Dimensionnement des buffers pour les routeurs IP de cœur de réseau

Le contrôle de congestion dans l'Internet repose principalement sur les mécanismes implémentés par TCP, auxquels s'ajoute une limite au débit des flots imposée par les débits d'accès. La performance des flots, et notamment du contrôle effectué par TCP, nécessite un dimensionnement correct des buffers au sein des routeurs de cœur de réseau, qui sont gérés en FIFO. La règle empirique de dimensionnement traditionnellement utilisée – dite du *Bandwidth Delay Product* (BDP) – montre ses limites avec l'accroissement constant des capacités des liens, et les contraintes techniques qui limitent la taille des buffers. D'abord mis en évidence par l'article de Appenzeller *et al.* [7], le problème du dimensionnement a depuis suscité une attention croissante.

Dans ce chapitre, nous examinons le problème au vu de notre compréhension des caractéristiques du trafic, et de la performance du partage de bande passante statistique. Nous montrons au moyen d'un modèle analytique simple couplé aux résultats de simulations ns2 que, tandis qu'un buffer de taille équivalente au BDP n'est pas forcément nécessaire, la proposition récente de les réduire à quelques dizaines de paquets est certainement trop drastique. La taille nécessaire pour le buffer dépend de manière significative du débit crête exogène des flots multiplexés.

Sommaire

Publications et présentations :

Ce chapitre à fait l'objet des publications suivantes :

- Jordan Augé, James Roberts, **Buffer Sizing for Elastic Traffic**, *NGI2006, 2nd Conference on Next Generation Internet Design and Engineering, València, April 3-5 2006*
- Jordan Augé, James Roberts, **A Statistical Bandwidth Sharing Perspective on Buffer Sizing**, *ITC'20 – Ottawa, Canada – June 17-21, 2007*

4.1 Introduction

Le dimensionnement des buffers au sein des routeurs a reçu beaucoup d'intérêt récemment [7, 129, 163, 128, 49, 3, 2], avec notamment la remise en cause de la règle empirique dite du *Bandwidth Delay Product* initialement proposée par Villamizer *et al.* [149]. La réalisation de grand buffers est un défi technique pour des liens ayant des débits de plusieurs dizaines de gigabits par seconde. C'est un but pour l'instant inaccessible pour des routeurs purement optiques qui constitueront vraisemblablement la prochaine génération d'équipements. La pertinence de cette règle empirique est remise en cause au vu de l'évolution des capacités des liens de cœur de réseau et du volume de trafic transporté. Il serait par exemple moins coûteux d'accepter de sacrifier un petit pourcentage de l'utilisation, quitte à prévoir un supplément de bande passante.

Qualité de service

Lors des périodes de congestion, le lien est utilisé à 100% et des paquets s'accumulent dans le buffer. Comme nous l'illustrerons dans le chapitre suivant, sans différenciation, la présence de buffers importants est une source de délais et de gigue pour les flots *streaming*, aggravée par les émissions en rafales des flots TCP. Réduire la taille des buffers permettrait de mitiger ces problèmes. Les conséquences d'une telle réduction sur la performance du trafic élastique restent un sujet d'étude, auquel nous consacrons une grande partie de ce chapitre. Néanmoins, un dimensionnement correct de ces buffers n'éliminera pas les pertes de paquets subies par les flots lors des instants de saturation du buffer causés par des rafales.

Avrachenkov *et al.* [9] présentent une analyse des performances de TCP en fonction de la taille du buffer. Une analyse par point fixe à partir d'un modèle assez complexe de TCP conclut sur l'inefficacité de buffers soit trop petits, soit trop grands. Elle est confirmée par des simulations ns-2. Si l'on se réfère à l'explication donnée précédemment pour justifier la règle du *Bandwidth Delay Product*, un buffer trop petit ne permet pas à TCP d'utiliser la totalité de la bande passante disponible (pendant les périodes où il n'émet pas de paquets), et il causera un fort taux de pertes étant souvent saturé. Réciproquement un buffer surdimensionné sera rarement vide et allongera les délais subis par les paquets, d'autant que la majorité des flots qui sont contraints par leur débit d'accès n'en profiteront pas.

La règle du *Bandwidth Delay Product* nécessite de connaître le RTT moyen des flots sur le lien, qui n'est pas facile à caractérise. Les effets d'un sous-dimensionnement étant considérés plus graves (utilisation du lien, équité entre les flots) que ceux d'un surdimensionnement, il est fréquent de prendre une valeur moyenne pour le RTT de 200 à 250ms (de l'ordre du temps de propagation transatlantique).

Conséquences sur la réalisation de routeurs IP

On trouve aujourd'hui des liens OC768 correspondant à un débit de l'ordre de 40Gb/s. En reprenant l'exemple cité dans [7], un lien de 10Gb/s et de RTT moyen 250ms recevra par la règle du BDP une taille de buffer de 2.5Gb. La présence de tels buffers dans les routeurs pose des problèmes technique de réalisation (vu la technologie actuelle des mémoires), d'encombrement et de chaleur. De plus l'évolution constante des capacités des liens remet en cause la pérennité d'une telle approche, et montre l'intérêt d'opérer avec de plus petits buffers. En outre, l'émergence de routeurs tout-optique, pour lesquels on ne sait que réaliser des buffers de quelques dizaines de paquets, requiert de comprendre la performance du trafic en présence de telles tailles de buffers.

A taille de buffer donnée, la performance peut être caractérisée par les délais et taux de pertes de paquets, ou par le débit réalisé par les flots. Elle dépend significativement des hypothèses sur les caractéristiques du trafic. Les articles cités précédemment fondent leur raisonnement sur une grande diversité d'hypothèses de base, ce qui explique leurs conclusions conflictuelles. Notre but dans le présent chapitre est d'identifier les caractéristiques essentielles du modèle au niveau flot, et d'évaluer le compromis entre la taille du buffer et la performance obtenue sous des hypothèses réalistes de trafic.

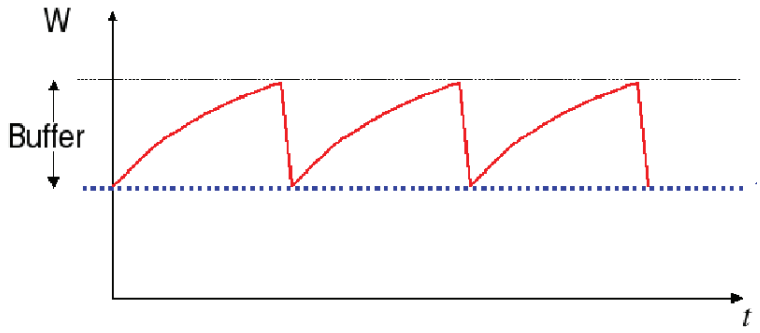

FIGURE 4.1 – Évolution de la taille de la fenêtre de congestion TCP sur un lien avec buffer.

4.2 Dimensionnement des buffers et TCP

Le trafic observé sur un lien de cœur de réseau est majoritairement composé de flots TCP. Il n'est donc pas étonnant que son comportement ait été au centre de nombreuses études et qu'il ait à lui seul dicté le dimensionnement des buffers et entraîné la règle dite du *Bandwidth Delay Product* (BDP). Nous verrons pourtant dans la suite de ce chapitre qu'un tel modèle n'est pas forcément réaliste et qu'il convient de mieux caractériser le trafic et les interactions entre les différents flots afin de pouvoir comprendre la performance réalisée.

4.2.1 Dimensionnement pour un flot TCP

La règle empirique du BDP a été proposée par Villamizar et Song [149] et suggère une taille de buffer calculée en fonction du produit entre le débit du lien multiplié par le délai moyen. L'objectif est d'assurer une utilisation totale du lien par une connexion TCP suffisamment longue [1].

La figure 4.1 [2] représente l'évolution typique de la taille de la fenêtre de congestion d'une connexion TCP seule sur un lien, dans le mode *congestion avoidance*. On y reconnaît le motif caractéristique de TCP en dent de scie dû à l'algorithme AIMD. Les instants de décroissance correspondent à la détection par TCP d'une congestion (perte d'un paquet, reconnue par la réception de trois acquittements identiques). La taille de la fenêtre de congestion *cwnd* est alors divisée par deux, et aucun nouveau paquet n'est émis jusqu'à ce qu'un nombre suffisant d'acquittements soient reçus et que le nombre de paquets en transit soit à nouveau inférieur à *cwnd*. La règle du BDP vise à dimensionner le buffer de telle façon qu'il emmagasine suffisamment de paquets à transmettre pour compenser la période pendant laquelle la connexion n'émettra pas.

