

Workshop MOCOSY

Automatique et réseaux: contrôle de congestion d'un routeur

Yassine Ariba

encadrants: F. Gouaisbaut et Y. Labit (MAC et OLC)



Université
de Toulouse

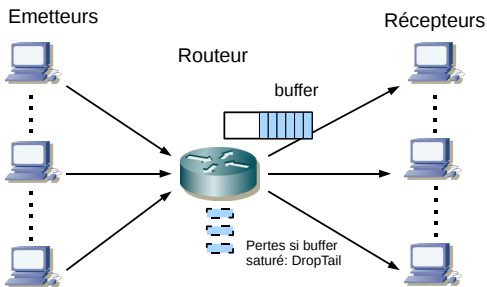


Université Paul Sabatier, LAAS-CNRS
Toulouse III

Introduction

Problème de congestion d'un routeur:

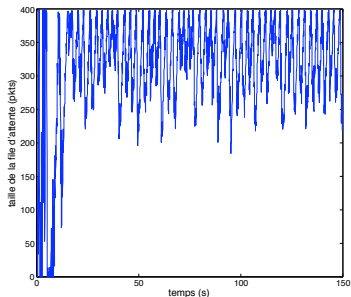
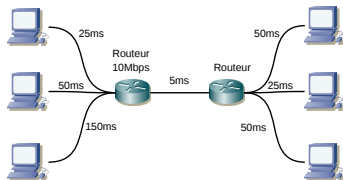
- TCP est un mécanisme de contrôle de congestion de bout en bout.
- Sur réception d'un paquet → le receveur renvoie un acquitement.
- Sur réception d'un acq (ou pas) → la source augmente (ou diminue) son taux d'envoi.
- Si le trafic est trop intense → le buffer du routeur se remplit.



- Les paquets sont éjectés quand le buffer est plein → mécanisme du *Drop tail*.

Introduction

Simulations du mécanisme de *Drop tail* sur NS-2



Lorsque le buffer est saturé (taille: 400pkts) → pertes:

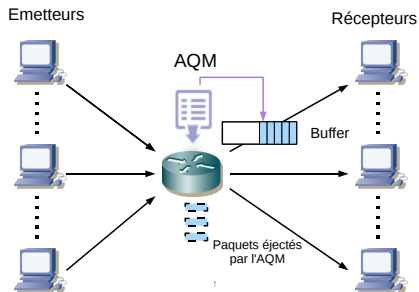
- grandes oscillations,
- proche de la saturation du buffer,
- délai d'attente important.

✪ Objectif: Réguler la taille de file d'attente du routeur.

Introduction

Active queue management (AQM):

- AQM aide TCP pour le contrôle de congestion.
- Idée: anticiper la saturation de buffer.
- AQM est un mécanisme qui force la perte de paquets.
- Sur éjection d'un paquet, AQM agit implicitement sur les taux d'envoi des sources.



✪ But: Améliorer la *Qualité de Service* (QoS) { délai → applications interactives, temps-réel,
gigue → flux multimédias.

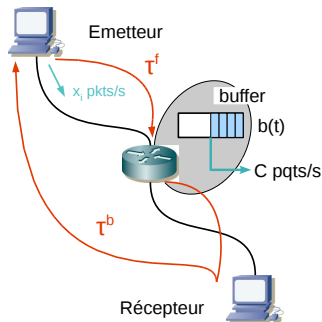
Modélisation

Modèle fluide de l'ensemble TCP/AQM

$$\begin{cases} \dot{x}_i(t) &= \frac{x_i(t-\tau_i)}{x_i(t)\tau_i(t)^2} (1 - p_i(t - \tau_i^b)) - \frac{x_i(t-\tau_i)x_i(t)}{2} p_i(t - \tau_i^b) + \frac{x_i(t)}{\tau_i(t)} - \frac{x_i(t)}{\tau_i(t)C} \sum_N x_k(t - \tau_k^f) \\ \dot{b}(t) &= -C + \sum_N x_i(t - \tau_i^f) \\ \tau_i &= \frac{b(t)}{C} + T_{p_i} = \tau_i^f + \tau_i^b \end{cases}$$

avec:

