TELECOM PARIS

Thèse de Doctorat de l'Ecole Nationale Supérieure des Télécommunications

Spécialité

Informatique et Réseaux

présentée par

Jean-Laurent COSTEUX

pour obtenir le titre de

Docteur de l'Ecole Nationale Supérieure des Télécommunications

Sujet de la Thèse

Performances des applications de transfert de données sur ATM

Soutenue publiquement le 07 octobre 1999, devant le jury composé de :

M. Eitan ALTMAN, INRIA	**Président**
M. Serge FDIDA, Université Pierre et Marie Curie	**Rapporteur**
M. Gérard HEBUTERNE, INT	**Rapporteur**
M. Patrick BROWN, CNET	**Examinateur et co-directeur**
M. Daniel KOFMAN, ENST	**Examinateur**
M. Jacques LABETOULLE, Eurécom	**Examinateur et directeur**
M. James ROBERTS, CNET	**Examinateur**

Je tiens à remercier

France Télécom-CNET qui, par l'intermédiaire de Monsieur Jérôme Chifflet, responsable du département DSE/ISE, m'a accueilli et permis de réaliser ma thèse dans les meilleures conditions,

Monsieur Jacques Labetoulle, directeur de l'Institut Eurécom, pour m'avoir fait l'honneur de diriger cette thèse,

Monsieur Patrick Brown, responsable du groupe AEP du département ISE, pour son soutien, sa confiance, ses conseils et sa précieuse collaboration tout au long de mes trois années de recherche,

Messieurs Serge Fdida et Gérard Hebuterne pour avoir accepté d'être rapporteurs, et Messieurs Eitan Altman, Daniel Kofman et James Roberts pour avoir accepté de faire partie du Jury. Ayant maintes fois pu apprécier leurs travaux de recherches et leurs enseignements, leur présence dans mon Jury est un immense honneur.

Un grand merci

A tous les membres du Groupe Algorithmique et Evaluation de Performances du CNET pour leurs conseils avisés, leurs compétences, leurs encouragements et toutes les relectures ou corrections qu'ils ont apportées à ma thèse,

A tout le personnel du CNET de Sophia Antipolis pour son accueil, sa gentillesse et sa bonne humeur,

A tous ceux qui, de près ou de loin, ont contribué à cette thèse et ont dépensé temps et énergie à la réalisation de ces travaux.

Performances des applications de transfert de données sur ATM

Jean-Laurent COSTEUX
France Télécom - CNET
Ecole Nationale Supérieure des Télécommunications

Tel : 04 92 94 53 08

E-mail : JeanLaurent.Costeux@cnet.francetelecom.fr

Institut Eurécom
Communications d'Entreprise
2229, route des Crêtes
06904 Sophia Antipolis Cedex

France Télécom - CNET
BD/CNET/DSE/ISE
905, rue Albert Einstein
06921 Sophia Antipolis Cedex

Table des matières

Liste des tableaux

Table des figures

Introduction

Aucune entreprise, aucun particulier n'admet à l'heure actuelle d'être privé de moyens de communication quels qu'ils soient (téléphone, messagerie, Internet), même pour une courte durée. Il est donc nécessaire de proposer à ces utilisateurs des réseaux fiables, performants et correctement conçus de manière à satisfaire leurs besoins à un prix acceptable. Cela constitue le principal travail des opérateurs de télécommunications.

Au début des années 1980, les réseaux RNIS-LB (Réseau Numérique à Intégration de Services - Large Bande) sont nés de la volonté de ces opérateurs d'offrir un réseau unique, multi-services, qui puisse s'adapter aux diverses exigences de leurs clients en termes de qualité de service et de variété du trafic offert. Cette notion d'"intégration de services" passe par la définition de mécanismes de contrôle de trafic et de contrôle de congestion adaptés aux réseaux à hauts débits. L'objectif principal de ces mécanismes est de trouver, d'un point de vue opérateur, le meilleur compromis entre utilisation des ressources et qualité de service offerte. L'efficacité de ce compromis détermine la capacité d'un réseau à offrir au moindre coût des services performants aux clients de l'opérateur.

