RAIRO. RECHERCHE OPÉRATIONNELLE

EMMANUEL BESSON

Modèles analytiques de routeurs

RAIRO. Recherche opérationnelle, tome 34, n° 2 (2000), p. 213-236

<http://www.numdam.org/item?id=RO_2000__34_2_213_0>

RAIRO Operations Research
RAIRO Oper. Res. **34** (2000) 213-236

MODÈLES ANALYTIQUES DE ROUTEURS (*)

par Emmanuel BESSON ([1])

Communiqué par Bernard LEMAIRE

Résumé. – *On étudie ici les performances d'un routeur dans un contexte de réseaux à haut débit, à travers un modèle analytique de réseaux de files d'attente. Un tel modèle permet d'obtenir des résultats concernant les critères de performance classiques. En particulier, l'étude analytique de la probabilité de perte d'un paquet dans l'équipement met en évidence un réseau de files d'attente sous forme produit. Les résultats obtenus sont alors confrontés à ceux donnés par la simulation. Ils fournissent de précieuses indications pour les opérateurs chargés de la configuration des mémoires internes de ces équipements.*

Mots clés : Réseaux haut débit, chaînes de Markov, réseaux de files d'attente, forme produit.

Abstract. – *We focus on performance study of routers in high-speed network through a queuing network analytical model. Such a model gives accurate results about classical performance criteria. For example, analytical study of packet loss probabilities in a router uses a product-form queuing network. The analytical results are compared to simulation results, and they provide routers managers with invaluable information for internal memories tuning.*

Keywords: High-speed networks, Markov chains, queuing networks, product form.

1. INTRODUCTION

L'évolution des réseaux de transmission de données vers le haut-débit a conduit les constructeurs à développer de nouvelles architectures pour les routeurs [1, 13]. Afin d'améliorer les performances de ces équipements et d'optimiser ainsi celles de l'ensemble des réseaux, des techniques de hiérarchisation des tables de routage au travers de systèmes de caches à plusieurs niveaux sont notamment utilisées dans la gestion des mémoires.

Dans un routeur, un paquet de données va être soumis à deux types d'opérations : un acheminement physique au travers de l'équipement, et un traitement de l'adresse portée par son en-tête. Cette dernière opération fait

(*) Reçu en décembre 1997.

([1]) France Télécom - CNET, 905 rue A. Einstein, 06921 Sophia-Antipolis Cedex, France.

Recherche opérationnelle/Operations Research, 0399-0559/00/02/$ 7.00

donc appel au système de tables de routage hiérarchisé, et constitue aussi le goulet d'étranglement. Le modèle analytique de performances présenté ici s'attache donc à analyser les délais et probabilités de perte d'un paquet induits par ce mécanisme caractéristique, ainsi que les limitations de débit de l'équipement. Les résultats, obtenus sous des hypothèses de trafic simples et usuelles à l'aide de la théorie des files d'attente et des réseaux sous forme produit, sont validés par simulation.

2. LE MODÈLE

2.1. Présentation

Les routeurs étudiés ici utilisent un système de tables de routage hiérarchisé. Ils comportent une table de routage complète centralisée, une table cache également centralisée qui contient les routes les plus demandées, et des tables cache distribuées dans chaque interface qui contiennent les listes de chemins les plus utilisés par les paquets incidents à cette interface. Le parcours successif de ces tables et le succès ou échec qui en découle définissent trois modes de traitement de l'adresse d'un paquet. En référence à la terminologie Cisco [1, 13], le premier mode *Autonomous Switching* (*AS*) correspond à une résolution de routage à l'aide des tables cache locales seules. Le second mode, conventionnellement dénommé *Fast Switching* (*FS*), utilise la table cache centrale, sans accéder à la table complète. Enfin, le troisième et dernier mode, ou *Process Swiching* (*PS*), nécessite l'accès à cette dernière. Il est évident que le temps moyen de traitement d'un paquet va intuitivement croître depuis le mode *AS* jusqu'au mode *PS*.

La figure 1 présente le modèle de réseau de files d'attente proposé dans [3] pour la modélisation du fonctionnement d'un routeur.

Ce modèle met en place le couple constitué d'une entrée logique (input) $i\,(1 \leq i \leq N)$ et d'une sortie logique (output) $j\,(1 \leq j \leq N)$, où N désigne le nombre d'interfaces logiques de l'équipement. Les travaux de Chen et Stern sur les modèles de commutateurs [8] montrent qu'un tel couplage est représentatif du comportement du routeur.

Les arrivées de paquets à chaque entrée sont supposées indépendantes et distribuées selon un processus de Poisson. On distinguera trois classes de paquets modélisant les différents traitements possibles dans un routeur :

- la classe A correspond aux paquets dont l'adresse va être traitée en *Autonomous Switching* ;

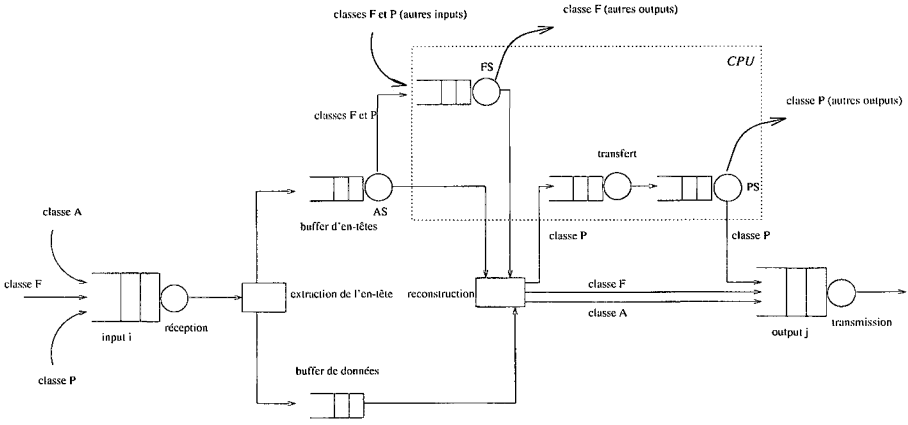

Figure 1. – Modèle de routeur.

- la classe *F* correspond aux paquets dont l'adresse va être traitée en *Fast Switching* ;
- la classe *P* correspond aux paquets dont l'adresse va être traitée en *Process Switching*.

Les paquets arrivent donc dans chaque entrée i selon un processus supposé Poissonnien. Ils sont stockés dans une file FIFO composée de buffers dédiés (un paquet occupe un buffer). La capacité de cette file sera donc traduite en nombre de paquets. De plus, le premier serveur correspond à une phase de réception dont la durée dépend d'une taille supposée exponentiellement distribuée. Il sera donc lui-même de type exponentiel.

La première phase de traitement qui succède à la réception consiste en une extraction de l'en-tête (opération qui se déroule au fil de l'eau et ne fait donc pas intervenir de temps de service supplémentaire) qui est envoyé dans une file spécifique. Les données sont alors placées en attente dans une autre file locale à l'interface i jusqu'à ce que l'en-tête leur correspondant ait subi son traitement et soit prêt à être recomposé. Pour ces en-têtes, le premier serveur représente le passage dans le mode *AS*, c'est-à-dire une consultation de la table cache locale. Cette consultation nécessite un temps *a priori* non constant, mais dont l'étude de la distribution n'est pas l'objet de ce travail. Dans le cadre de l'analyse, il sera dans un premier temps supposé que cette distribution est exponentielle et il sera dans la suite discuté de la validité d'une telle hypothèse.