- $x_i \doteq$ taux d'envoi (paquets/s)
- $b \doteq$ taille de la file d'attente (paquets)
- $\tau_i \doteq$ round trip time (RTT) (secondes)
- $\tau_i^f \doteq$ forward delay (secondes)
- $\tau_i^b \doteq$ backward delay (secondes)
- $C \doteq$ capacité du lien (paquet/s)
- $N \doteq$ nombre de sessions TCP
- $p_i \doteq$ probabilité d'éjection d'un paquet



Modélisation

Linéarisation

Equilibre:

$$\begin{cases} \tau_i^* &= T_{pi} + b^*/C \\ \dot{b}(t) &= 0 \rightarrow \sum_N x_i^* = C \\ \dot{x}_i(t) &= 0 \rightarrow p_i^* = \frac{2}{2+(x_i^* \tau_i^*)^2} \end{cases}$$

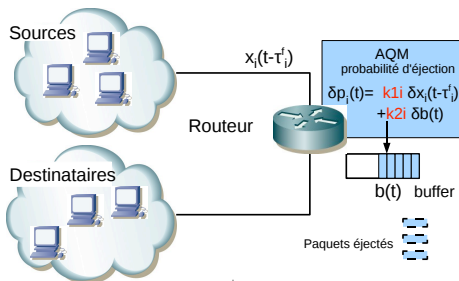
Modèle linéaire:

$$\begin{bmatrix} \dot{x}_1(t) \\ \vdots \\ \dot{x}_N(t) \\ \dot{b}(t) \end{bmatrix} = A \begin{bmatrix} \delta x_1(t) \\ \vdots \\ \delta x_N(t) \\ \delta b(t) \end{bmatrix} + A_d \begin{bmatrix} \delta x_1(t - \tau_1^f) \\ \vdots \\ \delta x_N(t - \tau_N^f) \\ \delta b(t) \end{bmatrix} + B \begin{bmatrix} \delta p_1(t - \tau_1^b) \\ \vdots \\ \delta p_N(t - \tau_N^b) \end{bmatrix}$$

- Approximation linéaire du modèle de TCP autour d'un point d'équilibre,
- δx_i et δb sont valeurs des débits et file d'attente autour du point d'équilibre,
- il faut "jouer" sur les taux de pertes δp_i pour contrôler le trafic.

Contrôle de congestion du routeur

Calcul du taux de perte $\delta p_i(t)$ appliqué par l'AQM:



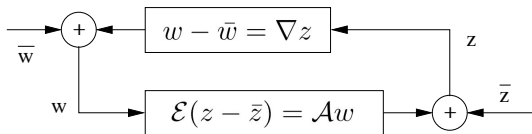
Système interconnecté obtenu:

$$\begin{bmatrix} \delta \dot{x}_1(t) \\ \vdots \\ \delta \dot{x}_N(t) \\ \delta \dot{b}(t) \end{bmatrix} = A \begin{bmatrix} \delta x_1(t) \\ \vdots \\ \delta x_N(t) \\ \delta b(t) \end{bmatrix} + A_d \begin{bmatrix} \delta x_1(t - \tau_1^f) \\ \vdots \\ \delta x_N(t - \tau_N^f) \\ \delta b(t) \end{bmatrix} + B K_1 \begin{bmatrix} \delta x_1(t - \tau_1) \\ \vdots \\ \delta x_N(t - \tau_N) \end{bmatrix} + B K_2 \begin{bmatrix} \delta b(t - \tau_1^b) \\ \vdots \\ \delta b(t - \tau_N^b) \end{bmatrix}$$

✪ Objectif: Synthétiser K_1 et K_2 tels que l'interconnexion soit stable

Analyse de stabilité

Approche par séparation quadratique (analyse robuste):



Théorème

Le système incertain représenté sur la figure ci dessus est stable si et seulement s'il existe une matrice $\Theta = \Theta^*$ satisfaisant les conditions

$$\begin{bmatrix} \mathcal{E} & -\mathcal{A} \end{bmatrix}^{\perp*} \Theta \begin{bmatrix} \mathcal{E} & -\mathcal{A} \end{bmatrix}^{\perp} > 0$$

$$\begin{bmatrix} 1 & \nabla^* \end{bmatrix} \Theta \begin{bmatrix} 1 \\ \nabla \end{bmatrix} \leq 0.$$

⇒ Transformer notre modèle sous la forme d'une interconnexion.