Puisque *cwnd* régule l'envoi des paquets pendant chaque RTT, le débit d'une connexion TCP est $cwnd/RTT$. Notons $cwnd_p = cwnd$ lors de la détection d'une perte ; TCP n'émet plus de paquets le temps de recevoir $cwnd_p/2$ acquittements, en raison de la réduction de *cwnd* de moitié, qui arrivent au débit du lien C. La taille du buffer B doit donc être suffisante pour permettre l'émission de paquets au débit C durant cette période : $B \geq cwnd_p/2$ paquets. Le cas idéal correspond à un buffer vide lorsque la source émet à nouveau au débit du lien : $C = cwnd_p/2RTT$. Nous avons $cwnd_p = 2C \cdot RTT$ et ainsi $B = C \cdot RTT$.

4.2.2 Synchronisation des flots TCP : extension à N flots

Le raisonnement que nous venons de présenter est valable lorsqu'un seul long flot TCP occupe la totalité de la bande passante disponible sur le lien. Il se maintient toutefois dans le cas de plusieurs flots TCP simultanés en considérant un phénomène de synchronisation entre les connexions. Sur un lien FIFO/DropTail, les flots ont une tendance à tous subir des pertes au même instant de congestion, ce qui entraine une évolution similaire des fenêtres de congestion au cours du temps. Il suffit alors de considérer la somme de l'ensemble des fenêtres de congestion, qui connait également une évolution en dent de scie, pour justifier l'application du *Bandwidth Delay Product*.

Ce phénomène est généralement exhibé dans des simulations avec des connexions permanentes, mais est moins connu avec du trafic aléatoire (arrivées Poisson de session par exemple). Il est remis en cause dans [7] dans le cas d'un lien de cœur de réseau où de très nombreuses connexions sont en cours et se partagent la capacité disponible (un tel cas est irréaliste sous nos hypothèses de trafic). L'étude

1. c'est-à-dire pour que la connexion sorte du mode *slow start* de TCP en faveur du mode *congestion avoidance*. Les flots courts voient en effet leur débit conditionné principalement par l'évolution de la fenêtre de congestion pendant *slow start*.
2. Cette figure est extraite de la présentation de [7] à la conférence Sigcomm'04.

est approfondie dans [129] afin d'établir le lien entre la synchronisation des flots et le problème de dimensionnement des buffers.

Dans la suite de ce chapitre, nous étudions la performance d'un environnement FIFO/DropTail, mais les résultats obtenus restent compatibles avec l'utilisation d'un ordonnancement *fair queueing*. Ils seront généralement différents avec l'utilisation d'autres mécanismes de gestion du trafic tels que RED, etc. Le chapitre suivant considérera l'impact de l'introduction de nouveaux protocoles TCP, et d'un ordonnancement équitable.

4.3 État de l'art sur le dimensionnement de buffers

La problématique du dimensionnement des buffers à suscité un intérêt croissant suite aux travaux de Appenzeller *et al.* [7]. Nous présentons ici les résultats les plus significatifs, qui peuvent sembler contradictoires à première vue. Nous verrons plus loin qu'ils proviennent en fait d'hypothèses différentes sur le trafic.

De nombreux articles envisagent des approches adaptatives de dimensionnement des buffers, se basant sur l'observation de la complexité et de la variabilité du trafic. Nous nous intéressons à l'allocation physique de mémoire au sein d'un routeur, et non à sa gestion. L'algorithme PFQ, que nous avons présenté dans le chapitre précédent, remplit ce rôle d'allocation et permet une isolation entre les différents flots.

4.3.1 Dimensionnement proportionnel au nombre de flots

La proposition de Morris [111] s'appuie sur la constatation suivante : si une connexion TCP a une taille de fenêtre de congestion inférieure à quelques segments, elle subit de fortes pertes et souffre de fréquents *timeouts* de retransmission qui interrompent le fonctionnement en *congestion avoidance*. Un tel régime dégradé peut être évité si l'on permet à chaque connexion d'émettre quelques segments dans le buffer, et l'auteur recommande ainsi un dimensionnement proportionnel au nombre de flots.

Un buffer suffisamment dimensionné est crucial pour des raisons de performance et d'équité entre les flots. Par exemple, il est possible de limiter le taux de pertes à 2% avec un buffer 6 fois proportionnel au nombre de flots, et à 1% pour un facteur 9.

L'article ne précise toutefois pas quels flots doivent être considérés, et nous pouvons supposer qu'il s'agit de ceux partageant effectivement la bande passante sur le lien, que nous qualifions de *bottlenecked*.

4.3.2 Vers une réduction drastique de la taille des buffers

Appenzeller *et al.* [7] mettent en évidence la difficulté de réaliser des buffers dimensionnés selon le BDP, au vu de l'évolution des débits des liens. Ils questionnent d'autant son utilité sur des liens de cœur de réseau où un très grand nombre de flots sont en compétition.

Ils montrent qu'au delà d'une centaine de flots en compétition, les évolutions des différentes fenêtres de congestion *cwnd* des flots TCP sont désynchronisées, et qu'une taille de buffer proportionnelle à la racine carrée du nombre de flots suffit. Ils remarquent l'intérêt d'estimer l'équité réalisée par les connexions pour des tailles de buffer intermédiaires.

Le phénomène peut se comprendre intuitivement en considérant le comportement en "dents de scie" de l'évolution de *cwnd* : la taille requise par un dimensionnement selon le BDP correspond à la diminution de *cwnd* lors d'une perte. Lorsque les connexions sont nombreuses et désynchronisées, la superposition des différentes fenêtres *cwnd* permet une atténuation du phénomène, qui suggère des besoins moins importants en buffer pour garantir la performance des flots. En supposant que les flots sont identiques, la somme des tailles des fenêtres tend vers une distribution normale (théorème central limite), de laquelle se déduit la proposition de dimensionnement. Appenzeller *et al.* [7] utilise un tel modèle afin d'estimer la probabilité de sous-utilisation du lien.

Les auteurs questionnent également la validité du dimensionnement lorsque les flots sont courts et leur débit conditionné par le *slow start* de TCP. Le modèle utilisé est une file M/G/1, ils montrent alors la dépendance du dimensionnement à la charge et à la longueur des rafales.

L'étude d'un modèle mixte mélangeant flots longs et flots courts est complexe. Une hypothèse avancée et confirmée au travers de simulations par les auteurs est qu'il est suffisant de ne considérer que les longs flots pour le dimensionnement d'un routeur de cœur de réseau. D'autres simulations montrent que les performances avec des flots courts sont meilleurs avec de petits buffers. Il serait intéressant de vérifier ces résultats plus formellement.

Dans la suite, nous qualifierons de telles tailles de buffer de "moyennes".

Quelques remarques sur les modèles de trafic

Il a été remarqué, cependant, par Raina et Wischik [129] ainsi que Raina *et al.* [128] que la synchronisation des flots dépend de la taille du buffer et que, pour un très grand nombre de flots, le dimensionnement proposé dans [7] est toujours trop important. Ils suggèrent que la capacité du buffer doit être réduite à quelques dizaines de paquets. Aucune instabilité n'a été observée dans [7] parce que les auteurs ont seulement effectué des simulations avec quelques centaines de flots alors que le phénomène se manifeste pour quelques milliers. Dans le chapitre 3, nous avons expliqué pourquoi il n'est pas raisonnable de supposer que tant de flots sont en fait *bottlenecked* sur le lien, ce qui nous permet de remettre en cause la validité de cet argument favorable à de petits buffers.

Une publication ultérieure [162] précise que des buffers de 20 paquets suffisent si les paquets émis par les flots sont espacés suffisamment régulièrement (*pacing*). Sinon, il convient plutôt de considérer une taille de l'ordre de 20 rafales. L'accent est mis sur l'instabilité des tailles intermédiaires de buffers.

Nous qualifierons de telles tailles de buffer de "petites" dans la suite du chapitre.

4.3.3 Vers une distinction flots *bottlenecked/non-bottlenecked*

Dhamdhere et al. [3] reconnaissent l'importance de distinguer les flots *bottlenecked* des flots *non-bottlenecked*. Ils suggèrent qu'il est nécessaire d'obtenir de faibles taux de pertes tout en maintenant une utilisation élevée des liens. En conséquence, ils recommandent d'avoir un buffer relativement grand qui est proportionnel au nombre de flots *bottlenecked*. Les auteurs poussent plus en avant leur analyse dans [2], notamment en introduisant des modèles de trafic dynamiques au niveau flot, *open* et *closed loop* qui correspondent à notre notion de partage statistique de bande passante.