Si le paquet est de classe A, c'est-à-dire si sa nouvelle adresse a été construite à partir des tables cache locales, la reconstruction s'opère en fin de traitement et le paquet complet est envoyé vers sa destination. Dans le modèle homogène, pour une sortie j donnée, il n'y aura qu'un paquet sur N (taille du routeur) qui ira vers cette sortie.

Si le paquet appartient à une autre classe, il subit le mode *FS*. Dans la pratique, son en-tête est alors copié dans un buffer « système » qui reçoit aussi les en-têtes similaires venus des autres interfaces. L'étude analytique modélise ici ce traitement centralisé en faisant intervenir une file FIFO, dont la capacité s'exprime en paquets (car les en-têtes ont une taille fixe), alimentée par un flux local et un flux exogène indépendant (car les processus d'arrivées au niveau des interfaces sont indépendants). Pour la simulation, ce flux exogène sera directement issu de modèles d'interfaces similaires non agrégés. Dans un premier temps, les tables cache centrales vont être examinées (mode *FS*). Le serveur correspondant à cette consultation sera supposé de type exponentiel pour les raisons précédemment mentionnées. Dans la pratique, cette opération est effectuée par le *Switch Processor* indépendant de la CPU en ce qui concerne l'opération de traitement qui nous intéresse ici.

Le paquet subit alors sa reconstruction, mais pour une sortie j donnée, seul un paquet de classe F sur N suivra ce chemin. Si le paquet est de classe F, il part vers la sortie. S'il appartient à la classe P, il est envoyé dans la file FIFO de la CPU par une opération de transfert sur le bus suivi d'une réception similaire à celle en entrée des interfaces (l'une se fait néanmoins sur des liens physiques, l'autre sur un bus interne). Cette réception fait donc apparaître un premier serveur de type exponentiel. La file gère un buffer par paquets et sa capacité sera donc exprimée en nombre de paquets. Ensuite, les paquets attendent leur traitement : les tables de routage complètes sont alors consultées (mode *PS*). Ce service est assuré par la CPU elle-même (en fait par le *Route Processor* indépendant du *Switch Processor* pour le mode *FS*) et obéit aux mêmes hypothèses que pour les autres serveurs de parcours de table. La succession de la réception et du traitement oblige à introduire deux files successives séparées. Dans la pratique, il n'existe qu'une seule et même file. La contrainte de capacité s'appliquera donc à l'ensemble de ces deux files, et non à l'une puis à l'autre. Le paquet est ensuite envoyé vers la sortie avec le même mécanisme que ci-dessus.

Enfin, la file de transmission représente l'acheminement physique du paquet traité sur le lien de sortie et obéit également à une discipline FIFO.

La topologie particulière du routeur, et l'application du théorème de Burke [7] assure que la superposition des flux supposés indépendants est telle qu'elle peut se ramener à un flux Poissonien de taux identique à celui considéré en entrée.

Dans un réseau de transmission global, on attend d'un routeur qu'il dirige les paquets avec le minimum de délai et dans les meilleures conditions de sécurité (c'est-à-dire avec une probabilité de perte la plus faible possible). Cependant, on ne peut pas concevoir d'équipements capables de soutenir n'importe quel débit en entrée. Les limitations technologiques des processeurs, et les contraintes au niveau de la place mémoire sont telles qu'un routeur ne peut soutenir, sans saturer, qu'un débit maximal à determiner. En conséquence, les trois critères retenus seront :

- le délai de transfert d'un paquet à travers la structure de routage (depuis la file de réception jusqu'à la file de transmission) ;
- le débit maximal autorisé par l'implémentation sans qu'il y ait saturation ;
- la probabilité de perte d'un paquet à la suite d'un dépassement de capacité mémoire.

2.2. Notations

Considérons en premier lieu un système homogène et équilibré : l'intensité du trafic est la même pour chacune des N entrées, et les destinations sont uniformément distribuées entre les N sorties. Un paquet arrivant à une entrée i a donc une probabilité égale à $1/N$ d'aller vers une sortie j donnée. Ces hypothèses seront remises en cause ultérieurement. Les arrivées suivent un processus Poissonnien et les serveurs sont de type exponentiel.

Le taux d'arrivée global des paquets en entrée d'une interface, noté λ, est décomposé selon les trois taux d'arrivée λ_A, λ_F et λ_P respectivement pour les paquets de classes A, F, et P. Ces grandeurs s'expriment toutes en *paquets par seconde* (ou p/s).

La taille des paquets suit une loi exponentielle de moyenne $1/\mu$ (en *bits par paquets* ou b/p). De la même façon, les modes *AS*, *FS* et *PS* sont respectivement caractérisés par leur taux de service μ_A, μ_F et μ_P exprimés en p/s.

Il est nécessaire de définir la probabilité p_A de succès en mode *AS* ($p_A = \lambda_A/\lambda$) qui définit la probabilité qu'un paquet puisse être routé à partir des seules informations contenues dans la table cache locale, et la

probabilité p_F de succès en mode *FS* ($p_F = \lambda_F/\lambda$) qui définit la probabilité qu'un paquet n'ayant pu être routé localement (en mode *AS*) le soit grâce aux informations contenues dans la table cache centrale.

C_b désigne la capacité du bus interne (en b/s) qui transfère les paquets vers la CPU, et C_l la capacité des liens (en b/s) sur lesquels le routeur travaille (entrée/sortie).

Enfin, on reprend les notations courantes pour : le taux d'utilisation de la file de réception/transmission $\rho = \lambda/(C_l.\mu)$, le taux d'utilisation de la file *Autonomous* $\rho_A = \lambda/\mu_A$, le taux d'utilisation de la file *Fast* $\rho_F = N.(\lambda_F + \lambda_P)/\mu_F$, le taux d'utilisation de la file de transfert $\rho_B = N.\lambda_P/(C_b.\mu)$ et le taux d'utilisation de la file *Process* $\rho_P = N.\lambda_P/\mu_P$.

Dans la suite de ce travail, on supposera que le système est stable, à savoir que les taux d'utilisation cités sont tous inférieurs à 1. On dira que le routeur « sature » ou approche de la saturation, dès qu'au moins un de ces taux d'utilisation sera égal ou supérieur à 1. Ces notations seront complétées au cours de l'analyse.

3. EXPRESSIONS DES CRITÈRES DE PERFORMANCES

3.1. Délai de transit

Pour étudier le délai de transit d'un paquet, le « plus mauvais » cas est retenu : on suppose les capacités des files infinies. Ainsi, les paquets peuvent être conservés durant de longues périodes (la limite théorique est infinie) dans les différentes files traversées. Les valeurs obtenues par cette analyse donnent une borne supérieure du délai de traversée [8].

On ne donnera que les valeurs moyennes des délais. Ainsi, pour un temps de réponse estimé W_*, on exprimera sa valeur moyenne en la notant : $\bar{W}_*$.

3.1.1. Étude des files de réception et de transmission

Délai en transmission. La file de sortie depuis le port j se comporte comme une file M/M/1 en discipline FIFO. Il faut cependant noter que les paramètres de cette file sont *a priori* corrélés avec ceux des systèmes qui la précèdent car, comme le service requis par un paquet dépend de sa longueur, il demeure fixé pour toutes les files qu'il va visiter. Il s'agit d'un obstacle courant dans la modélisation de flux de paquets par des réseaux de files d'attente. On adoptera donc à ce niveau l'hypothèse d'indépendance proposée par Kleinrock dans [11], et retenue par Chen et Stern dans leur

étude des commutateurs de paquets [8], sur la base des travaux de Boxma [6] et d'analyses de simulation.