Analyse de stabilité

Construction de la matrice incertaine ∇ :

$$\overbrace{\begin{bmatrix} X(t) \\ \hline x_1(t - \tau_1^f) \\ \vdots \\ x_N(t - \tau_N^f) \\ \hline b(t - \tau_1^b) \\ \vdots \\ b(t - \tau_N^b) \\ \hline x_1(t - \tau_1) \\ \vdots \\ x_N(t - \tau_N) \end{bmatrix}}^w = \overbrace{\begin{bmatrix} s^{-1} \mathbf{1}_{N+1} & & & \\ & \mathcal{D}_{\tau^f} & & \\ & & \mathcal{D}_{\tau^b} & \\ & & & \mathcal{D}_{\tau} \end{bmatrix}}^{\nabla} \overbrace{\begin{bmatrix} \dot{X}(t) \\ \hline x_1(t) \\ \vdots \\ x_N(t) \\ \hline b(t) \\ \vdots \\ b(t) \\ \hline x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix}}^z$$

$$\text{avec } \mathcal{D}_{\diamond} = \begin{bmatrix} e^{-\diamond_1 s} & & 0 \\ & \ddots & \\ 0 & & e^{-\diamond_N s} \end{bmatrix} \text{ et } X(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \\ b(t) \end{bmatrix}.$$

Modélisation

Construction des matrices \mathcal{E} et \mathcal{A} :

$$\begin{array}{c} \overbrace{\left[\begin{array}{c} \dot{X}(t) \\ \hline x_1(t) \\ \vdots \\ x_N(t) \\ \hline b(t) \\ \vdots \\ b(t) \\ \hline x_1(t) \\ \vdots \\ x_N(t) \end{array} \right]}^z \end{array} = \begin{array}{c} \overbrace{\left[\begin{array}{cccc} A & \bar{A}_d & BK_2 & BK_1 \\ E_1 & 0 & 0 & 0 \\ E_2 & 0 & 0 & 0 \\ E_1 & 0 & 0 & 0 \end{array} \right]}^{\mathcal{A}} \end{array} \begin{array}{c} \overbrace{\left[\begin{array}{c} X(t) \\ \hline x_1(t - \tau_1^f) \\ \vdots \\ x_N(t - \tau_N^f) \\ \hline b(t - \tau_1^b) \\ \vdots \\ b(t - \tau_N^b) \\ \hline x_1(t - \tau_1) \\ \vdots \\ x_N(t - \tau_N) \end{array} \right]}^w \end{array}$$

avec $E_1 = \left[\begin{array}{cc} 1_N & 0_{N \times 1} \end{array} \right]$ et $E_2 = \left[\begin{array}{c|c} & 1 \\ & \vdots \\ 0_N & 1 \end{array} \right]$.

Analyse de stabilité

Définition du séparateur Θ satisfaisant la 2nd inégalité du théorème:

$$\forall s^{-1} \in \mathbb{C}^+ : -s^{-1}P - s^{-1*}P \leq 0, P > 0,$$

$$\begin{bmatrix} 1 & s^{-1*} \end{bmatrix} \begin{bmatrix} 0 & -P \\ -P & 0 \end{bmatrix} \begin{bmatrix} 1 \\ s^{-1} \end{bmatrix} \leq 0.$$

$$\forall s^{-1} \in \mathbb{C}^+ : |e^{-\tau s}| \leq 1,$$

$$\begin{bmatrix} 1 & e^{-\tau s*} \end{bmatrix} \begin{bmatrix} -Q & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} 1 \\ e^{-\tau s} \end{bmatrix} \leq 0$$

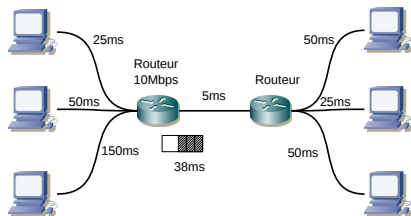
La 1^{ière} inégalité constitue la condition de stabilité.

$$\begin{bmatrix} \mathcal{E} & -\mathcal{A} \end{bmatrix}^{\perp*} \Theta \begin{bmatrix} \mathcal{E} & -\mathcal{A} \end{bmatrix}^{\perp} > 0$$

On en déduit les valeurs de K_1 et K_2 de notre correcteur.