Cependant le modèle dans [3] est annoncé valide lorsque les flots *bottlenecked* constituent plus de 80% de la charge du lien et la performance est évaluée dans [2] dans un cas de très forte charge (où environ 200 flots *bottlenecked* sont en compétition). Une nouvelle fois nous remettons en cause la pertinence de ces hypothèses sur le trafic en vue d'évaluer les besoins en buffer.

4.3.4 Cas où les flots sont tous *non-bottlenecked*

Le modèle proposé par Enachescu *et al.* [49, 50] évalue la taille de buffer nécessaire lorsque les capacités des réseaux d'accès et de cœur sont d'ordres de grandeur différents : les paquets se retrouvent alors naturellement espacés lors de leur transmission. Il est également possible de recourir à des piles TCP modifiées qui espacent les paquets au lieu de les envoyer par rafales (*pacing*).

Lorsque les paquets sont espacés au débit moyen déterminé par la taille de la fenêtre plutôt qu'émis en rafales, [50] suggère que la taille du buffer doit être proportionnelle au logarithme de la fenêtre de congestion maximale de TCP.

Cette hypothèse est licite pour de nombreux liens et correspond à ce que nous appelons le régime transparent (Ch.2, Sec.2.2.3) : tous les flots sont *non-bottlenecked*. Nous remarquons que les analyses dans [129, 128] se basent sur un modèle fluide et supposent ainsi implicitement que les paquets n'arrivent pas par rafales.

4.4 Impact de la taille du buffer sur la performance des flots

4.4.1 Régimes opérationnels

Si l'utilisation globale du lien est préservée pour des buffers de taille moyenne, ou en grande partie pour de petites tailles de buffers, nous ne savons rien de la performance individuelle de chaque flot : un débit satisfaisant et équitable pour les flots élastiques, ainsi que des délais et taux de pertes négligeables pour les flots *streaming*.

En fonction de la charge offerte et des débits des flots considérés, le lien se retrouvera dans l'un des trois régimes de bande passante, que nous avons introduits dans le Chapitre 2 :

transparent : La somme des débits des flots est inférieure à la capacité du lien, et il est possible de dimensionner le buffer afin de ne gérer que les arrivées simultanées de paquets (multiplexage des flots sans buffer). En assurant des délais et taux de pertes négligeables à l'ensemble du trafic, on garantit également la performance des flots élastiques. L'ensemble du trafic conserve son débit exogène à la traversée du lien.

élastique : La taille nécessaire des buffers doit être évaluée pour un mélange de flots *non-bottlenecked* – pour lesquels les paquets sont espacés au débit du flot – et de flots *bottlenecked* – qui généralement émettent leurs paquets par rafales – correspondant à un mélange caractéristique d'une charge réaliste sur un lien de cœur de réseau. Il est notamment important de tenir compte des émissions par rafales des paquets appartenant aux connexions TCP.

surcharge : il s'agit d'une situation anormale devant être gérée, par exemple, par un contrôle d'admission ; nous ne nous intéresserons pas à ce régime pour le dimensionnement des buffers.

4.4.2 Performance des flots *streaming*

Nous supposons que la charge des flots *streaming* reste inférieure à la capacité du lien. Cette hypothèse est réaliste dans des conditions normales pour le réseau, et garantie par l'utilisation d'un mécanisme de contrôle de la surcharge, tel qu'un contrôle d'admission.

Lorsque le lien n'est pas saturé, en régime transparent, le trafic ne s'accumule pas dans le buffer et les flots concurrents n'ont qu'un impact négligeable sur la performance, pourvu que le buffer soit suffisamment dimensionné pour absorber les arrivées simultanées.

Lorsque des flots possèdent un débit crête suffisamment élevé et/ou sont en nombre suffisant pour saturer le lien (régime élastique), le buffer se remplit et la performance des flots *streaming* se retrouve dégradée :

- ils subissent des pertes aux instants où le buffer est plein (ces instants sont plus ou moins longs en fonction de la durée des rafales TCP).
- ils subissent des délais importants, ainsi que de la gigue produite par les oscillations du buffer (voir [129]).

Nous présentons des simulations de telles situations dans le Chapitre 5, et montrons l'intérêt d'introduire un ordonnancement de type *fair queueing* pour y remédier : les flots se retrouvent isolés, et leur performance est alors indépendante de la taille du buffer.

4.4.3 Performance des flots élastiques

Les modèles qui considèrent un très grand nombre de flots permanents, comme par exemple [7], ne correspondent pas à ce que l'on peut observer sur un lien de cœur de réseau à un niveau de charge nominal. Nous considérons un modèle de trafic réaliste tel que décrit dans le chapitre précédent.

La majorité des flots ne sont pas *bottlenecked*. Ces flots émettent des paquets de temps à autre ; ils transitent dans le buffer lorsque ce dernier est saturé, et ne s'y accumulent généralement pas si le lien n'est pas surchargé. La participation de ces flots à l'allocation de bande passante faite par TCP est négligeable. Ces flots conservent leur débit exogène, à l'exception des instants où ils subissent des pertes.

Ces sont les flots de plus fort débit (leur nombre dépend de la charge du lien) – qui accumulent un grand nombre de paquets dans le buffer et causent des pertes – qui réalisent un partage des ressources. Une première approximation est de considérer qu'ils utilisent la capacité laissée disponible par les autres flots. Cette hypothèse est pratique, et donne de bons résultats dans la mesure ou les instants de pertes restent des événement rares. Elle devient réaliste avec l'utilisation de *fair queueing*. Il est alors possible de considérer que les flots *non-bottlenecked* constituent un trafic de fond de charge donnée [3].

Une analyse de l'équité du partage entre les flots est faite dans le Chapitre 5. Nous verrons que l'utilisation de *fair queueing* permet un multiplexage efficace des flots, et garantit leur isolation et l'équité de leurs débits.

Dans la suite de l'évaluation, nous acceptons comme dans [7] de sacrifier une fraction du taux d'utilisation du lien si ce choix permet une diminution importante de la taille des buffers : une telle décision peut être économiquement rentable et permettre la conception de routeurs de capacité plus importante. Nous allons pour les régimes transparent et élastique examiner si la règle du BDP reste de rigueur ou non.

4.5 Dimensionnement des buffers dans le régime transparent

Un régime transparent à été défini plus tôt tel que la somme des débits crête des flots reste inférieure à la capacité du lien avec une forte probabilité. Le buffer doit être dimensionné de telle sorte à absorber des arrivées de paquets simultanées, provenant de flots indépendants. Nous supposons que le débit

3. On peut supposer que la charge induite par les flots *streaming* n'est pas affectée par les pertes qu'ils subissent (modèle *open loop*), et que les pertes sont faibles et distribuées sur un grand nombre de flots pour les flots élastiques de faible débits, d'où une influence négligeable sur la charge.

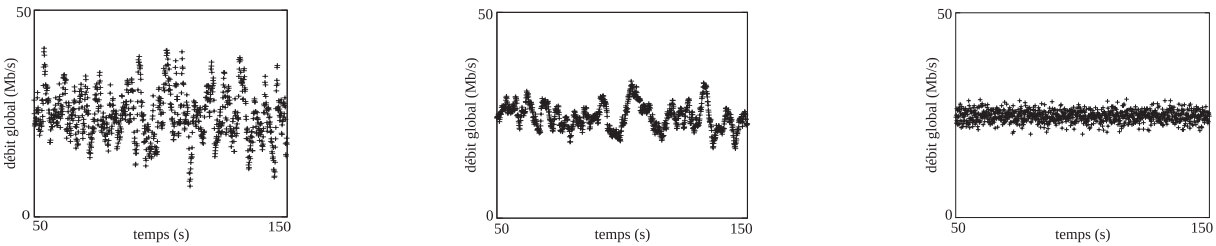

FIGURE 4.2 – Processus d'arrivée des paquets : débit des paquets arrivant dans des intervalles successifs de 100ms pour des flots de débit crête $p = 200$Kb/s (à gauche), $p = 50$Kb/s (au milieu) et $p = 0$ (à droite).