Soit W_O le temps d'attente dans la file de sortie. On a :

$$\bar{W_O} = \frac{\rho/(C_l.\mu)}{1-\rho}. \tag{3.1}$$

Délai en réception. La file d'entrée agit comme un tampon et elle a le comportement typique d'une file M/M/1. Soit W_I le temps de réponse associé à cette file. On a :

$$\bar{W_I} = \frac{1/(C_l.\mu)}{1-\rho}. \tag{3.2}$$

3.1.2. Délai de transit pour les paquets de classe A

Les paquets de classe A subissent le mode de traitement le plus simple et le plus rapide. La seule file traversée par leurs en-têtes est la file *Autonomous*, qui est une file M/M/1.

Soit W_A le temps de réponse pour la file *Autonomous*. On a :

$$\bar{W_A} = \frac{1/\mu_A}{1-\rho_A}. \tag{3.3}$$

Dès que le traitement de l'en-tête est terminé, on suppose que les données correspondantes sont disponibles (hypothèse de temps d'extraction/reconstruction nuls). Finalement le délai de transit moyen pour les paquets de classe A, noté $\bar{D_A}$, sera :

$$\bar{D_A} = \bar{W_I} + \bar{W_A} + \bar{W_O},$$

les valeurs étant données par les équations (3.1, 3.2), et (3.3).

3.1.3. Délai de transit pour les paquets de classe F

Pour les paquets de classe F, à la suite d'une tentative infructueuse en mode *AS* (l'adresse n'est pas dans la cache locale), leurs en-têtes sont envoyés dans une file centrale du CPU gérée par le *Switch Processor* pour une consultation de la cache centrale. Cette file est de type M/M/1 également.

Soit W_F le temps de réponse pour la file *FS*. On a :

$$\bar{W_F} = \frac{1/\mu_F}{1-\rho_F}, \tag{3.4}$$

ce qui donne le délai de transit moyen $\bar{D_F}$ des paquets de classe F selon :

$$\bar{D_F} = \bar{W_I} + \bar{W_A} + \bar{W_F} + \bar{W_O}.$$

3.1.4. Délai de transit pour les paquets de classe P

Les paquets de classe P rejoignent la mémoire centrale de la CPU dans leur intégralité (données et en-tête). Il y a donc une phase de transfert sur le bus, représentée par un premier système, puis une phase de traitement (consultation de la table de routage globale) qui constitue un second système. L'ensemble de ces deux files en tandem peut être ici étudié successivement puisqu'on suppose les capacités infinies. Ce ne sera plus le cas pour l'étude de la probabilité de perte (*cf.* Sect. 3.3).

Ainsi, si W_B et W_P désignent respectivement les temps de réponse pour la file de transfert et la file *PS*, on a :

$$\bar{W_B} = \frac{1/(C_b.\mu)}{1-\rho_B}, \tag{3.5}$$

et :

$$\bar{W_P} = \frac{1/\mu_P}{1-\rho_P}, \tag{3.6}$$

ce qui donne le délai de transit moyen $\bar{D_P}$ des paquets de classe P selon :

$$\bar{D_P} = \bar{W_I} + \bar{W_A} + \bar{W_F} + \bar{W_B} + \bar{W_P} + \bar{W_O}.$$

3.1.5. Résultats numériques

Pour visualiser les délais de traversée dans un routeur, une étude numérique est proposée en fonction de la charge représentée par le paramètre λ. Les valeurs retenues pour les autres paramètres sont les suivantes, et respectent l'ordre de grandeur des valeurs réelles fournies par Cisco (hors les taux de succès fixés arbitrairement) : $N = 8$, $p_A = 0,3$, $p_F = 0,5$, $1/\mu = 4000$ b/p, $C_l = 155$ Mb/s, $\mu_A = 250000$ p/s, $\mu_F = 100000$ p/s, $C_b = 500$ Mb/s, $\mu_P = 25000$ p/s.

La figure 2 montre que dans ce cas, le mode *PS* limite la charge à 8928 p/s. Le délai de traversée pour un paquet de classe A ou F demeure dès lors relativement insensible à la charge. Finalement, en moyenne, un paquet entrant subit un délai situé entre 50 µs et 350 µs lors d'une utilisation de l'équipement loin du point de saturation. Toutefois, un paquet traité en mode

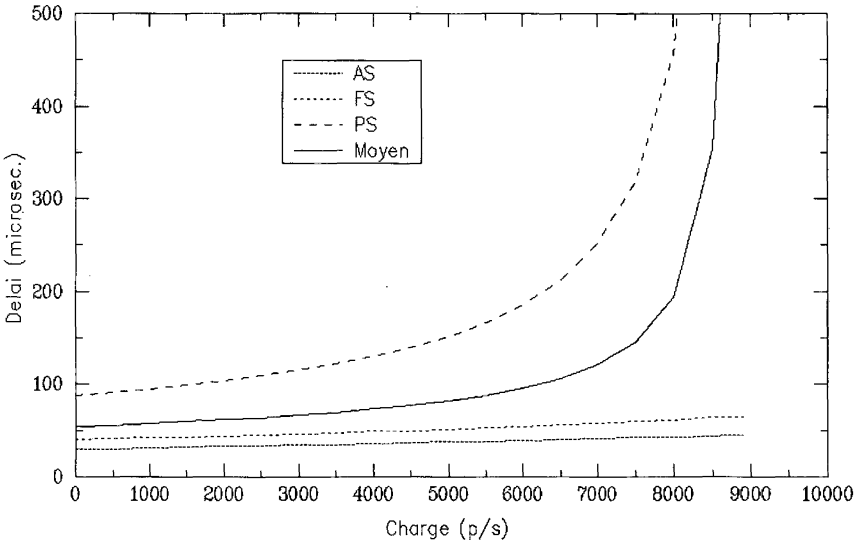

Figure 2. – Délai de transit.

PS pour une charge offerte de 8500 p/s sur chaque interface peut mettre près d'1s à traverser le routeur.

3.2. Débit de l'équipement

Le débit maximal $thp_{\max}$ sera le minimum des débits maximaux des files *AS*, *FS*, *PS*, du transfert sur le bus interne et de la réception/transmission sur les liens. Ces débits sont calculés en fixant le taux d'utilisation de chacune des files correspondantes à sa valeur maximale, soit 1. On obtient ainsi :

$$thp_{\max} = \min\left(\mu_A, \frac{\mu_F}{N(1-p_A)}, \frac{C_b.\mu}{N(1-p_A)(1-p_F)}, \frac{\mu_P}{N(1-p_A)(1-p_F)}, C_l.\mu\right). \tag{3.7}$$

La valeur du débit maximal dépend donc essentiellement du profil de trafic, à travers les probabilités de succès dans les différents modes de traitement. Ainsi, l'exemple numérique du paragraphe précédent montre un débit maximal limité par la file *PS* à $\frac{\mu_P}{N.(1-p_A).(1-p_F)} = 8928$ p/s. Ceci peut paraître faible en comparaison des performances affichées par les constructeurs. Toutefois, il convient de considérer la charge totale du routeur, sur l'ensemble de ses interfaces, qui traite alors plus de 70000 p/s.

