \xi^T \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \mathbf{Q} & \vdots \\ 0 & \dots & 0 \end{bmatrix} \xi - \xi^T \begin{bmatrix} 0 & \dots \\ 0 & \dots \\ \vdots & \mathbf{Q} \end{bmatrix} \xi < 0 \\ \text{t.q.} \quad \begin{bmatrix} -1 & A & 0 & \dots & 0 & A_d \end{bmatrix} \xi = 0 \end{array} \right. \quad (11)$$

$$\Leftrightarrow \left\{ \begin{array}{l} \dot{V}(x_t) = \xi^T \Gamma \xi < 0, \\ \text{t.q.} \quad \begin{bmatrix} -1 & A & 0 & \dots & 0 & A_d \end{bmatrix} \xi = 0 \end{array} \right. \quad (12)$$

où $\Gamma \in \mathbb{R}^{(r+2)n \times (r+2)n}$ dépend de P, R, Q et du retard h_m . En appliquant le lemme de projection [26], l'expression (12) est équivalente à (7).

Remarque 2 :

- Il existe une forme équivalente de ce critère exprimée dans [25] et obtenue à partir du principe de séparation quadratique.
- Dans le même article [25], il est montré que pour $r = 1$, la condition proposée (6) donne les mêmes résultats que la plus part des critères de la littérature. De plus, il est également montré que lorsque r est augmenté, le pessimisme est réduit.

B. Un premier résultat pour la synthèse

Etant donné la condition d'analyse (7) et considérant la loi de commande (3) appliquée au système (4) (dans un premier temps, la perturbation $d(t)$ ne sera pas prise en compte), soit la proposition suivante.

Proposition 2: S'il existe des matrices définies positives $P, R \in \mathbb{R}^{n \times n}$, $Q \in \mathbb{R}^{rn \times rn}$, une matrice $X \in \mathbb{R}^{(r+2)n \times n}$, un

scalaire $h_m > 0$, un entier $r \geq 1$ et une matrice $K \in \mathbb{R}^{m \times n}$ tels que

$$\Gamma + \mathbf{X}S + S^T \mathbf{X}^T < 0 \quad (13)$$

où Γ est défini par (8) et

$$S = \begin{bmatrix} -1 & A & 0_{n \times (r-1)n} & A_d + BK \end{bmatrix} \quad (14)$$

alors, le système (4) peut être stabilisé pour tout $h \leq h_m$ avec la loi de commande $u(t) = Kx(t)$ (et pour $d(t) = 0$).

Preuve : Soit le système (4) avec le retour d'état (3), le système interconnecté suivant est déduit

$$\dot{x}(t) = Ax(t) + \bar{A}_d x(t-h), \quad (15)$$

où $\bar{A}_d = A_d + BK$ et A , A_d et B sont définies par (5). Il est alors possible d'appliquer la condition (7) à (15). En appliquant le lemme de Finsler [26], on peut montrer qu'il existe une matrice $X \in \mathbb{R}^{(r+2)n \times n}$ telle que si (13) est satisfaite alors (7) est vraie. La matrice X , généralement appelée « slack variable » est intéressante pour des objectifs de synthèse et de robustesse.

Remarque 3 :

- Pour résoudre la condition (13), il est nécessaire d'utiliser un solveur BMI.
- Dans [27], un algorithme de relaxation a été développé afin d'obtenir une condition LMI pour trouver un retour d'état stabilisant.

C. Retour d'état avec un effet intégral

Dans le paragraphe précédent, la commande par retour d'état pour les systèmes à retards a été exposé. L'utilisation d'un tel correcteur pour la gestion d'un routeur a été effectué dans [27]. Cependant, il apparait lors de simulation sous NS que dans certains cas, la file d'attente n'est plus régulée au niveau désiré. En effet, une légère erreur en régime permanent est observée et peut être expliquée par une imprécision du modèle. De plus, l'introduction de trafics concurrents de protocoles différents et insensibles aux pertes tels que des flux UDP, affectent également l'équilibre de la file d'attente.