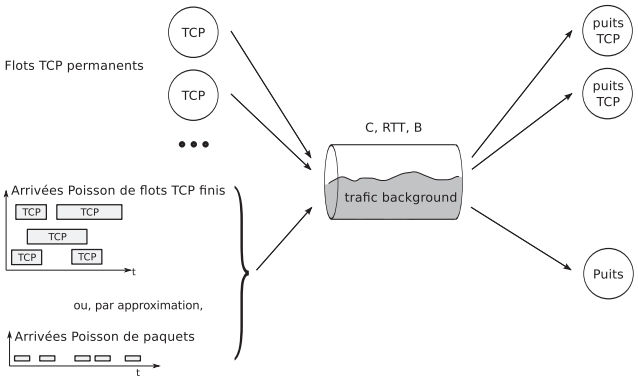

FIGURE 4.3 – Environnement de simulation : sauf précision supplémentaire, les simulations utilisent les paramètres suivants : $C = 50$ Mb/s, $B = 20$ paquets, $RTT = 100$ ms, $\rho_b = 0.5C$, ordonnancement FIFO, paquets de 1000 octets.

crête des flots est bien défini à l'échelle de temps de l'émission des paquets comme, par exemple, lorsqu'il est limité par un lien d'accès en amont.

4.5.1 Arrivées localement Poisson

La figure 4.2 décrit le débit global d'une superposition de flots TCP Reno ayant un débit crête limité, en utilisant l'environnement de simulation présenté en figure 4.3. Sans trafic *bottlenecked*, les flots arrivent selon un processus de Poisson et leur taille est tirée aléatoirement depuis une distribution exponentielle de moyenne 100 paquets. Les figures représentent le débit moyen dans des intervalles successifs de 100ms. Le débit est montré pour deux débits crête, $p = 200$Kb/s et $p = 50$Kb/s, ainsi que celui mesuré pour pour des arrivées de paquets selon un processus de Poisson ($p = 0$).

Un processus d'arrivée de paquets Poisson n'est visiblement pas une bonne approximation à moins que le débit crête des flots ne soit relativement faible par rapport au débit du lien. Cependant, dans un petit intervalle de temps (par exemple dans chaque intervalle de 100ms), le processus d'arrivée des paquets, en tant que superposition d'un grand nombre de processus périodiques, est approximativement Poisson. La figure présente une réalisation de ce processus de Poisson modulé. Nous notons son intensité Λ_t.

Une telle supposition permet d'approximer l'occupation du buffer localement par celle d'une file M/G/1. Si l'on simplifie d'avantage en supposant des tailles de paquets exponentielles, la probabilité de perte d'un paquet étant donné un buffer B est approchée par la formule $(\Lambda_t/C)^B$.

4.5.2 Calcul de la taille requise du buffer

Un approche pour le dimensionnement est de calculer un taux de perte moyen en conditionnant sur la distribution $F(\lambda)$ de Λ_t, avec pour objectif que $\int(\lambda/C)^B dF(\lambda) < \epsilon$. C'est un choix raisonnable quand les variations de débits sont rapides de telle sorte que ϵ est une bonne mesure de la performance d'un flot choisi arbitrairement. Il s'avère que, pour un débit crête inférieur à $.1C$ et $\epsilon > .001$, la taille requise en buffer est la même que celle qui aurait été nécessaire pour un processus d'arrivée Poisson. En d'autres termes, la formule de la M/M/1 $\rho^B < \epsilon$ demeure un critère de dimensionnement utile. Par

exemple, un buffer de 20 paquets limite la charge admissible à $\rho = .79$ pour $\epsilon = .01$ ou $\rho = .7$ pour $\epsilon = .001$.

Wishik [162] propose une autre justification pour cette hypothèse Poissonnienne, qui peut sembler contradictoire avec la présence généralement acceptée de dépendance à long terme au sein du trafic Internet. D'après les auteurs, ce phénomène ne se manifeste vraiment qu'à de plus grandes échelles de temps, alors qu'ici on s'intéresse à une granularité plus faible. En effet, de petits buffers seront rapidement saturés, et les variations de la file d'attente seront très rapides, comparativement à la charge du lien (plusieurs RTT). Le taux de perte sera ainsi très proche de celle d'une file connaissant des arrivées Poisson.

4.6 Dimensionnement des buffers pour le régime élastique

Il n'est pas possible dans le cas général de garantir que le débit crête des flots est limité sur un lien. Même les réseaux d'accès ADSL transportent des tunnels agrégeant d'autres connexions. Il se peut alors que certains flots saturent (temporairement) le lien et voient leurs paquets stockés dans le buffer. Le taux de pertes de paquets est alors fortement dépendant de la taille de ce buffer. Nous considérons une charge de lien inférieure à 1 (régime stable).

Il est alors nécessaire de comprendre l'impact de la taille des buffers sur la performance du régime élastique – c'est-à-dire lorsqu'un ou plusieurs flots *bottlenecked* se combinent avec la charge *background* pour saturer momentanément le lien pendant des périodes qui sont longues comparées à l'échelle de temps d'émission des paquets. Le comportement de chaque type de flot est différent, et leur performance est mal connue dans le cas général.

Les flots *non-bottlenecked* n'atteignent pas le débit équitable, notamment à cause des contraintes en débit qu'ils subissent, mais aussi à cause de pertes dues à la saturation du buffer. Si ces pertes ne se produisent que de temps en temps (aux instants de saturation) et n'affectent qu'un petit nombre de flots, elles sont toutefois néfastes pour ces flots qui ont déjà un débit restreint. Elles seront d'autant plus importantes que les protocoles de transport mis en œuvre pour le transfert des flots à fort débit seront agressifs. Nous envisagerons ce cas dans le prochain chapitre.

En ce qui concerne les flots *bottlenecked*, nous allons voir que la présence de trafic concurrent (*background*) n'est pas sans conséquence sur le débit réalisé. Nous commençons par analyser quelques résultats simples de simulation, avant de proposer une étude plus exhaustive basée sur un modèle PS. Cette dernière nous permettra de proposer des recommandations pour un dimensionnement raisonnable des buffers.

4.6.1 Comportement des flots *bottlenecked*

Environnement de simulation

Afin de simplifier l'analyse et la discussion, nous supposons une stricte dichotomie entre ces deux classes de flots. Les flots *non-bottlenecked* représentent une charge *background* (ρ_b) qui est par hypothèse fixe (régime quasi-stationnaire), le reste de la bande passante est partagé par les flots *bottlenecked*. Les flots *non-bottlenecked* sont en grand nombre et possèdent chacun au maximum un paquet dans le buffer. Si leur taux de perte n'est pas trop élevé, la charge offerte au lien ne sera pas trop impactée. La présence de *fair queueing*, que nous considérons dans le chapitre suivant, permettra de rendre cette hypothèse plus robuste.

Le scénario de simulation s'appuie sur une topologie dite *dumbbell* (Fig. 4.3) avec un lien central de 50Mb/s et un RTT égal à 100ms. Les flots *non-bottlenecked* sont des flots TCP de taille finie et de débit égal à 1Mb/s, arrivant selon un processus de Poisson. Cependant, afin de faciliter l'évaluation d'un grand nombre de configurations, nous avons remplacé le processus au niveau flot (qui résulte en un trafic *background* de débit variable) par un processus d'arrivée de paquets Poisson de même intensité. Les résultats présentés ici ne sont pas affectés par cette simplification.

Dans les simulations qui suivent, le trafic *background* occupe la moitié de la capacité disponible ($\rho_b = 0.5$). Nous avons simulé 1, 2 et 4 flots TCP NewReno permanents, chargés de représenter différents états du lien en régime quasi-stationnaire. En effet, sur une période grande par rapport à l'échelle de temps des variations, le nombre de flots peut être considéré constant, et TCP est supposé atteindre rapidement un régime stationnaire. La performance des flots est analysée lorsque l'on fait varier la taille du buffer, entre 20 paquets (la valeur recommandée dans [129]) et un dimensionnement selon le *Bandwidth Delay Product*. Ces quelques simulations nous permettent déjà de tirer un certain nombre de remarques sur la performance des flots.