Cette étude montre néanmoins de façon claire que les performances données par les équipementiers induisent en erreur puisqu'elles tiennent généralement compte d'un trafic commuté en mode le plus rapide (par exemple *AS* pour Cisco). Dans la pratique ces performances sont bien entendu amoindries. Quand on sait que le mode *PS* est automatiquement invoqué dès que des fonctions de firewall sont appliquées, ou que le mode *FS* est également obligatoire lors d'une fonction de ré-encapsulation de protocoles, un tel modèle prend tout son intérêt pour le concepteur ou l'administrateur de réseaux.

Sur la figure 3, le graphique montre la valeur du débit maximal par interface de l'équipement en fonction des taux de succès dans les modes *AS* et *FS*, et ce avec les données du paragraphe 3.1.5 pour les autres paramètres.

Ce débit plafonne à 38750 p/s qui correspond au débit maximal accepté par les liens (et donc par le réseau) ce qui est compréhensible. On peut notamment remarquer que cette valeur optimale n'est atteinte que pour des valeurs de taux de hit dans les tables cache élevées (supérieures à 80 %) ce qui rejoint notre commentaire précédent sur la pertinence des valeurs affichées par les constructeurs. Ainsi, on peut descendre jusqu'à 3125 p/s lorsque seul le mode *PS* est activé, ce qui donne un modeste débit global de 25000 p/s, soit dix fois moins que le débit donné pour le mode *AS*.

3.3. Taux de perte

On affecte maintenant des capacités aux différentes files du modèle. Celles-ci sont respectivement notées K_I, K, K_F, K_P et K_O pour la file de réception, la file des données en interface, la file des en-têtes en mode *FS*, la file du CPU (transfert et mode *PS*) et la file de transmission.

La file des en-têtes en mode *AS* ne sature que si la file de l'interface sature. En effet, à chaque en-tête correspond exactement un paquet en attente de traitement ; c'est pourquoi fixer une capacité à la file des en-têtes en mode *AS* est inutile.

La première difficulté réside dans le fait que l'introduction de capacités sur le réseau de files d'attente du routeur entraîne une possible réduction du trafic incident des files autres que la file de réception. Ainsi, si des paquets sont bloqués, et donc le trafic incident réduit (notamment le trafic exogène), les temps de réponse dans les files seront modifiés. Dès lors, la probabilité de perte l'est aussi, et ainsi de suite... Pour contourner ce problème, il suffit de calculer le taux de perte correspondant de manière itérative (et convergente) comme proposé dans [8].

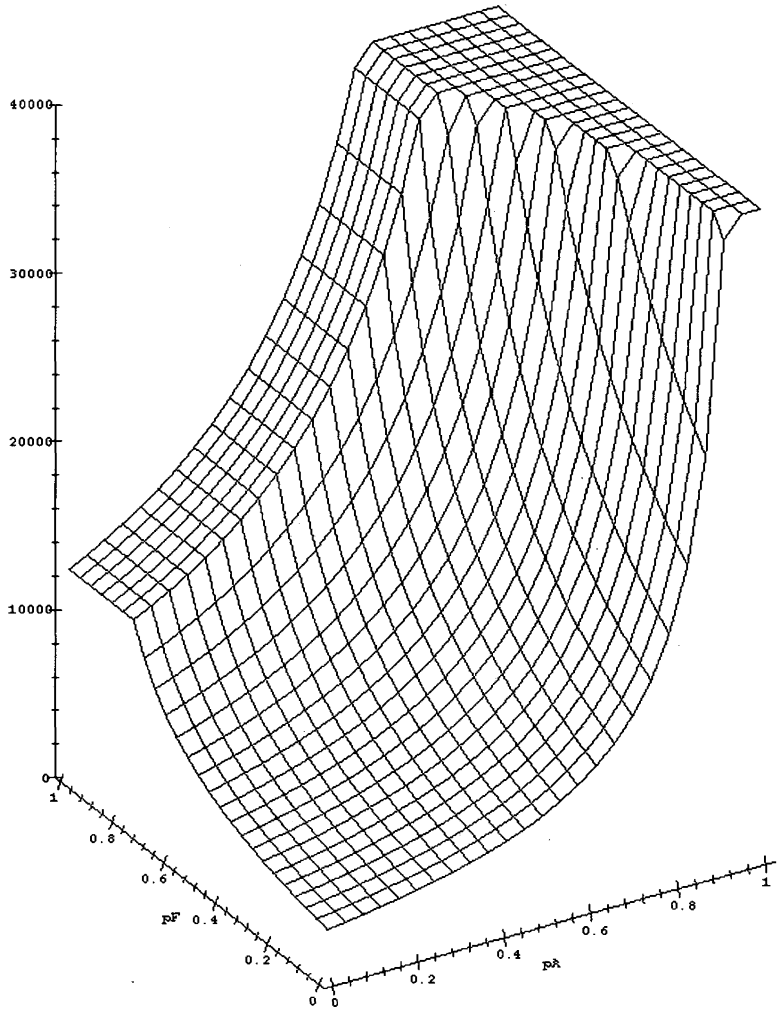

Figure 3. – Débit maximal.

3.3.1. Perte en réception

On utilise ici les résultats connus sur les files M/M/1/K cités en [10]. La probabilité P_p^I de perdre un paquet en entrée vaut alors :

$$P_p^I = \frac{(1-\rho).\rho^{K_I}}{1-\rho^{K_I+1}}. \tag{3.8}$$

3.3.2. Perte en interface

Avec le modèle présenté, la perte ne peut intervenir qu'au niveau de la file contenant les données qui attendent le traitement de leur en-tête. Le problème est que, dans cette file, des paquets de toutes les classes sont présents, et que la discipline n'est pas précise. En effet, on suppose seulement que, dès que l'en-tête correspondant sort de son service, les données sont évacuées, libérant la place. À chaque instant, le nombre de paquets dans cette file sera égal au nombre d'en-têtes en cours ou en attente de traitement. Donc l'étude du nombre d'en-têtes donne implicitement celle du nombre de paquets de données en interface.

Le problème réside ici dans le fait que les en-têtes peuvent expérimenter deux types de perte : pour le traitement en mode *AS* et pour le traitement en mode *FS* dans la file correspondante (de discipline FIFO).

Dans une interface, il est possible qu'un paquet arrivant après un autre en ressorte avant lui car il a subi un traitement moins important et plus rapide. Ceci amène une difficulté supplémentaire de calcul de la probabilité de perte associée. Sur le sous-réseau de la figure 4, il existe deux contraintes de capacité, l'une limitant le nombre d'en-têtes locaux dans *l'ensemble* des deux files, l'autre limitant le nombre d'en-têtes (locaux ou exogènes) dans la deuxième file.

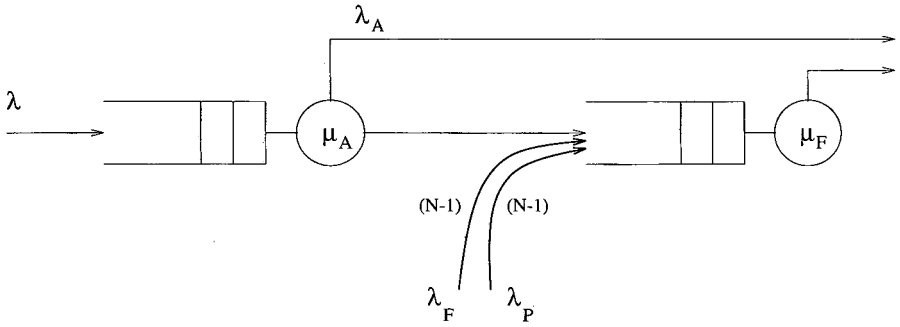

Figure 4. – Sous-réseau de traitement en interface.