Depuis la création du concept de réseaux RNIS-LB, les centres de recherche des opérateurs, tels que le CNET, centre d'études de France Télécom, ont travaillé sans relâche sur l'avènement de la technologie ATM (Asynchronous Transfer Mode) et sur les méthodes de contrôle du trafic de ces réseaux. Entre la mise à l'étude d'une technique et son exploitation, une quinzaine d'années sont généralement nécessaires et l'ATM n'a pas dérogé à la règle. En effet, c'est seulement depuis 1997 que France Télécom propose des services ATM aux entreprises et que ce mode de transfert apparaît sur toutes les lèvres et dans de nombreux journaux informatiques. Néanmoins, plusieurs interrogations subsistent, notamment en ce qui concerne la variété des services offerts et les méthodes d'acheminement des données, de la voix et de la vidéo.

Dans ce document, nous proposons d'apporter des solutions aux problèmes posés par la mise en place d'un contrôle du trafic sur ATM pour les réseaux d'entreprises, en particulier pour le transport de données. Nous nous intéressons au contrôle de trafic assumé par l'opérateur. Le but de ce travail est alors de répondre aux questions suivantes :

- Comment utiliser les ressources disponibles ou inutilisées des réseaux?
- Quelle qualité de service effective peut espérer offrir un opérateur à ses clients pour un coût donné?
- Comment protéger ou isoler les services offerts?

Pour répondre à ces questions, nous définissons l'architecture globale du contrôle de trafic dans les réseaux ATM afin de faire ressortir les choix qui sont à faire du point de

vue de l'opérateur. Dans ce but, nous organisons notre travail en trois parties.

La première partie est consacrée aux différents mécanismes de contrôle de trafic envisageables. Nous présentons les procédures et protocoles de contrôle existants, et distinguons ceux qui ont été conçus spécifiquement pour l'ATM. Cette présentation permet de comprendre les concepts généraux du contrôle de trafic et l'enjeu de ce contrôle. Nous montrons aussi pourquoi le contrôle de trafic dans l'ATM doit se différencier du contrôle de trafic existant. Nous nous efforçons tout au long de cette partie de guider l'opérateur dans ses choix des mécanismes, procédures et services les plus appropriés pour le mode de transfert ATM. Mais nous posons également le problème de l'utilisation des mécanismes de contrôle de trafic existants, tels que ceux implémentés dans le protocole TCP, au-dessus du multiplexage statistique proposé par l'ATM. En effet, de par leur utilisation mondiale, les protocoles de transfert de données actuels ne seront pas amenés à disparaître avec l'avènement des réseaux à hauts débits, et les mécanismes de contrôle de l'ATM devront composer avec ces protocoles, en particulier avec TCP. L'interrogation principale est alors la suivante : quels mécanismes, quelles procédures et quels services utiliser au niveau ATM pour assurer un acheminement correct du trafic TCP ? La première partie de notre document apporte certains éléments de réponse à cette question.

En complément et en illustration de ces éléments de réponse, nous étudions en deuxième partie les performances de la superposition de deux mécanismes de contrôle de trafic, l'un situé au-dessus d'ATM, et l'autre dans la couche ATM : nous analysons ainsi, à l'aide d'un modèle fluide, les interactions du protocole TCP et de la capacité de transfert ABR définie par l'ATM Forum, et nous montrons l'intérêt de cette superposition. Pour cela, nous étudions d'abord le comportement d'une connexion TCP sur un circuit virtuel ABR en montrant les évolutions dans le temps des différentes variables. Nous calculons ensuite les performances moyennes d'une telle connexion en termes de débit acheminé, de pertes de paquets, de délai de transit et d'occupation des mémoires tampons. Ces résultats conduisent à plusieurs interprétations concernant le dimensionnement des liens, les temps de service des commutateurs et les tailles des mémoires tampons. A la fin de cette partie, nous étendons nos résultats à l'analyse de multiples connexions TCP synchronisées sur plusieurs circuits virtuels ABR.

Notre troisième partie s'appuie sur l'étude précédente pour fournir des outils d'analyse de performances et de simulation de réseaux. Nous présentons notre programme informatique d'analyse des performances d'une connexion TCP sur ABR, appelé "PACTA", et le simulateur de réseau TCP sur ATM, appelé "STCP", que nous avons adapté aux besoins de notre étude. Ce simulateur permet dans un premier temps de valider notre analyse et de compléter les résultats obtenus. Puis, grâce à ces deux outils et à la méthode de Taguchi, nous déterminons numériquement un paramétrage adéquat du service ABR de manière à assurer un contrôle efficace et robuste du trafic des connexions TCP sur ATM dans de multiples configurations de réseaux. Nous montrons à cet effet la sensibilité des débits et des temps de réponse de TCP en fonction des paramètres spécifiés pour ABR par l'ATM Forum. L'objectif est d'établir des repères et des conseils pratiques aboutissant à la rédaction d'une notice d'utilisation de la capacité de transfert ABR prenant en considération les paramètres, les équipements et la qualité de service de TCP. Nous concluons notre thèse en dégageant les avantages de cette capacité pour le transfert de données sur ATM, dans l'optique d'un élargissement des services de l'offre proposée par France Télécom. Nous évoquons alors les recherches qui peuvent encore être menées dans cette direction.