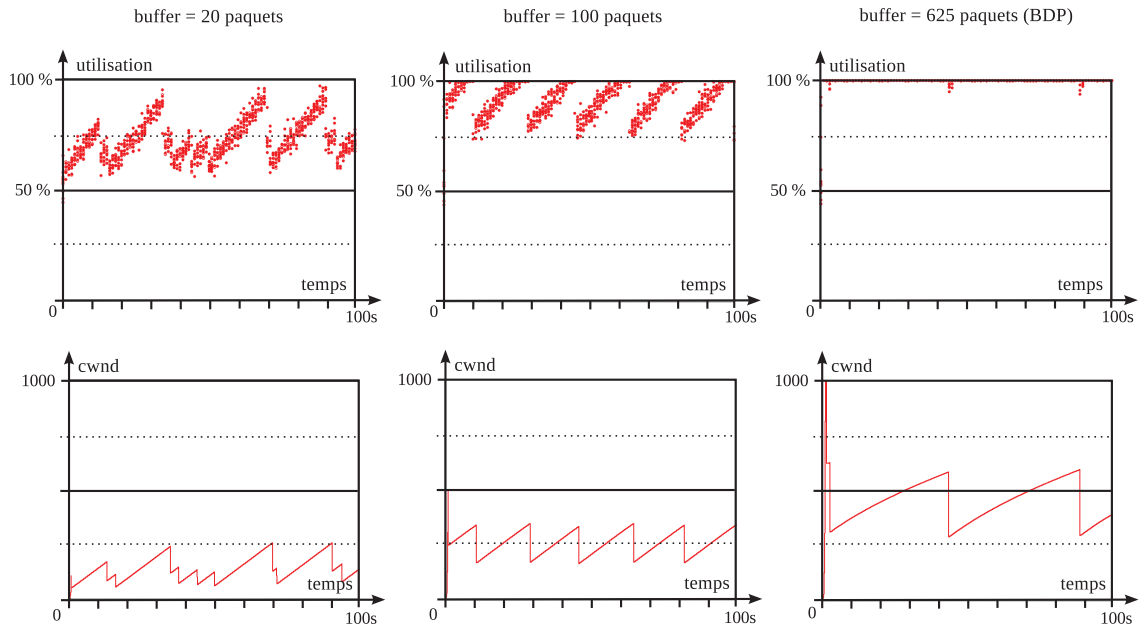

FIGURE 4.4 – De gauche à droite, taux d'utilisation et taille de la fenêtre de congestion (*cwnd* en fonction du temps, pour différentes tailles de buffer : 20, 100 and 625 paquets (BDP).

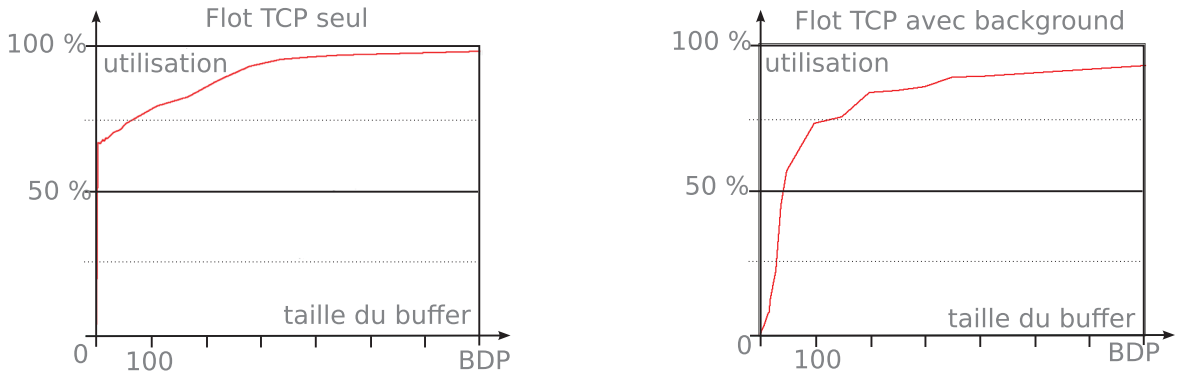

FIGURE 4.5 – Comparaison de l'utilisation de la bande passante disponible par un flot TCP, avec ou sans trafic concurrent

Impact du trafic *background* sur le comportement d'un flot TCP

La figure 4.4 illustre l'impact de la taille du buffer B sur l'évolution de la fenêtre de congestion *cwnd* (graphe du bas), ainsi que l'utilisation du lien moyennée sur un RTT (graphe du haut) pour un seul flot *bottlenecked*. Nous présentons des résultats pour trois tailles de buffers : petite (20 paquets, [129]), grande (625 paquets, correspondant au *Bandwidth Delay Product*), ainsi qu'une taille intermédiaire de 100 paquets.

Les résultats pour $B = 20$ montrent clairement que ce choix est inadapté au modèle de trafic considéré. Si l'on s'attend à ce que le flot n'utilise pas toute la capacité laissée disponible, l'importance de la dégradation de performance peut surprendre. Un flot TCP seul sur un lien vide disposant d'un buffer de 1 paquet devrait acquérir un débit voisin de 75% de la capacité disponible sur le lien (avec un raisonnement identique à celui permettant d'établir le BDP). Et nous nous attendons à ce que cette valeur dépasse les 80% pour un buffer de 20 paquets. Nous avons vérifié expérimentalement ces valeurs, en simulant un flot TCP seul sur un lien à 25 Mb/s, ainsi qu'un autre flot sur un lien à 50Mb/s où 50% de la charge est constitué de trafic *background*. Les résultats sont représentés sur la figure 4.5. On remarque que la présence de trafic *background* réduit considérablement le débit réalisé jusqu'à seulement 40% environ de la bande passante résiduelle.

D'autres simulations avec une charge *background* différente montrent que le taux d'utilisation de la

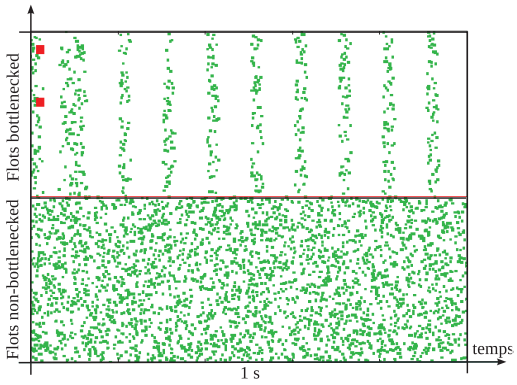

FIGURE 4.6 – Représentation des arrivées de paquets TCP NewReno sur un lien avec un buffer de 20 paquets. Chaque point est un paquet ; sa position verticale est aléatoire ; les gros carrées sur la gauche représentent des paquets perdus.

capacité résiduelle décroît avec l'augmentation de cette charge.

La raison est que le trafic *background* en compétition se combine avec les rafales des flots TCP *bottlenecked* pour saturer momentanément le lien. Nous illustrons ce phénomène dans le cas d'un seul flot *non-bottlenecked* en figure 4.6. Chaque point dans la moitié inférieure représente un paquet émis par un flot *background*, tandis que ceux de la moitié supérieure appartiennent au flot *bottlenecked*. La position sur l'axe des abscisses indique l'instant d'émission et celle sur l'axe des ordonnées est choisie aléatoirement. L'apparition de bandes reflète le mécanisme d'émission de la fenêtre TCP, qui malgré l'auto-régulation par le mécanisme d'acquittements, tend à émettre les paquets en rafales. Ces bursts sont de deux natures : des "micro-*bursts*" dus aux paquets émis dos à dos lorsque la fenêtre de congestion croît, qui possèdent souvent un débit supérieur au débit équitable (et d'autant plus fréquents en *slow start*) [6] ; ainsi que des "rafales sub-RTT", dont le débit moyen respecte le débit équitable, mais qui sont émises pendant un intervalle plus court que le RTT [77].

Tant que la fenêtre TCP reste petite comparée au *Bandwidth Delay Product* résiduel $C(1-\rho_b)\times$ RTT, ces rafales sont émises à partir d'instants séparés par un RTT. L'auto-ajustement de TCP fait que la somme des débits des rafales et du débit du processus *background* est très proche de la capacité du lien, voire supérieure. L'occupation du buffer a ainsi tendance à croître dans ces conditions de forte charge pendant que la rafale est en cours d'émission, puis à décroître quand le lien est à nouveau uniquement en présence d'arrivées du processus *background*.

En l'absence de perte, TCP augmente *cwnd* de 1 paquet par RTT, prolongeant ainsi la durée de la prochaine période de surcharge. A moment donné, les arrivées combinées des flots *background* et *bottlenecked* se combinent pour saturer le buffer et causer la perte d'un paquet. Pour un buffer de 20 paquets, cet événement se produit assez souvent, même pour de très faibles valeurs de *cwnd*. Dans la figure, la perte de paquet se produit à la fin de la première rafale que nous avons représentée. Elle est détectée un RTT plus tard ce qui mène à la diminution de moitié de la taille de la fenêtre courante. Sans la présence de trafic *background*, la perte ne se serait produite que lorsque *cwnd* aurait excédé le *Bandwidth Delay Product*.

Ces premiers résultats montrent que le protocole TCP majoritairement utilisé de nos jours n'est pas adapté à la présence de petits buffers dans le réseau. Le flot sort de sa période *slow start* prématurément, et la phase *congestion avoidance* sera fortement pénalisée par les pertes de paquets. L'inefficacité du protocole sera accentuée par la lenteur de la croissance de la fenêtre de congestion dans l'algorithme AIMD après une perte. Le choix d'un grand buffer nous permet ici d'assurer une utilisation totale du lien.