Sous ces contraintes, ce sous-réseau ne répond malheureusement pas aux hypothèses des réseaux de Jackson formulées dans [10]. L'évaluation de cette probabilité de perte, égale à P_p^A, sera présentée en section 4.

3.3.3. Perte dans la file du CPU

En pratique, le CPU ne fait intervenir qu'une file qui reçoit les paquets en transfert sur le bus, puis les traite en mode *PS*. On ne tient donc compte que d'une capacité globale sur l'ensemble des deux files (transfert et traitement en mode *PS*) notée K_P. On dispose donc d'un sous-système de deux files M/M/1 en tandem avec un nombre de clients limité. Les résultats courants [10, 12] donnent l'expression de P_p^P, probabilité de perte dans la file du CPU :

$$P_p^P = \frac{\sum_{n_B=0}^{K_P} (\rho_B^*)^{n_B}.(\rho_P^*)^{K_P-n_B}}{\sum_{n_B=0}^{K_P} \sum_{n_P=0}^{K_P-n_B} (\rho_B^*)^{n_B}.(\rho_P^*)^{n_P}},$$

avec :

$$\rho_B^* = \rho_B.(1 - P_p^A) \text{ et } \rho_P^* = \rho_P.(1 - P_p^A).$$

3.3.4. Perte en transmission

Par un raisonnement similaire, en tenant compte des pertes en amont, on obtient l'expression de P_p^O, probabilité de perte dans la file de transmission :

$$P_p^O = \frac{(1 - \rho_O^*).{\rho_O^*}^{K_O}}{1 - {\rho_O^*}^{K_O+1}},$$

$$\text{avec} \quad \rho_O^* = \frac{\lambda.(1 - P_p^A) - \lambda_P.(1 - P_p^A).P_p^P}{C_l.\mu}. \tag{3.10}$$

3.3.5. Perte globale

Finalement, le taux de perte P_p dans un routeur dépend de la classe de paquets considérée. Pour les paquets de classes A et F, on applique :

$$P_p = 1 - (1 - P_p^I).(1 - P_p^A).(1 - P_p^O), \tag{3.11}$$

et pour la classe P :

$$P_p = 1 - (1 - P_p^I).(1 - P_p^A).(1 - P_p^P).(1 - P_p^O). \tag{3.12}$$

On peut dès lors calculer une valeur moyenne du taux de perte, notée $\bar{P}_p$, pour l'équipement qui autorise des comparaisons avec des routeurs utilisant d'autres technologies et qui affichent des taux de perte fournis par le constructeur. Ce critère s'exprime alors selon :

$$\bar{P}_p = 1 - (1 - P_p^I).(1 - P_p^A).(1 - P_p^O) \cdot \Big(1 + (1 - p_A).(1 - p_F).P_p^P\Big). \tag{3.13}$$

3.4. Déséquilibre du trafic

Les résultats ont été obtenus jusqu'ici sous l'hypothèse d'un trafic équilibré. Cependant, les performances d'un routeur sont *a priori* dépendantes du déséquilibre éventuel du trafic. Pour tenir compte de ce paramètre, il est aisé d'adapter notre modèle, comme il a été fait en [8].

On peut considérer un taux d'arrivée λ_i^I pour $1 \leq i \leq N$, Poissonnien, dépendant de l'entrée. On définit alors λ comme la moyenne des λ_i^I, soit :

$$\sum_{i=1}^{N} \lambda_i^I = N.\lambda.$$

En sortie, on introduit, pour un paquet quelconque, la probabilité P_j que sa destination soit le port de sortie j avec $1 \leq j \leq N$ ($\sum_{j=1}^{N} P_j = 1$).

Le trafic entrant dans la file de sortie j est caractérisé par son intensité λ_j^O donnée par :

$$\lambda_j^O = P_j.\sum_{i=1}^{N} \lambda_i^I = N.P_j.\lambda.$$

Toutes les formules précédemment présentées peuvent s'appliquer en substituant au taux λ les valeurs ci-dessus. Pour déterminer une expression moyenne sur le système, il suffit de faire des sommes pondérées, par exemple :

$$\bar{W_O} = \sum_{j=1}^{N} P_j.\bar{W_{O,j}}, \quad \text{avec} \quad \bar{W_{O,j}} = \frac{\lambda_j^O/(C_l.\mu)}{C_l.\mu - \lambda_j^O}.$$

4. ÉTUDE APPROFONDIE DU MODÈLE D'INTERFACE

4.1. Modèle local

4.1.1. Exposé du modèle

Le modèle de la figure 1 permet d'obtenir aisément le délai de traversée et le débit maximal de l'équipement en supposant les capacités de files infinies. Par contre, pour la probabilité de perte d'un paquet, dès lors que l'on affecte des capacités finies aux différentes files, le mécanisme mis en oeuvre localement dans l'interface soulève de plus grandes difficultés de résolution analytique, car le modèle particulier d'interface de la figure 5 concernant les traitements *AS* et *FS* met en oeuvre trois files d'attente aux comportements fortement corrélés.

La *file 0* correspond au buffer qui accueille les données après extraction de leur en-tête. Elle possède une capacité K. La *file 1* correpond au buffer recevant les en-têtes extraits ; on ne lui affecte pas de contrainte de capacité puisqu'on suppose que cette dernière est au moins égale à K et donc qu'une

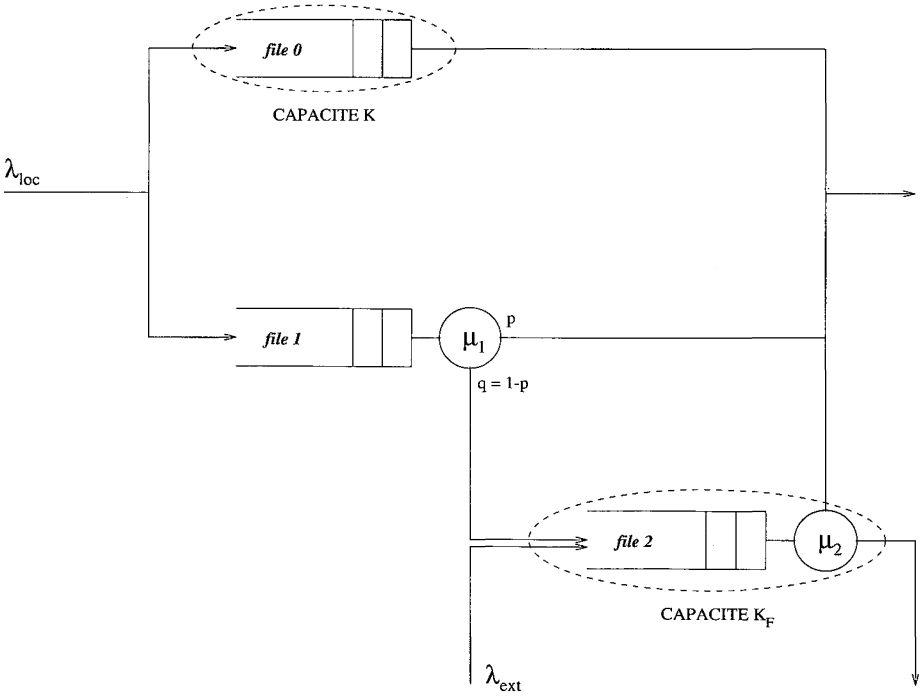

Figure 5. – Modèle d'interface.

saturation de la *file 0* interviendra toujours simultanément ou avant une saturation de cette *file 1*. Enfin, la *file 2* représente le buffer du mode *FS*. Elle est affectée d'une capacité K_F, et reçoit les en-têtes provenant du flux local λ_{loc} et du flux exogène λ_{ext} correspondant aux $(N-1)$ autres interfaces.