Afin de compenser le décalage induit par l'ajout de trafics non modélisés, l'AQM est doté d'un effet intégral. L'idée est d'appliquer le critère de synthèse précédemment proposé à un système à retards augmenté. Ce dernier est composé du système original (4) et d'un intégrateur pur (voir figure 3). Le système augmenté est de la forme

$$\dot{z} = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} z(t) + \begin{bmatrix} A_d & 0 \\ 0 & 0 \end{bmatrix} z(t-h) + \begin{bmatrix} B \\ 0 \end{bmatrix} \delta p(t-h) \quad (16)$$

avec $z^T = [\delta W \quad \delta q \quad u]^T$ le vecteur d'état étendu. Ainsi, la loi de commande s'écrit

$$\delta p(t) = K \begin{bmatrix} \delta W(t) \\ \delta q(t) \\ u(t) \end{bmatrix} = k_1 \delta W(t) + k_2 \delta q(t) + k_3 \int_0^t \delta q(\theta) d\theta. \quad (17)$$

Les trafics concurrents modélisés en tant que perturbation $d(t)$ occupent une partie de la capacité du routeur et s'ajoutent aux autres flux (TCP) dans le buffer (voir figures 2 et 3). La dynamique de la file d'attente est donc de la forme

$$\dot{q}(t) = \frac{W(t)}{R(t)} N - C + d(t) \quad (18)$$

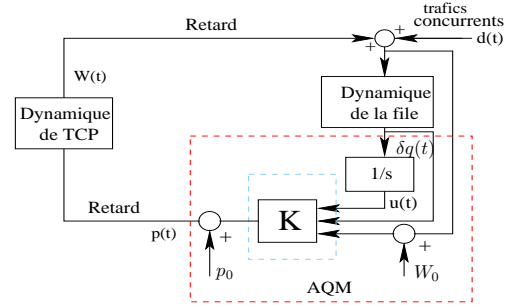


Fig. 3. Synthèse d'un AQM en tant que retour d'état dynamique

Si l'on considère un trafic parallèle UDP utilisé par des applications de type CBR (*Constant Bit Rate*), alors la perturbation $d(t)$ peut être modélisée par un signal de type échelon. A partir de la première équation de (2) et les équations (17) (18), la fonction de transfert $T(s)$ entre la perturbation $d(t)$ et la file d'attente (autour de son point d'équilibre) $\delta q(t)$ s'exprime par :

$$T(s) = \frac{b(s)s}{\left(s + \frac{1}{R_0} \right) s b(s) + \frac{N}{R_0} \left[\frac{s}{R_0^2 C} (1 - e^{-hs}) + a(s) s k_2 + a(s) k_3 \right]}, \quad (19)$$

avec $a(s) = -\frac{R_0 C^2}{2N^2} e^{-hs}$ et $b(s) = s + \frac{N}{R_0^2 C} (1 + e^{-hs}) + a(s) k_1$. On peut aisément observer que pour une perturbation de type échelon, la file d'attente converge toujours vers son équilibre.

D. Estimation de la fenêtre de congestion W

Dans les deux derniers paragraphes, deux méthodes de synthèse d'AQM pour le contrôle de congestion et la régulation de la file d'attente du routeur ont été proposées. Jusque là, nous avons supposé que l'ensemble de l'état était accessible. Cependant, bien que la fenêtre de congestion soit mesurable dans NS (moyennant l'ajout de quelques lignes dans le code de TCP), ce n'est pas le cas en réalité. C'est pourquoi, il est proposé d'estimer cette dernière variable à partir de l'agrégat de flux entrant au niveau du routeur. Le taux d'envoi d'une seule source TCP peut être approximé par

$$x_i(t) = \frac{W(t)}{R(t)}. \quad (20)$$

Cette approximation est valide tant que le modèle ne décrit pas la communication à une échelle de temps plus fine que quelques RTT (voir [21]). Par conséquent, l'ensemble de l'agrégat observé en entrée du routeur est $x(t) = NW(t)/R(t)$. La mesure de l'agrégat de flux n'est pas forcément implémentée dans les routeurs mais est une solution technique exploitable (voir [28], [29] et [30]) comme il a été envisagé respectivement dans [12] et [18] pour la réalisation des AQMs AVQ et PID. Il est à noter qu'il existe principalement deux classes d'AQM. Le RED et le PI, par exemple, sont des AQM basés sur la taille de la file d'attente et peuvent être assimilés à un retour de sortie vis-à-vis de la file. Par opposition, il y a les AQMs dit basés sur les débits de flux tels que AVQ et peuvent être vus comme des retours de sortie vis-à-vis de l'agrégat de flux. L'AQM présenté dans cet article, est finalement un hybride des deux classes puisqu'il s'agit d'un retour d'état prenant en compte les deux types d'information.