Une taille intermédiaire de 100 paquets apparaît comme un compromis raisonnable, à la fois au vu du débit réalisé mais aussi parce qu'elle permet d'assurer de plus faibles délais au trafic *streaming* potentiellement multiplexé avec les autres flots *background*.

Multiplexage de plusieurs flots *bottlenecked*

La figure 4.7 illustre la performance obtenue lorsque le nombre de flots *bottlenecked* multiplexés augmente. La taille du buffer est toujours de 20 paquets. On observe que même en présence de seulement deux flots, il y a très peu de pertes synchronisées et l'utilisation du lien s'améliore avec le nombre de flots. L'évolution de *cwnd* pour chaque flot montre que la bande passante est partagée approximative-

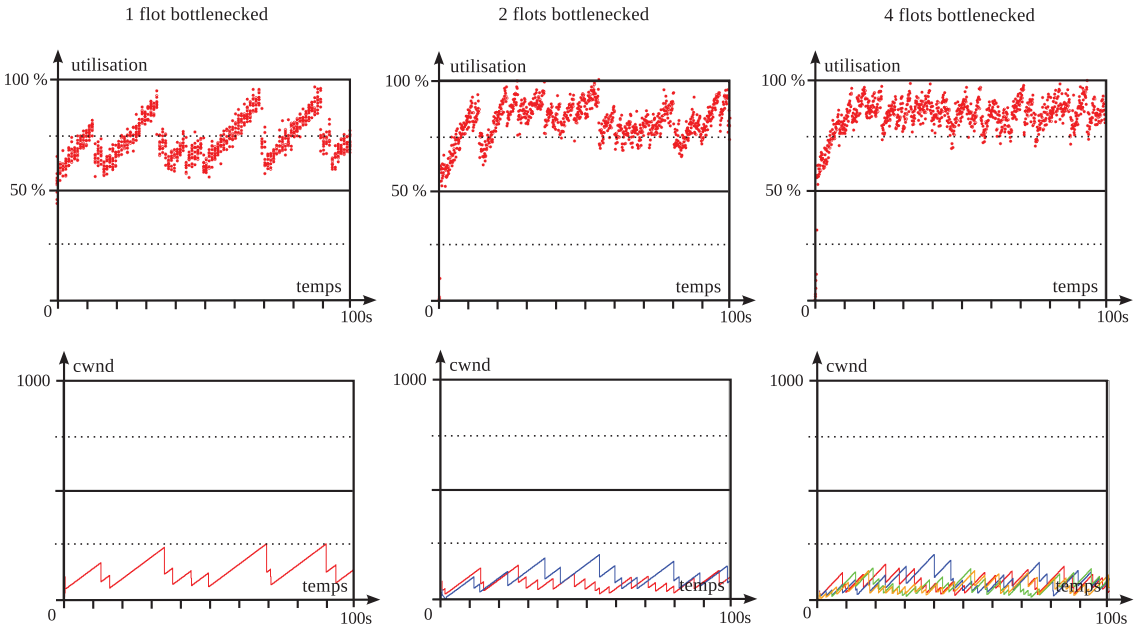

FIGURE 4.7 – De gauche à droite, utilisation et taille de fenêtre *cwnd* en fonction du temps pour 1, 2 et 4 flots *bottlenecked* avec un buffer de 20 paquets. Tous les flots utilisent TCP NewReno et l'ordonnancement est FIFO.

ment équitablement.

Le modèle de trafic considéré prévoit un nombre de flots en compétition réduit à quelques unités, le cas d'un seul flot étant à la fois le plus probable et le pire cas. A la lumière de ces résultats, une taille de buffer réduite à 20 paquets n'est pas justifiée, et des tailles plus importantes – comme suggérées par le scénario B=100 – doivent être considérées.

La suite de cette section approfondit l'étude des phénomènes causés par des petits buffers, en étendant notamment le modèle de trafic à des cas plus réalistes. Nous tentons d'une part de comprendre l'impact de buffers réduits (qui peuvent être une contrainte technique, par exemple pour des buffers optiques), et d'autre part de proposer un dimensionnement des buffers permettant de préserver la performance des flots élastiques. Une taille plus importante permettra d'absorber les fluctuations causées par le processus aléatoire d'arrivée des paquets *background*, permettant ainsi une évolution satisfaisante de la fenêtre de congestion des flots TCP.

4.6.2 Flots *bottlenecked* au débit crête non limité

Méthode

Afin d'évaluer le débit des flots, nous procédons comme suit. Pour une capacité de lien, une taille de buffer et une charge *background* donnée, nous simulons successivement un nombre de flots TCP *bottlenecked* permanents. Pour chaque nombre i (entre 1 et 500, ce qui permet une estimation suffisamment précise), nous évaluons le débit global réalisé $\phi(i)$ exprimé en tant que fraction de la capacité résiduelle $C(1-\rho_b)$. Nous dérivons alors l'espérance du débit des flots γ par la formule 3.2 présentée dans le Chapitre 3, Section 3.3.2. Cela correspond à une analyse quasi-stationnaire qui nous permet d'ignorer des phénomènes tels que limitations de débit dues au *slow start* et les iniquités temporaires entre les flots. Il s'agit de trouver la probabilité stationnaire du système représenté en figure 4.8, et de calculer l'espérance du nombre de flots.

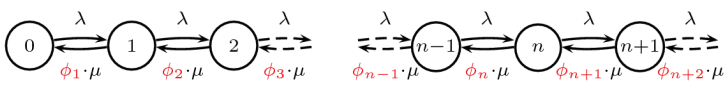

FIGURE 4.8 – File d'attente représentant le nombre de flots en cours, pour un régime élastique régi par TCP

Les figures 4.9, 4.10, 4.11 représentent les valeurs de $\phi(i)$ et γ en fonction de la charge du lien ρ

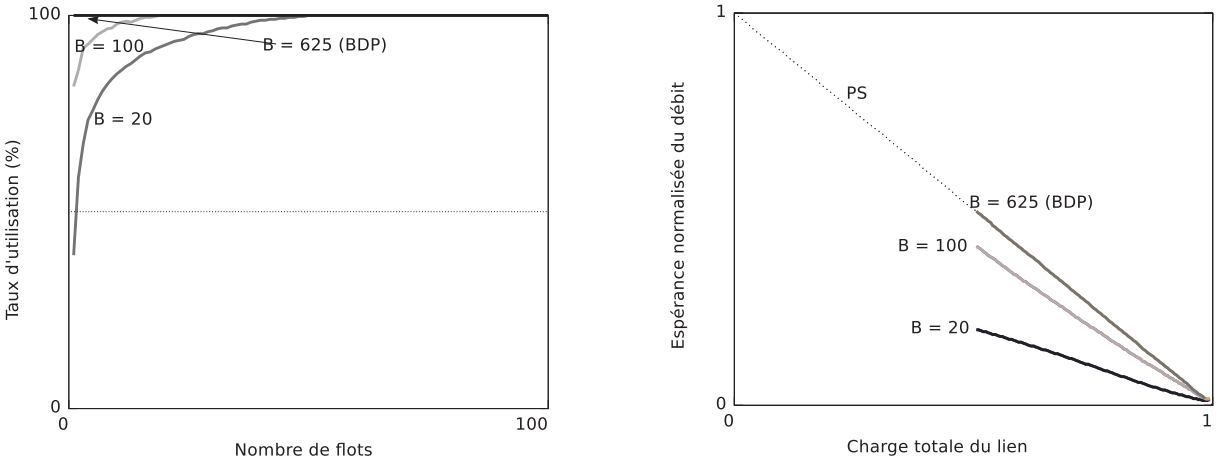

FIGURE 4.9 – Taux d'utilisation de la capacité résiduelle $\phi(i)$ atteint pour chaque nombre i de flots en cours ; et espérance du débit γ d'un flot en fonction de la charge ρ, **pour différentes tailles de buffer**