Première partie

Le contrôle de trafic dans les réseaux ATM

Chapitre 1

Généralités sur le contrôle de trafic

Dans un réseau utilisant le mode de transfert ATM, un circuit virtuel est établi avant la transmission des informations de l'utilisateur par des cellules (ou des trames). Les circuits virtuels ouverts se partagent la capacité des liens de transmission. Différentes cellules de différents circuits virtuels peuvent donc requérir un même lien de transmission au même instant. Pour éviter les pertes, des buffers sont alors prévus dans les commutateurs et les multiplexeurs. Mais les cellules subissent dans ces buffers des délais d'attente variables et peuvent être perdues en cas de débordement. Afin que le réseau puisse répondre aux objectifs de performance fixés, des mécanismes de contrôle de trafic (ou de flux) doivent par conséquent être prévus au niveau de ces commutateurs et multiplexeurs.

Les évolutions d'ATM par rapport aux réseaux existants, notamment en ce qui concerne la largeur de la bande passante, l'intégration de services ou la simplification des fonctions internes au réseau, imposent de définir de nouveaux mécanismes de contrôle de trafic destinés à suppléer les protocoles ou les méthodes de contrôle existants. Dans ce chapitre, nous éclaircissons le concept de contrôle de trafic dans l'ATM et nous montrons en quoi les mécanismes de contrôle existants ne sont plus suffisants ou nécessitent d'être adaptés en vue d'une utilisation sur ce type de réseaux.

1.1 Définition et objectifs

L'objectif principal du contrôle de trafic est de garantir la qualité de service requise par les utilisateurs à un coût suffisamment bas pour l'opérateur. Cela sous-entend une optimisation de l'utilisation des ressources, que ce soit au niveau de la capacité des liens de transmission ou au niveau de la capacité des nœuds, par exemple les mémoires tampons des commutateurs. Cela sous-entend également un partage des ressources entre les utilisateurs afin d'exploiter au mieux la capacité du réseau.

Le contrôle de trafic, tel qu'il est défini dans [KG96], est censé réaliser le meilleur compromis entre utilisation des ressources et qualité de service. Les mécanismes de contrôle de trafic doivent permettre de satisfaire certains critères de performance, que ce soit en terme de débit acheminé, de délai, de probabilité de perte ou d'erreur. Pour cela, ils doivent

prévenir tout risque de congestion dans le réseau[1]. Malheureusement, la complexité des mécanismes de contrôle de trafic réside dans le fait qu'il est difficile de prévoir le trafic qui transite sur le réseau. De plus, en raison de la variabilité et du caractère aléatoire de ce trafic (on parle également de "sporadicité" du trafic), il est également difficile de prétendre trouver les mécanismes adéquats qui permettraient de transporter efficacement tout type de flux sur un réseau. Il s'agit parfois de sacrifier l'utilisation optimale des ressources au profit de la qualité de service dans le cas de trafic temps réel par exemple, ou bien l'inverse, dans le cas du transfert de données. Dans ce dernier cas, nous pouvons aboutir à des situations de congestion où les mécanismes de contrôle de congestion prennent le relais du contrôle de trafic afin de minimiser les conséquences de la congestion sur la qualité de service, en réduisant sa durée, son intensité et son étendue.

1.2 Contrôle de trafic existant

La classe des méthodes de contrôle de trafic la plus souvent utilisée dans les réseaux existants est celle des contrôle de flux par fenêtre de bout en bout. Ces mécanismes, dits réactifs, agissent en fonction d'informations implicites obtenues du réseau et indiquant un risque de congestion :

- Une fenêtre de congestion, symbolisée par W ou $Cwnd$, représente le nombre de paquets que peut envoyer la source de trafic sans savoir s'ils ont effectivement été reçus par la destination, c'est-à-dire sans que la source n'ait reçu d'acquittement de ces paquets. Plus cette fenêtre est grande, plus le nombre de paquets qui circulent dans le réseau est important.