IV. SIMULATIONS

Afin d'illustrer et valider l'approche exposée dans les parties précédentes, un exemple numérique extrait de [10] (voir

la figure 1) est présenté. On souhaite réguler la file d'attente à une taille de $q_0 = 175$ paquets. Le temps de propagation et la capacité du routeur sont, respectivement de $T_p = 0.2$ secondes et $C = 3750$ paquets/s (ce qui correspond à une capacité de 15 Mb/s avec une taille de paquet de 500 octets). Par conséquent, pour une charge de $N = 60$ sessions TCP, on a $W_0 = 15$ paquets, $p_0 = 0.008$, $R_0 = 0.246$ secondes. En appliquant les critères de synthèse développés dans les parties III-B et III-C, les deux gains (K_{SF} pour *state feedback* et K_{SFI} pour *state feedback with integral action*)

$$K_{SF} = 10^{-3} \begin{bmatrix} -0.2372 \\ 0.0429 \end{bmatrix} \text{ and } K_{SFI} = 10^{-4} \begin{bmatrix} 0.9385 \\ 0.5717 \\ 0.3559 \end{bmatrix} \quad (21)$$

sont calculés respectivement pour la construction des lois de commande (3) et (17).

Il s'agit ensuite de valider notre approche par des simulations via NS-2 [1], un simulateur de réseaux de communication.

Les valeurs des gains précédemment trouvées sont utilisées pour l'implémentation de l'AQM. La taille de file d'attente désirée est de 175 paquets tandis que la taille du buffer est de 800 paquets. Le protocole de transport utilisé est TCP-New Reno (version de TCP la plus répandue dans l'Internet) et l'option ECN n'est pas disponible. Le réseau possède la même configuration que dans [10]. Les valeurs des coefficients du PI a et b sont respectivement fixées à $1.822e-5$ et $1.816e-5$. Le RED est également paramétré tel que recommandé dans [10]. Les évolutions temporelles de la taille de file d'attente du routeur avec RED et PI en tant qu'AQM sont montrées sur la figure 4. On peut observer que les deux AQM développés dans la partie III-C (avec et sans estimation de W) régulent la file plus rapidement que les autres. Cependant, le second (voir $K_{SFI(aggfl)}$) qui utilise la mesure de l'agrégat plutôt que la fenêtre de congestion à un temps de réponse comparable à celui du PI. Néanmoins, les deux AQMs appliquent une commande efficace dans le sens où la taille de la file est gardée proche de l'équilibre (les oscillations sont relativement faibles). Cette dernière caractéristique est d'autant plus importante qu'elle garantit la validité de l'hypothèse sur l'invariance du RTT.

Sur la figure 5, les mêmes simulations ont été effectuées en introduisant du trafic concurrent. Ce dernier est composé de 7 sources supplémentaires (chacune faisant tourner une application CBR avec le protocole UDP) injectant des paquets de 1000 octets avec un débit de 1Moctets/s entre $t = 40s$ et $t = 100s$. Encore une fois, nos deux AQMs (voir figure 5) sont capables de rétablir l'équilibre malgré la présence de trafic non modélisé supplémentaire. Le PI arrive également à rejeter la perturbation mais plus lentement et avec plus d'oscillations. Par contre, le RED et le retour d'état simple K_{SF} (3) stabilisent toujours la file d'attente mais à un équilibre différent de l'équilibre souhaité. La présence du trafic UDP a introduit un décalage sur le remplissage du buffer.