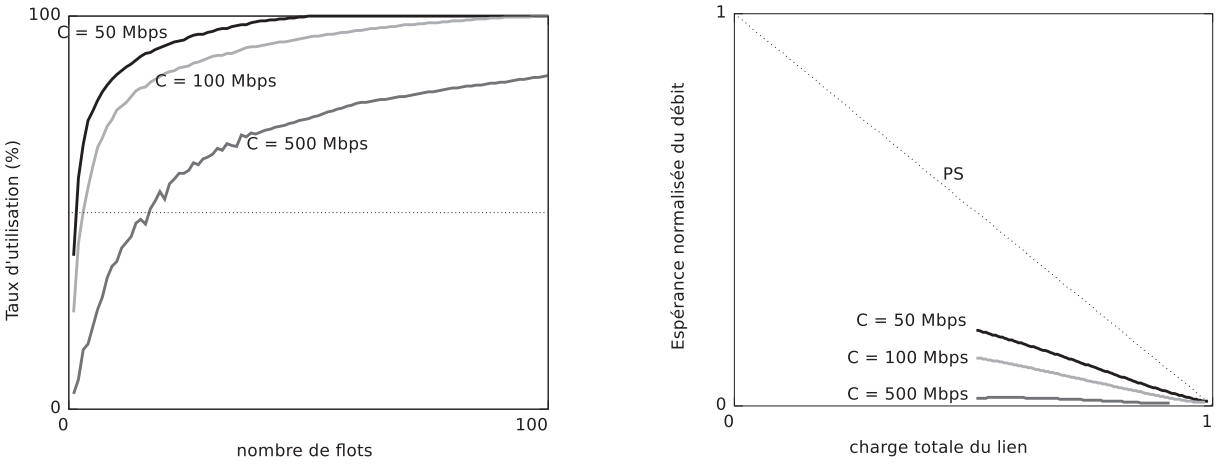

FIGURE 4.10 – Taux d'utilisation de la capacité résiduelle $\phi(i)$ atteint pour chaque nombre i de flots en cours ; et espérance du débit γ d'un flot en fonction de la charge ρ, **pour différentes capacités de lien**

pour un ensemble de configurations. Il convient de noter que γ est seulement défini pour des charges supérieures à la charge *background* ρ_b et que sa valeur pour cette charge est déterminée par $\phi(1)$, débit résiduel utilisé par un seul flot *bottlenecked*.

Exprimons le débit moyen γ d'un flot dans le cas d'un partage équitable [26] :

$$\gamma = \frac{\rho}{E[X]}$$

où $E[X]$ représente l'espérance du nombre de flots en cours. Nous cherchons à déterminer la valeur de γ lorsque la charge tend vers 0. Numérateur et dénominateur tendant vers 0, nous pouvons appliquer le théorème de l'Hôpital :

$$E[X](\rho) = \pi_0(\rho) \cdot \sum_{i=1}^{\infty} \frac{i\rho^i}{\prod_{j=1}^{i} \phi_j}$$

$$\frac{dE[X](\rho)}{d\rho} = \pi_0 \cdot \sum_{n=1}^{\infty} \frac{i^2 \cdot \rho^{i-1}}{\prod_{j=1}^{i} \phi_j}$$

qui tend vers $\frac{1}{\phi_1}$ en 0 ($\pi_0(0) = 1$). γ tend donc vers ϕ_1 pour une charge nulle. C'est en effet le débit qu'aurait un flot arrivant dans un système vide.

Nous avons expliqué plus tôt les raisons causant une dégradation de la valeur de $\phi(1)$, qui détermine également la forme de la courbe (elle est approximativement linéaire dans le cas d'un partage équitable entre les flots). Elle décroît de sa valeur maximale atteinte en $\rho = \rho_b$ vers 0 pour $\rho = 1$.

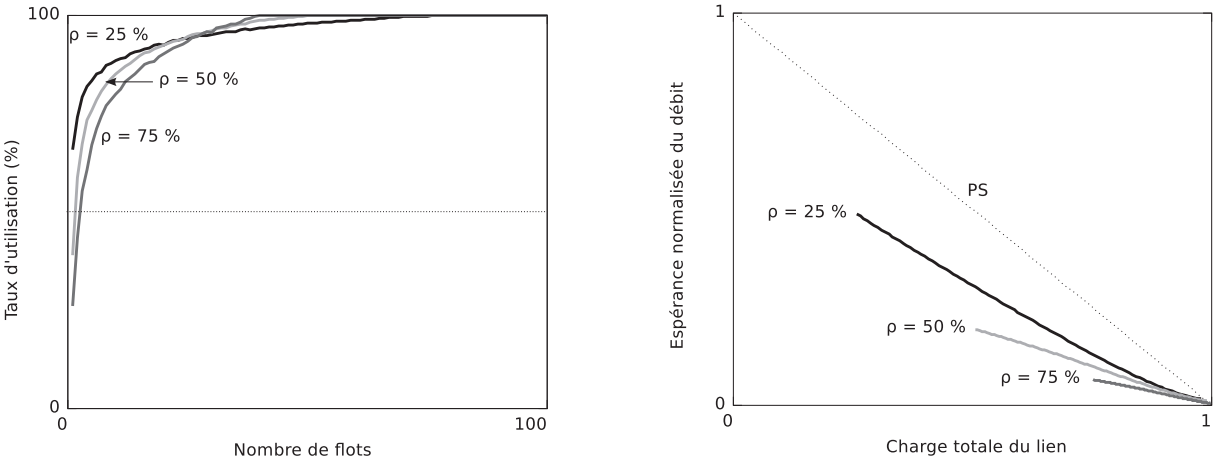

FIGURE 4.11 – Taux d'utilisation de la capacité résiduelle $\phi(i)$ atteint pour chaque nombre i de flots en cours ; et espérance du débit γ d'un flot en fonction de la charge ρ, **pour différentes charges de trafic *background***

Les résultats montrent qu'il y a une chute significative du débit avec de petits buffers (fig. 4.9) et que cette perte est encore plus marquée quand la capacité du lien augmente (fig. 4.10). Plus la charge *background* est importante, plus il est difficile pour TCP d'utiliser la bande passante résiduelle (fig. 4.11).

Débit moyen d'un flot *non-bottlenecked* en fonction de la taille du buffer

Nous avons vu que la connaissance de $\phi(1)$ détermine la performance du système pour des conditions données de capacité de lien, de taille du buffer et de charge (charge *background* + charge due aux flots TCP). Si peu d'articles évaluent la performance d'une connexion TCP dans les conditions que nous avons présentées, [80] propose toutefois un modèle analytique de TCP Reno dans un tel contexte, lorsque le trafic *background* suit un processus D-BMAP (*Discrete Batch Markov Arrival Process*). La complexité de ce modèle ne permet pas d'estimer facilement l'impact de la taille du buffer sur la performance de TCP. De plus, nous ne possédons pas de tels résultats pour les autres protocoles que nous étudierons dans le chapitre suivant ; c'est pourquoi nous nous contentons de simulations dans le reste de cette section.

La figure 4.12 représente le produit $\phi(1)C(1-\rho_b)\texttt{RTT}$ comme une fonction de la taille du buffer. Cette grandeur représente le taux d'utilisation de la bande passante résiduelle renormalisé par le BDP de cette même capacité résiduelle. Trois configurations sont représentées ; la capacité du lien et le RTT varient (C=50Mb/s et RTT=100ms ; C=100Mb/s et RTT=50ms ; C=100Mb/s et RTT=100ms) ; la charge du trafic *background* reste la même (50%). Dans les deux premiers cas, la valeur du BDP est identique (2.5Mb) ; dans le troisième elle est le double (5Mb). On constate que la courbe d'utilisation croît d'abord fortement avec l'augmentation de la taille du buffer, puis subit un point d'inflexion pour converger vers une utilisation complète de la capacité résiduelle (une ligne horizontale sur la figure). Dans cette représentation, on note que les flots qui subissent le même BDP ont le même comportement.

Si l'on représente des configurations similaires en faisant varier la charge *background*, les portions du graphe correspondant aux petites tailles de buffers se superposent à charge fixée. Dans un souci de clarté, elles ne sont pas représentées sur la figure ; seule l'est la tendance qu'elles évoquent (en pointillés). Ainsi, pour de faibles tailles de buffers, cela suggère une dépendance à la charge *background* uniquement.

Nous pouvons ainsi distinguer deux zones, caractérisées par la dépendance de la performance à la taille du buffer :

- Si le *Bandwidth Delay Product* résiduel est suffisamment grand et que le buffer est petit, alors le processus décrivant la valeur de *cwnd* lorsque la perte se produit ne dépend que de ρ_b. Cette charge détermine la taille moyenne de la fenêtre de congestion, et ainsi le débit du flot.
- Pour une taille de buffer plus importante, *cwnd* peut croître jusqu'à éventuellement atteindre une valeur suffisante pour une utilisation complète de la bande passante disponible.

Vers une proposition de dimensionnement...