Le problème est qu'une perte d'un paquet local à l'interface considérée peut intervenir, soit dans la file locale (*file 0*), soit dans la file partagée (*file 2*) qui sont deux files de type FIFO. Mais, dans le deuxième cas, le modèle analytique doit pouvoir déterminer, lors d'une perte, si cette dernière concerne un paquet local ou un paquet du flux exogène. Il est alors *a priori* nécessaire de décrire non seulement le nombre de paquets respectifs dans la *file 2*, mais aussi l'ordre de ces paquets à l'intérieur de la file, ce qui conduit à des espaces d'états inexploitables.

Toutefois, les réseaux étudiés par Lam [12] répondent à ce type de contraintes et une extension du théorème BCMP [2] montre dans ce cas l'existence d'une solution sous forme produit que l'on présente maintenant.

4.1.2. Notations

Le modèle de réseau de files d'attente considéré sera celui de la figure 6, comprenant deux files 1 et 2 et deux flux (ou classes) de clients notés 1 pour les clients dits *locaux* et 2 pour les clients dits *extérieurs*.

λ_j est le taux d'arrivée des clients de classe j, μ_i le taux de service de la file i, et ρ_i^j le taux d'utilisation de la file i par les clients de classe j.

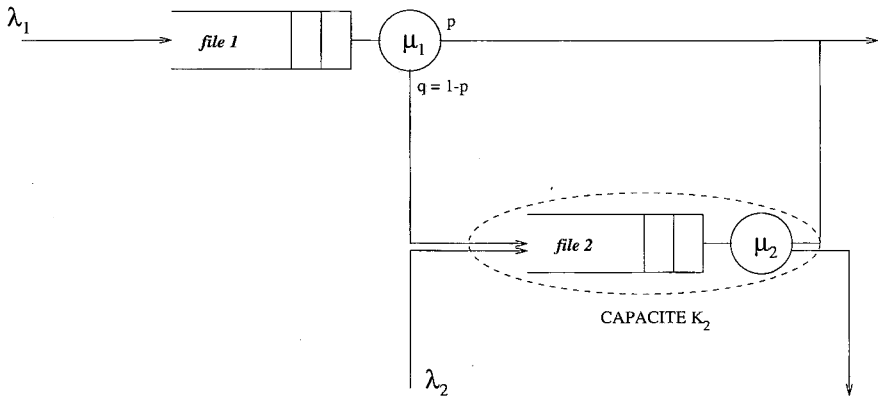

Figure 6. – Modèle de réseau étudié.

Le nombre de clients de classe j dans la file i est noté n_i^j ; n_i désigne le nombre total de clients dans la file i ($n_i = \sum_j n_i^j$), et n^j le nombre total de clients de classe j dans le réseau ($n^j = \sum_i n_i$).

Le routage vers la sortie en fin de service de file 1 se fait avec une probabilité p. Inversement, le routage vers la file 2 en fin de service de file 1 se fera avec la probabilité $q = 1 - p$.

Enfin, K_2 représente la capacité de la file 2, et K le nombre total de clients de classe 1 admis dans l'ensemble du réseau.

On suppose dans toute la suite que ces paramètres sont tels que la chaîne de Markov associée, notée Γ, est ergodique. Un état S de cette chaîne de Markov sera désigné par : $S = (n_1; n_2^1; n_2^2)$. On notera $P(S)$ la probabilité stationnaire associée à l'état S de la chaîne Γ. Le nombre d'états de cette chaîne est alors donné par :

$$\mathcal{N}(K, K_2) = \frac{(K+2).(K+1).K_2}{2} - \frac{(K+1).K.(K-1)}{6}. \tag{4.1}$$

Ainsi, on aura par exemple $\mathcal{N}(50, 200) = 244375$, ce qui donne un ensemble de transitions possibles entre les états généralement trop important à énumérer.

4.2. Résolution sous forme produit

4.2.1. Principe de résolution et résultats

Dans tous les cas, une perte survient dès qu'un client local arrive en entrée du système et trouve la file 1 pleine. Ceci correspond aux états vérifiant $n_1 + n_2^1 = K$. Par ailleurs, on peut avoir un problème pour un client local au sein même du système si ce client doit aller dans la file 2 et que celle-ci est saturée. Deux possibilités se présentent alors, modifiant chacune la chaîne de Markov associée : soit le client est abandonné (cas 1), soit le client est bloqué dans la file 1 (cas 2).

En fait, Cisco indique, pour sa ligne de produits [9], qu'un paquet est ignoré s'il arrive en entrée d'une interface alors que son buffer est plein, et qu'il est ejecté s'il y a saturation du buffer du CPU. En conséquence, le cas 1 a été retenu.

Dans le réseau considéré, la perte d'un client *local* peut donc intervenir, soit dans la file 1, soit dans la file 2. Dans le premier de ces cas, pour un client de classe 1 arrivant en entrée de la file 1, la probabilité de perte $P_{p,1}^A$ est :

$$P_{p,1}^A = \sum_{S/(n^1=K)} P(S).$$

Dans le second cas, il faut que le client soit, à son arrivée, accepté dans la file 1, et qu'ensuite il soit servi et dirigé vers la file 2 alors que celle-ci est pleine. Il s'agit de calculer la probabilité $P_{p,2}^A$ pour un paquet, *à son entrée dans l'interface* (et non à son entrée dans la file 2), de subir, après son passage dans la file 1, un rejet de la file 2. Son expression nécessite de calculer les probabilités de transition vers l'état (n_1, K_2) depuis tous les autres états en un nombre quelconque d'états, ce qui s'avère plus complexe que le calcul d'une simple constante de normalisation. On est quand même en mesure de calculer, pour un client sortant d'un service dans la file 1, la probabilité $\hat{P}_{p,2}^A$ de perte au niveau de la file 2 :

$$\hat{P}_{p,2}^A = q. \sum_{S/(n_1>0 \text{ et } n_2=K_2)} P(S).$$

On fera donc l'approximation $P_p^A \approx P_{p,1}^A + \hat{P}_{p,2}^A$.

Toutefois, l'obtention de la probabilité de perte devra tenir compte que le flux exogène va être lui-même soumis à des pertes car il provient des autres interfaces. On propose une démarche itérative classique : le taux de départ initialise le processus et permet de calculer une première probabilité de perte, il est alors réduit en conséquence et une nouvelle probabilité de perte est calculée, et ainsi de suite jusqu'à convergence. En général, comme dans [3], l'implémentation montre qu'une telle convergence est assuré pour un nombre d'itérations inférieur à 10 pour une précision relative de 10^{-9}.