Les tableaux I et II résument les résultats en terme de régulation apportés par chaque AQM. Certains indices statistiques ont été calculés lors des simulations (mesures effectuées sur l'ensemble de la simulation : tableau I et pendant la perturbation : tableau II). Ces caractéristiques sont la moyenne, l'écart type (ET) et le carré du coefficient de variation ($CV2 = (ET/moy)^2$). Ce dernier paramètre évalue la dispersion relative de la taille de la file d'attente autour de sa valeur d'équilibre. La moyenne rend compte de

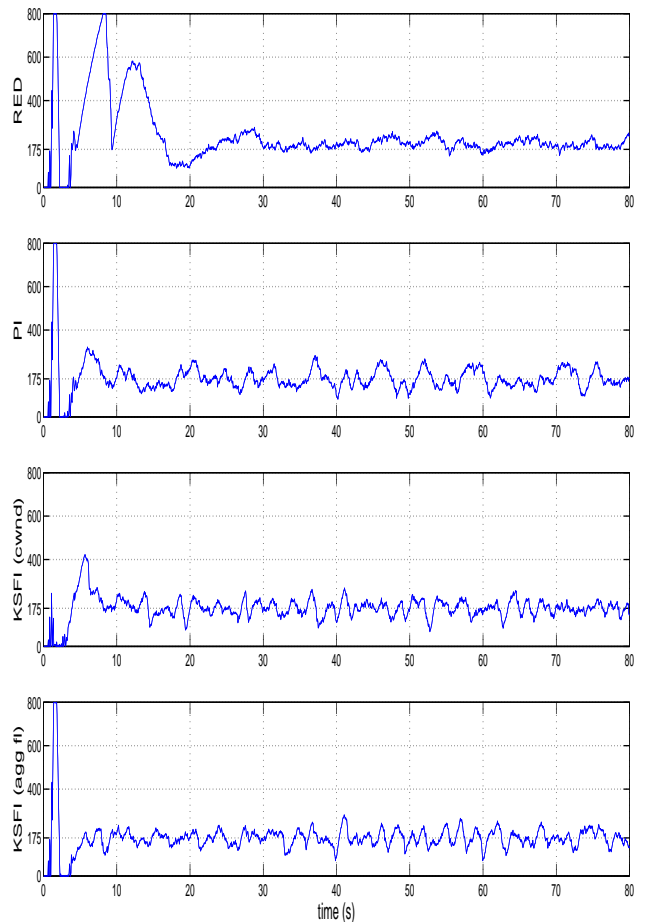


Fig. 4. Evolutions temporelles de la file d'attente pour différents AQMs (RED , PI , $K_{SFI(cwnd)}$, $K_{SFI(Aggfl)}$).

la précision de la commande tandis que l'écart type montre la capacité de l'AQM à garder la taille de file proche de sa valeur d'équilibre. Dans les tableaux I et II, on peut observer que $K_{SFI(cwnd)}$ maintient un contrôle efficace sur la file du buffer durant l'ensemble de la simulation. Même si $K_{SFI(aggfl)}$ est légèrement plus lent que le premier, ses statistiques (ET et $CV2$) montrent également une bonne régulation. Bien que PI rejette la perturbation assez rapidement, il apparaît de larges fluctuations autour de sa valeur moyenne.

AQMs	RED	PI	K_{SF}	$K_{SFI}(cwnd)$	$K_{SFI}(aggfl)$
Moy	235.7	176.7	263.9	175.9	175.5
ET	112.40	71.19	78.59	54.57	63.64
CV2	0.227	0.162	0.088	0.096	0.131

TABLE I
CARACTÉRISTIQUES STATISTIQUES SUR L'ENSEMBLE DE LA SIMULATION
POUR LES DIFFÉRENTS AQMS (UNITÉS EN PAQUETS)

V. CONCLUSION

Dans cet article, l'élaboration d'un AQM pour le contrôle de congestion dans les réseaux de communication a été proposé. Le développement de cet AQM a été effectué en se basant sur un résultat de synthèse d'un retour d'état pour les systèmes à retards. Ensuite, une action intégrale a été ajoutée par augmentation de l'état du système afin de compenser le remplissage du buffer dû à la présence de