L'allure de $\phi(1)$ en fonction de B suggère que le buffer devrait être dimensionné afin d'éviter au moins la forte dégradation initiale du débit, due à l'interaction de TCP avec le trafic *background*

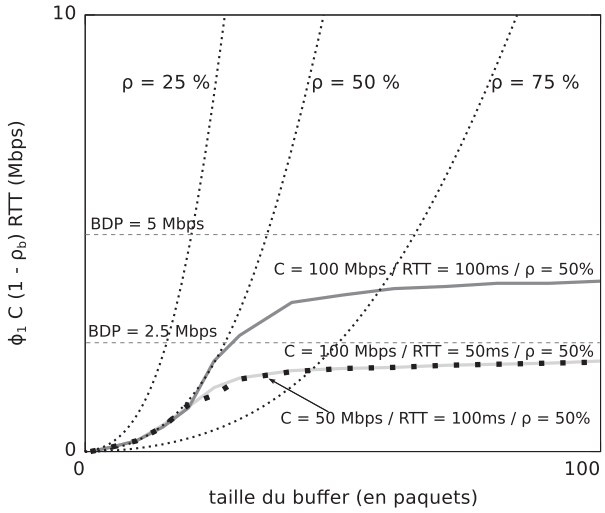

FIGURE 4.12 – Utilisation de la bande passante disponible faite par 1 flot TCP *bottlenecked*, pour différentes valeurs de la capacité C, du RTT des flots, et de la taille du buffer B (représentée en abscisse). Les courbes de *Bandwidth Delay Product* (BDP) équivalents se retrouvent superposées. Elles sont comparées à l'utilisation atteinte pour un dimensionnement suffisamment important, selon la règle empirique du BDP (lignes horizontales en pointillés). Enfin, il n'est pas possible de représenter toutes les combinaisons explorées, mais les tendances des courbes pour de petites tailles de buffer sont représentées en pointillés pour différents niveaux de charge (ρ), et suggèrent un dimensionnement en la racine carrée du BDP.

concurrent. Il est pas nécessaire cependant d'atteindre une utilisation de 100%, et une taille bien plus faible que le *Bandwidth Delay Product* semble suffire. Une possibilité est de choisir une valeur de B à la jonction des deux zones définies précédemment.

L'inspection de la courbe caractéristique des petites tailles de buffer suggère une dépendance de l'ordre de B^2. En d'autres termes, la taille requise pour B serait alors grossièrement proportionnelle à la racine carrée de la bande passante résiduelle. Cette constatation mérité d'être approfondie, mais il s'agit néanmoins d'une indication précieuse pour le dimensionnement.

Nous comprenons également pourquoi une taille de buffer intermédiaire, que nous avons évoqué plus tôt, est satisfaisante. Elle représente un bon compromis pour la performance des flots, tant que la charge *background* n'est pas trop importante (et ca sera notamment le cas sur des liens opérationnels qui sont généralement peu chargés), même pour de grandes valeurs du BDP. Par exemple, l'allure de la courbe pour une charge de 50% suggère que 100 paquets seront suffisants même si le BDP devient très élevé.

4.6.3 Trafic élastique avec des flots à débit crête limité

Notre hypothèse de flots *bottlenecked* de débit crête illimité peut être remise en cause sachant que le débit des liens d'accès n'est souvent qu'une fraction de celui des liens du cœur de réseau. Il s'ensuit un espacement naturel des paquets qui nous pousse à considérer l'impact de flots *bottlenecked* mais de débits limités [4]. Ce cas fait notamment la transition avec le régime transparent, puisque le lien ne devient saturé que lorsque plusieurs flots se combinent entre eux, ce qui se produit à forte charge. Il s'agit des cas que nous avons exclu précédemment, le débit crête définissant la charge au delà de laquelle le lien entre en régime élastique.

Afin d'illustrer l'impact de flots à débit crête limité p, nous supposons que de tels flots partagent un lien avec un trafic *background* Poisson, comme précédemment. La figure 4.13 représente le débit $\phi(i)$, en fonction du nombre de flots *bottlenecked* i, et γ/C, en fonction de la charge du lien pour différentes configurations.

Les résultats montrent que $\phi(i)$ croît linéairement tant que le débit global des flots *bottlenecked* reste relativement inférieur à la capacité résiduelle, vu que chaque flot réalise son débit crête et que le lien opère en régime transparent. Quand la charge agrégée atteint toutefois un niveau où les flots *bottlenecked* commencent à perdre des paquets, l'inefficacité des petits buffers est à nouveau visible.

4. En fait ces flots ne sont *bottlenecked* que si leur nombre est assez grand.

FIGURE 4.13 – Taux d'utilisation de la capacité résiduelle $\phi(i)$ atteint pour chaque nombre i de flots en cours ; et espérance du débit γ d'un flot en fonction de la charge ρ, pour des flots *non-bottlenecked* de débit limité à 2Mb/s

Par exemple pour $p = 2$ Mb/s dans la figure 4.13, le débit $\phi(i)$ chute lorsqu'il y a plus de 11 flots, et n'augmente à nouveau vers 100% de la capacité que lorsque ce nombre devient bien plus important.

Cependant la perte d'efficacité avec de petits buffers est dans ce cas moins significative au niveau du débit des flots comme le montre le comportement de γ en fonction de la charge du lien. La perte de débit est seulement visible pour de fortes charges où elle accentue la dégradation qui se produit dans le modèle PS idéal (vu ici lorsque B=625 paquets). Plus le débit crête des flots sera faible, plus le lien opèrera en régime transparent jusqu'à de fortes charges.

4.7 Conclusions

La relation entre la taille du buffer et la performance réalisée dépend clairement des hypothèses sur les caractéristiques du trafic. La plus significative est le mélange des débits crête exogènes des flots, c'est-à-dire les débits qu'ils atteindraient si le lien considéré était de capacité infinie. La charge du lien (taux d'arrivée des flots $\times$ taille moyenne des flots / capacité du lien) détermine alors quels flots, s'il y en a, parmi ceux de plus forts débits sont *bottlenecked*, les autres représentant pour eux une charge *background*. Nous distinguons trois principaux régimes de partage statistique de bande passante :

- lorsque tous les débits crête sont relativement peu élevés et que la charge n'est pas trop proche de 1, la somme des débits crête reste inférieure à la capacité du lien avec une forte probabilité ; nous nommons cela régime transparent ; un simple modèle de files d'attente M/M/1 peut être utilisé pour évaluer la relation entre la taille du buffer et la probabilité de perte des paquets ; un petit buffer est alors approprié ; par exemple, un buffer de 20 paquets déborde avec une probabilité de 0.01 à une charge proche de 80% ;
- lorsque quelques flots peuvent individuellement saturer la bande passante résiduelle non utilisée par la charge *background* (flots de faible débit crête), le partage de bande passante est réalisé par le contrôle de congestion de bout en bout (TCP) ; nous appelons cela le régime élastique ; avec la pratique actuelle des protocoles d'envoyer les paquets aussitôt un acquittement reçu, un petit buffer tend à déborder trop rapidement pour permettre à la fenêtre de congestion de TCP de croître complètement, ce qui peut mener à une utilisation très faible des ressources ; la taille des buffers nécessaire dans ce régime augmente avec le *Bandwidth Delay Product* résiduel ; une analyse empirique suggère que la taille du buffer soit proportionnelle à la racine carrée du *Bandwidth Delay Product* résiduel ;
- lorsque les flots de plus forts débits crête doivent se combiner (nous entendons par là plusieurs flots en parallèle) pour saturer la bande passante résiduelle, nous avons un régime intermédiaire plus général transparent/élastique ; lorsque le débit crête des flots (*potentiellement*) *bottlenecked* est une fraction relativement faible de la bande passante résiduelle (par exemple 1/10), et que la charge globale n'est pas trop proche de 1, le lien est rarement saturé et un petit buffer dimensionné comme pour le régime transparent demeure adéquat.

Il semble ainsi possible d'envisager un cœur de réseau entièrement optique, impliquant de petits buffers, si l'on peut continuer à assurer la transparence du lien par une disparité entre les débits d'accès et la capacité du lien de cœur de réseau [50]. Toutefois, dès lors qu'il sera possible pour des flots d'avoir un débit crête non négligeable par rapport à la capacité du lien, la performance se verra dégradée (plus ou moins en fonction encore du débit de ces flots). Le pire cas correspond à un flot pouvant saturer à lui seul la capacité résiduelle.

Toutefois cette étude s'intéresse uniquement à la performance de l'agrégat de flots, et ne prends pas en compte leur performance individuelle, notamment au niveau du partage réalisé. Elle ne tient pas non plus compte des évolutions possibles et probables des protocoles de transport, ni des mécanismes de qualité de service qu'il est possible de mettre en œuvre afin peut-être d'obtenir une meilleure performance. C'est ce que nous nous proposons de considérer dans le prochain chapitre.