Contrôle de congestion du routeur: exemple

Considérons la topologie:



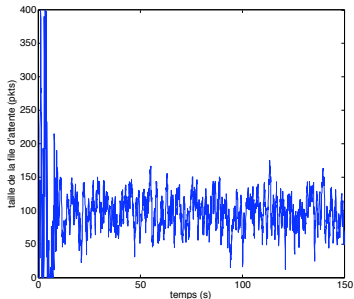
Modèle:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{b}(t) \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 & h_1 \\ 0 & a_2 & 0 & h_2 \\ 0 & 0 & a_3 & h_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta x_1(t) \\ \delta x_2(t) \\ \delta x_3(t) \\ \dot{b}(t) \end{bmatrix} + \begin{bmatrix} f_1 & f_1 & f_1 & 0 \\ f_2 & f_2 & f_2 & 0 \\ f_3 & f_3 & f_3 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \delta x_1(t - \tau_1^f) \\ \delta x_2(t - \tau_2^f) \\ \delta x_3(t - \tau_3^f) \\ \delta b(t) \end{bmatrix} + \begin{bmatrix} g_1 \delta p_1(t - \tau_1^b) \\ g_2 \delta p_2(t - \tau_2^b) \\ g_3 \delta p_3(t - \tau_3^b) \\ 0 \end{bmatrix}$$

✪ Objectif: Réguler la file d'attente $b(t)$ à une valeur désirée: 100pqts.

Simulations: NS-2

Simulations de notre AQM sur NS-2



Régulation de la file d'attente à 100 paquets.

- Le temps de file d'attente est fixé à une valeur désirée (ici $\sim 38ms$),
- peu d'oscillations entraine une gigue faible,
- en jouant sur le point d'équilibre, on peut effectuer un *service différencié*.

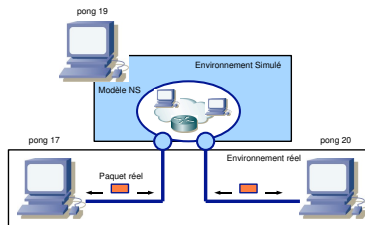
Emulation: 2 approches possibles

Configuration du noyau:

- émuler les caractéristiques du réseau en utilisant les outils linux
route, tc, netem...
- créer la discipline de la file d'attente du routeur,
- ajouter le code de notre AQM dans l'utilitaire tc,
- recompiler le noyau.

★ Implémentation réel de l'AQM.

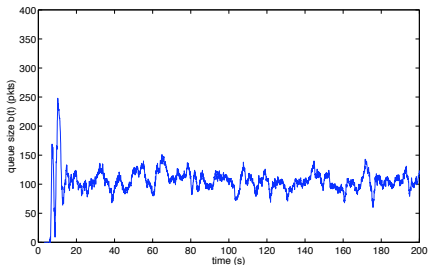
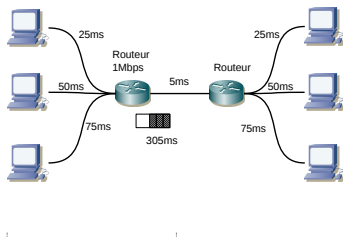
Emulation avec NS temps-réel:



- Pong19 crée un environnement simulé,
- Capture et génération des paquets d'un réseau physique: récupérés dans NS par les "Network object",
- Liaison avec l'environnement NS via des "Tap agent".

Emulation

Régulation de la file d'attente à 80 paquets.



Test avec NS en temps réel de notre correcteur K

- Régulation de réels flux TCP.
- erreur en régime permanent: pb de modèle, pb d'implémentation, pb de précision des paramètres réseaux ?
- travaux en cours ...

Conclusion

Théorie

- condition de stabilité et synthèse de l'AQM,
⇒ à améliorer (réduction du conservatisme, pertinence de la loi de commande)
- simulations Matlab/Simulink ⇒ OK,
- trafic monitoring: estimation de flux, détection d'anomalies,
⇒ théorie des observateurs.

Expérimentation

- simulations NS ⇒ OK
- emulation caractéristiques du réseau (retards, capacité...)
- implémentation de l'AQM
 - NS émulation avec un ordonnanceur temps réel, ⇒ AQM émulé,
 - ajout de code et recompilation du noyau ⇒ AQM réel.