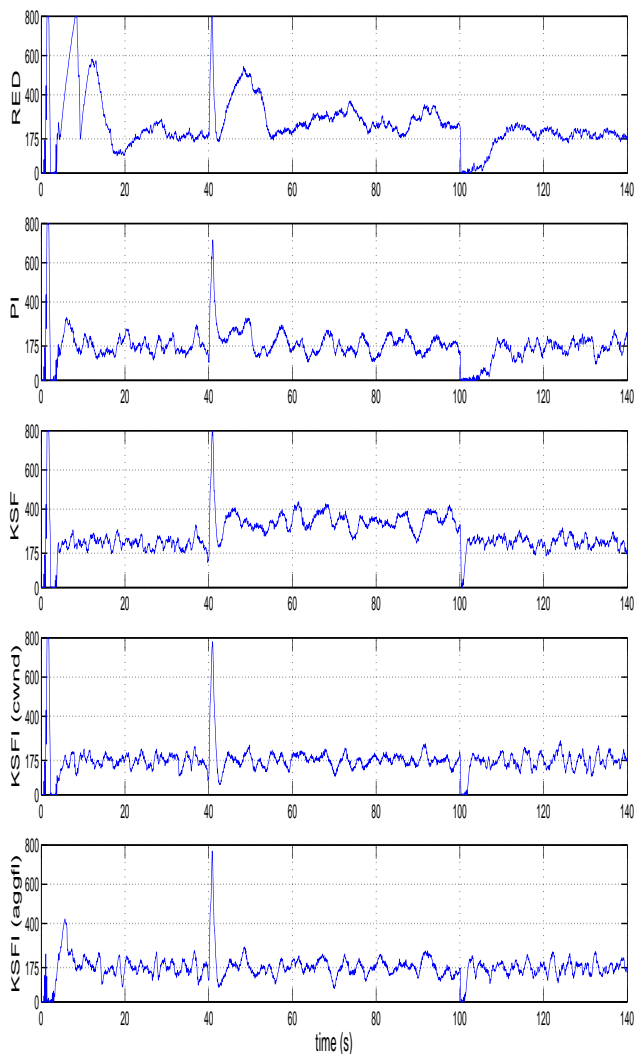


Fig. 5. Evolutions temporelles de la file d'attente pour différents AQMs (RED , PI , $K_{SFI}(cwnd)$, $K_{SFI}(Aggfl)$) avec du trafic UDP en parallèle.

AQMs	RED	PI	K_{SF}	$\bar{K}_{SFI}(cwnd)$	$\bar{K}_{SFI}(aggfl)$
Moy	270.3	178.3	338.0	173.4	175.6
ET	57.39	40.42	41.21	35.08	28.32
CV2	0.045	0.051	0.014	0.040	0.026

TABLE II

CARACTÉRISTIQUES STATISTIQUES LORS DE LA PERTURBATION (ENTRE 40 ET 100 SECONDES) POUR LES DIFFÉRENTS AQMs (UNITÉS EN PAQUETS)

trafics concurrents. Enfin, des simulations sous NS-2 ont montré l'efficacité des AQMs développés. Des travaux futurs consisteront d'un point de vue théorique à améliorer les lois de commandes et réduire le pessimisme et, d'un point de vue expérimental, valider ces lois sur une plateforme d'émulation de réseaux.

RÉFÉRENCES

[1] K. Fall et K. Varadhan. The ns manual. notes and documentation on the software ns2-simulator, 2002. URL : www.isi.edu/nsnam/ns/.

[2] V. Jacobson. Congestion avoidance and control. *ACM SIGCOMM*, pages 314–329, Stanford, CA, August 1988.

[3] B. Braden, D. Clark, et J. Crowcroft. Recommendations on queue management and congestion avoidance in the internet. RFC 2309, April 1998.

[4] S. Floyd et V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1 :397–413, August 1993.

[5] W.-C. Feng, D. D. Kandlur, D. Saha, et K. G. Shin. A self-configuring red gateway. *IEEE INFOCOM'99*, volume 3, pages 1320–1328, March 1999.

[6] S. Athuraliya, D. Lapsley, et S. Low. An enhanced random early marking algorithm for internet flow control. *IEEE INFOCOM*, pages 1425–1434, December 2000.