Le résultat fondamental énoncé dans [2] est : pour tout K et pour tout K_2 ; et, pour tout état S de la chaîne de Markov, on a :

$$P(S) = \frac{1}{C} \cdot \prod_{i,j} \left(\rho_i^j\right)^{n_i^j},$$

avec la constante C définie par :

$$C = \sum_{n_1=0}^{K} \sum_{n_2^1=0}^{K-n_1} \sum_{n_2^2=0}^{K_2-n_2^1} \binom{n_2}{n_2^1} \cdot \left(\rho_1^1\right)^{n_1} \cdot \left(\rho_2^1\right)^{n_2^1} \cdot \left(\rho_2^2\right)^{n_2^2}.$$

Ce résultat est prouvé par simple vérification des équations de balance. L'obtention de cette constante de normalisation peut nécessiter un lourd travail si le nombre d'états devient important. On utilisera alors les algorithmes de calcul classiques.

4.2.2. Applications

Ce travail sur une classe de réseaux de files d'attente particulière nous a permis de montrer son caractère « forme produit » simple. On est donc en mesure de connaître les probabilités de tous les états, et ainsi de calculer les critères de performances associés à une interface du routeur telle qu'elle est présentée dans la figure 5.

Délai moyen de transit. Si N_1 désigne le nombre moyen de clients dans la file 1 :

$$N_1 = \sum_S n_1.P(S).$$

Si N_2 désigne le nombre moyen de clients dans la file 2 :

$$N_2 = \sum_S n_2.P(S).$$

D'après la formule de Little [10], le temps moyen passé par un paquet de classe A (mode *AS*) dans l'interface est :

$$W_A = \lambda_1^{-1}.N_1.$$

Pour un paquet de classe F ou P (modes *FS* et *PS*), ce délai est :

$$W_F = W_A + (q.\lambda_1)^{-1}.N_2.$$

Probabilité de perte. L'expression de la probabilité de perte pour un paquet s'exprime un peu plus difficilement, du fait de la complexité d'obtention, pour un paquet *arrivant en entrée de l'interface*, de la probabilité de perte dans la file 2. Cette probabilité de perte sera estimée alors par :

$$P_p^A = \sum_{S/(n^1=K)} P(S) + q. \sum_{S/(n_1>0 \text{ et } n_2=K_2)} P(S).$$

5. VALIDATION PAR SIMULATION

5.1. Modèle de simulation

Le modèle analytique permet de fournir des indicateurs de performances, tels que le délai de traversée et le débit maximal. Par contre, l'obtention de la probabilité de perte d'un paquet à l'intérieur même du routeur nécessite, pour l'analyse, l'utilisation de méthodes approchées dont la validité demeure à montrer. La simulation va permettre de mettre en œuvre chacune des interfaces d'un routeur en parallèle, sans faire appel à une aggrégation en un flux exogène. Elle doit aussi vérifier si la méthode itérative utilisée pour tenir compte de la réduction du trafic est valide. Enfin, la modification des paramètres du modèle (taux d'arrivée et taux de service, capacités des files...) doit être aisée.

D'autre part, le modèle présenté en [3] pose comme hypothèse fondamentale la prise en compte de serveurs de type exponentiel. Or, comme les services correspondants concernent des lectures de tables de routage maintenues en différents points de l'équipement, il paraît plus adapté de considérer des serveurs de type déterministe. Dans ce cas, les méthodes analytiques montrent très rapidement leurs limites.

On reprendra donc le modèle d'interface de la figure 5 en remplaçant le flux exogène par des systèmes similaires à celui du couple *(file 0, file 1)*. On cherchera à évaluer les valeurs de probabilités de perte et à les comparer avec celles obtenues par les méthodes analytiques.

5.2. Résultats

5.2.1. Données de simulation

Le modèle de simulation, développé sur l'outil OPNET[1], correspond à un routeur avec $N = 8$ ports d'entrée/sortie. Les valeurs des paramètres utilisées pour le trafic et le routage sont : $\lambda = 180000$ p/s, $\mu_1 = 200000$ p/s, $\mu_2 = 150000$ p/s et $p = 0,9$. Elles représentent un routeur de type « Cisco 7×00 » [14] fonctionnant dans un environnement lui permettant de mettre principalement en œuvre le mode *AS*.

[1] OPtimized Network Engineering Tools ©MIL3.

5.2.2. Services exponentiels

Sur la figure 7 sont portés les tracés obtenus pour $K = 50$ avec la simulation à partir de l'outil OPNET, dans un intervalle de confiance de 90 %, et les résultats de l'analyse forme produit.

5.2.3. Services déterministes

Des simulations avec des services déterministes ont été également menées, car le type de service (consultation de tables) semble plus correspondre, dans la réalité, à un processus déterministe. Les valeurs choisies sont celles données précédemment. Dans le cas $K = 30$, le tracé de la figure 8 donne les résultats de la simulation, pour un intervalle de confiance de 90 % et de l'implémentation des modèles en forme produit.

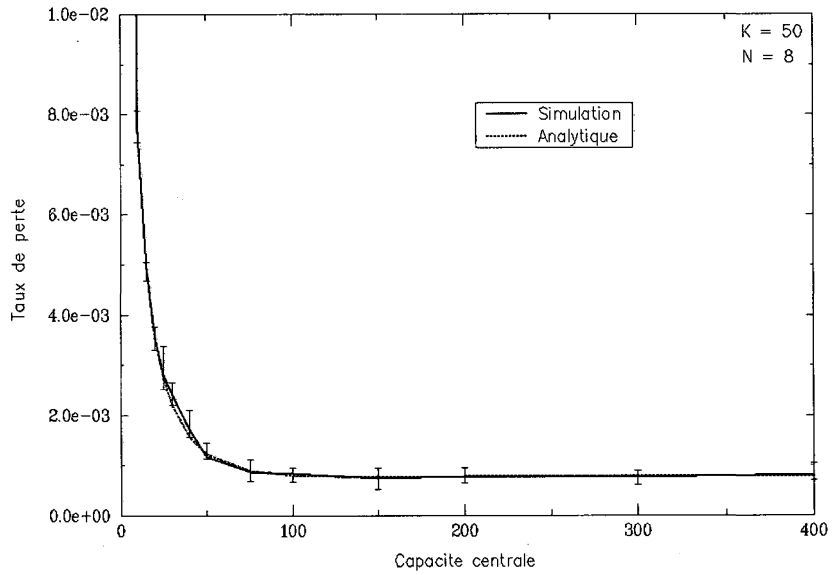

Figure 7. – Probabilité de perte pour $K = 50$.

5.3. Commentaires

Le modèle de réseau mis en place analytiquement fournit, comme on peut le voir au travers des graphiques, d'excellents résultats. Ceci permet d'affirmer que la méthode itérative mise en œuvre pour prendre en compte la réduction du trafic exogène est valable.

Dans ce cadre, simulation et analyse permettent conjointement d'aboutir à plusieurs conclusions. Tout d'abord, l'analyse a tendance à majorer les

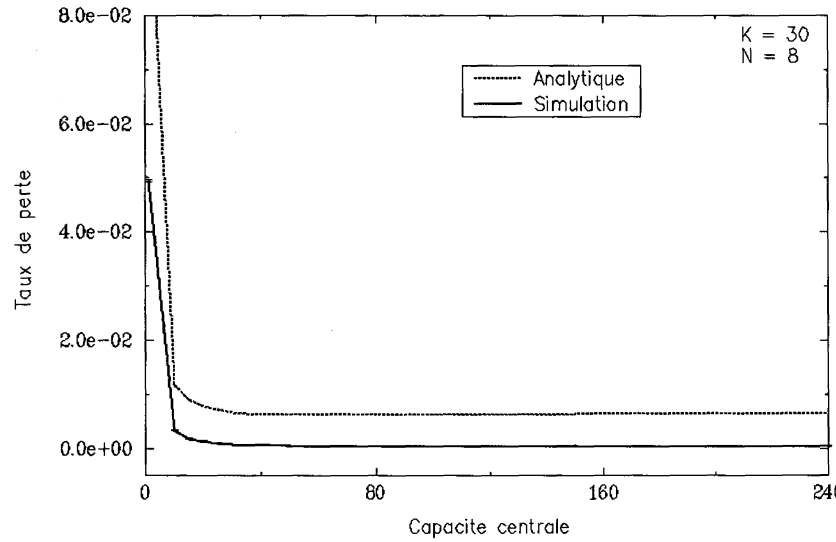

Figure 8. – Probabilité de perte pour $K = 30$ avec service déterministe.

valeurs de probabilités de perte obtenues par simulation (essentiellement pour des valeurs faibles de K). D'autre part, le cas $N = 8$ ports met en évidence un phénomène passé inaperçu dans l'analyse menée en [3], à savoir le passage de la probabilité de perte sous la valeur limite théorique pour des valeurs de K_2 inférieures à $N \times K$. Ainsi, pour $K = 30$, le taux de perte s'élève à $6,1 \times 10^{-3}$ pour $K_2 = 75$ et à $6,3 \times 10^{-3}$ pour $K_2 = 240$, soit un écart de $3\,\%$ qui se retrouve aussi pour de petites valeurs de K (pour $K = 3$, le taux de perte vaut $0,1457$ pour $K_2 = 12$ et $0,1500$ pour $K_2 = 24$, avec $N = 8$ et $p = 0,5$). On peut cependant tenter d'expliquer ce phénomène : pour des valeurs particulières de taux d'arrivée et de service, il semble que les pertes dans la file centrale sont bénéfiques vis-à-vis des pertes dans les files locales. En effet, une perte dans la file centrale entraîne la libération d'une place dans une des files locales, et offre ainsi la possibilité d'accepter à cet endroit un paquet supplémentaire qui sera peut-être servi sans nécessiter de passer par le CPU.

On touche ici à un résultat fort intéressant qui préconise un réglage de capacité centrale à une valeur plus faible que celle que certains utilisateurs seraient tentés de fixer. En effet, dans des équipements tels que les routeurs, le gestionnaire n'a souvent accès qu'au réglage de la mémoire centrale. Il existe donc pour lui un rapport de capacités à trouver pour obtenir les meilleures performances de la part du routeur utilisé.

Enfin, les simulations menées avec des services déterministes montrent, comme on pouvait s'y attendre, que l'approche analytique avec serveurs de type exponentiel fournit des valeurs bien supérieures, comme le montre le tracé de la figure 7. Les modèles analytiques ne peuvent alors constituer qu'un indicateur au « pire cas » et montrent l'intérêt d'une étude des mises à jour des tables cache [4].

Dans les deux cas, on remarque qu'une valeur de taux de perte très proche de la limite théorique est atteinte rapidement. Ainsi, pour $K = 50$, le taux de perte minimal avoisinant 8×10^{-4} est obtenu dès $K_2 = 100$. On pourrait intuitivement croire ce résultat lié à la fois à une forte valeur de p et une forte valeur de K. Mais pour $K = 3$ et $p = 0,5$, les modèles montrent qu'on atteint la limite de 15 % de pertes à moins de 1% près dès $K_2 = 6$. Dans ces études, il apparaît donc dans une première approche qu'une taille de capacité centrale égale à 25 % de la taille limite théorique $N \times K$ est suffisante pour obtenir des taux de pertes quasi-optimaux.

6. CONCLUSION

Un modèle original de routeur a donc été élaboré en tenant compte du fait que le goulet d'étranglement pour les performances de tels équipements se situe au niveau du traitement de l'adresse. Ce modèle analytique, adaptable à des conditions de trafic non équilibrées, permet d'obtenir des estimations en matière de délai de traversée et de débit maximal. En ce qui concerne la probabilité de perte des paquets au sein du routeur, l'étude analytique montre assez rapidement ses limites. Toutefois, les contraintes exprimées au niveau des capacités des files sont abordables par la théorie des réseaux forme produit. Les études de simulation montrent la qualité d'une telle approche qui permet d'obtenir de premiers résultats et des observations pratiques utiles aux gestionnaires des routeurs. Par contre, à l'avenir, des modèles de serveurs plus conformes aux conditions réelles de service (consultation de tables de routage) devront être considérés. Il sera notamment intéressant de coupler un modèle analytique simple comme celui que nous proposons, avec des modèles analytiques de cache [4].

Enfin, ce modèle doit être complété par une étude de trafic, notamment pour déterminer l'importance de chaque classe de trafic au sein d'un trafic réel. Il permet déjà d'offrir un outil d'évaluation des équipements de réseaux de transmission de données fiable et adapté, plus aisé et plus rapide qu'une batterie de tests [5].

RÉFÉRENCES

[1] J. Bashinski, Cisco's AGS+ Router: The architecture that keeps evolving. *The Packet, Cisco Systems users magazine* (1993) **5**.

[2] E. Besson, Modèle de routeur par réseau de files d'attente sous forme produit. Technical report, France Telecom-CNET (1996).

[3] E. Besson, Modélisation analytique d'équipements d'interconnexion au sein d'un réseau de transmission de données. Technical report, France Telecom-CNET (1996).

[4] E. Besson et P. Brown, Performance Evaluation of Hierarchical Caching in High-Speed Routers, in *Proc. of GLOBECOM'98* (1998) 2640–2645.

[5] A. Bichon, E. Besson et M. Carugi, Modélisation et mesures de l'interconnexion de réseaux locaux, cas du délai. Calculs, mesures et résultats. Technical report, France Telecom-CNET (1997).

[6] O.J. Boxma, *Analysis of models for tandem queues*. Ph.D. Thesis, University of Utrecht (1977).

[7] P.J. Burke, The output process of a stationary M/M/S queuing system. *Ann. Math. Statist.* **39** (1968).

[8] J.S. Chen et T.E. Stern, Throughput analysis, optimal buffer allocation, and traffic imbalance study of a generic nonblocking packet switch. *IEEE Journal on Selected Areas in Communications* (1991).

[9] B. Kelly., Cisco Router performance characteristics, in *Networkers'93* (1993).

[10] L. Kleinrock, *Queuing systems Volume I : Theory*. J. Wiley and Sons (1975).

[11] L. Kleinrock, *Queuing systems Volume II : Computer applications*. J. Wiley and Sons (1975).

[12] S.S. Lam, Queuing Networks with Population Size Constraints. *IBM J. Res. Develop.* **21** (1977) 370–37.

[13] The Packet, Cisco 7000 Router brings high applications availability and a platform for the future. *The Packet, Cisco Systems users magazine* **5** (1993).

[14] F. Vanney, ATM et réseaux locaux commutés : la vision d'une entreprise. *Actes du XII^e congrès : DNAC'97*. Laboratoire PRISM (1997) 67-74.