[7] W. et al. Feng. Blue : A new class of active queue management algorithms. Technical Report CSE-TR-387-99, University of Michigan, 1999.

[8] S. Ryu, C. Rump, et C. Qiao. Advances in active queue management (aqm) based tcp congestion control. *Telecommunication Systems*, 4 :317–351, 2004.

[9] L. Le, J. Aikat, K. Jeffay, et F. Donelson Smith. The effects of active queue management on web performance. *ACM SIGCOMM*, pages 265–276, August 2003.

[10] C. V. Hollot, V. Misra, D. Towsley, et W. Gong. Analysis and design of controllers for aqm routers supporting tcp flows. *IEEE Trans. on Automat. Control*, 47 :945–959, June 2002.

[11] V. Misra, W. Gong, et D. Towsley. Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. *ACM SIGCOMM*, pages 151–160, August 2000.

[12] S. Kunniyur et R. Srikant. Analysis and design of an adaptive virtual queue (avq) algorithm for active queue management. *ACM SIGCOMM*, pages 123–134, San Diego, CA, USA, aug 2001.

[13] Y. Fan, F Ren, et C. Lin. Design a pid controller for active queue management. *IEEE International Symposium on Computers and Communication (ISCC)*, volume 2, pages 985–990, 2003.

[14] D. Agrawal et F. Granelli. Redesigning an active queue management system. *IEEE Globecom*, volume 2, pages 702–706, December 2004.

[15] S. Tarbouriech, C. T. Abdallah, et J. Chiasson. *Advances in communication Control Networks*. Springer, 2005.

[16] A. Papachristodoulou. Global stability of a tcp/aqm protocol for arbitrary networks with delay. *IEEE CDC 2004*, pages 1029–1034, December 2004.

[17] W. Michiels, D. Melchior-Aguilar, et S.I. Niculescu. Stability analysis of some classes of tcp/aqm networks. *International Journal of Control*, volume 79 (9), pages 1136–1144, September 2006.

[18] K. B. Kim. Design of feedback controls supporting tcp based on the state space approach. *IEEE Trans. on Automat. Control*, volume 51 (7), July 2006.

[19] D. Supratim et R. Srikant. Rate-based versus queue-based models of congestion control. *IEEE Trans. on Automat. Control*, 51 :606–619, April 2006.

[20] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[21] H. S. Low, F. Paganini, et J.C. Doyle. *Internet Congestion Control*, volume 22, pages 28–43. IEEE Control Systems Magazine, Feb 2002.

[22] K. K. Ramakrishnan et S. Floyd. A proposal to add explicit congestion notification (ecn) to ip. RFC 2481, January 1999.

[23] S. Boyd, L. El Ghaoui, E. Feron, et V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, USA, 1994. in Studies in Applied Mathematics, vol.15.

[24] K. Gu, V. L. Kharitonov, et J. Chen. *Stability of Time-Delay Systems*. Birkhäuser Boston, 2003. Control engineering.

[25] F. Gouaisbaut et D. Peaucelle. Delay-dependent stability analysis of linear time delay systems. *IFAC Workshop on Time Delay System (TDS'06)*, Aquila, Italy, July 2006.

[26] R. Skelton, T. Iwazaki, et K. Grigoriadis. *A unified algebraic approach to linear control design*. Taylor and Francis series in systems and control, 1998.

[27] Y. Labit, Y Ariba, et F. Gouaisbaut. Design of lyapunov based controllers as tcp aqm. *2nd IEEE Workshop on Feedback control implementation and design in computing systems and networks (FeBID'07)*, pages 45–50, Munich, Germany, May 2007.

[28] Cisco Systems. Introduction to cisco ios netflow - a technical overview. Cisco NetFlow site reference. URL :<http://www.cisco.com/>.

[29] P. Barford et D. Plonka. Characteristics of network traffic flow anomalies. November 2001.

[30] Seong Soo Kim et A. L. Narasimha Reddy. Netviewer : a network traffic visualization and analysis tool. *LISA '05 : Proceedings of the 19th conference on Large Installation System Administration Conference*, pages 185–196. USENIX Association, 2005.