- Les acquittements des paquets reviennent en fonction du délai de ces paquets, donc de la charge du réseau. Plus la charge est importante, plus le délai de retour des acquittements est grand. L'envoi des paquets dans le réseau est donc ralenti implicitement lorsque la charge est élevée. C'est pourquoi l'on parle de contrôle "réactif".

Ces mécanismes de contrôle de flux sont largement répandus en raison de leur simplicité et des multiples études dont ils ont bénéficié. Parmi les protocoles implémentant ces solutions, TCP (Transmission Control Protocol) est le protocole de transport le plus utilisé dans le monde. Nous en détaillons les principes dans le paragraphe suivant. Puis nous expliquons en quoi ces mécanismes ne sont pas adaptés aux réseaux ATM et quelles seraient les modifications qu'il faudrait leur apporter pour les intégrer dans ce type de réseaux.

1.2.1 Le protocole TCP

TCP est un protocole de transmission de données courant sur le réseau Internet. Il utilise un mécanisme de contrôle de flux fondé sur une fenêtre de congestion. Le mécanisme de

1. Une congestion est définie comme un état du réseau dans lequel ce dernier n'est plus en mesure de répondre aux objectifs de performance fixés car certains éléments du réseau sont saturés ou encombrés par un flux d'informations trop important.

contrôle de congestion TCP incrémente ou décrémente la taille de cette fenêtre en fonction des acquittements reçus ou des paquets estimés perdus.

La version TCP-Tahoe proposée dans [JAC88] et révisée par son créateur Van Jacobson dans [JAC90], a donné naissance à la version TCP-Reno qui permet une meilleure utilisation de la bande passante du lien tout en limitant le nombre de retransmissions inutiles. Suite à certaines insuffisances et dysfonctionnements de TCP-Reno, une autre version baptisée New-Reno a été proposée par Hoe dans [HOE96]. Enfin, Brakmo et Peterson ont proposé dans [BP95] la version TCP-Vegas dont le but est d'améliorer encore le débit de la connexion et de retransmettre moins de données. La version de TCP la plus répandue actuellement est TCP-Reno.

La méthode de contrôle définie par Tahoe et Reno consiste à utiliser les pertes de paquets pour prévoir la bande passante disponible sur le lien et contrôler le débit de la connexion. Cette méthode peut engendrer de grandes variations du temps de traversée aller-retour du réseau (appelé Round Trip Time ou RTT) et provoquer une saturation de ce réseau à cause d'une agressivité trop importante du protocole. Vegas tente de s'affranchir de cette méthode et propose un mécanisme plus efficace et plus performant. Malheureusement, cette version du protocole TCP ne s'est pas répandue et nous en parlerons peu dans ce document.

Nous décrivons dans cette section les mécanismes de contrôle de flux et de congestion habituellement implémentés dans TCP et suscitant un vif intérêt de la part de la communauté Réseaux et Protocoles. Nous présentons également quelques problèmes connus des différentes versions. Une description complète de l'ensemble du protocole TCP peut être trouvée dans le document de référence [STE96]. Le rapport [FCB98] en donne un résumé appréciable.

1.2.1.1 La fenêtre de congestion TCP

La fenêtre de congestion TCP est définie comme le nombre maximal autorisé de paquets (ou d'octets) transmissibles et non encore acquittés : elle représente le nombre de paquets en transit dans le réseau. En fonction de cette taille de fenêtre, le débit de transmission peut alors être augmenté ou diminué. TCP apparaît donc comme un mécanisme de contrôle de trafic par fenêtre à part entière ; sa particularité réside dans l'évolution de sa fenêtre, selon qu'un paquet est correctement reçu et acquitté, ou bien considéré comme perdu.

Les versions de TCP diffèrent dans la manière d'augmenter ou de diminuer la fenêtre et dans la façon de détecter les congestions. Plusieurs modes de croissance de la fenêtre sont définis et explicités ci-après. Un exemple d'évolution de fenêtre TCP est représenté en figure 1.1.

Le mode "slow start" : Les paquets TCP sont envoyés avec un numéro de séquence. La destination s'attend à les recevoir dans l'ordre de leurs numéros et acquitte un nombre "a" de paquet reçus par un paquet (nommé "acquittement") indiquant la séquence du prochain paquet attendu. A la réception d'un acquittement, la source TCP peut augmenter la taille de sa fenêtre. Nous remarquons qu'un acquittement confirme généralement l'arrivée de plusieurs paquets, afin de limiter la surcharge du trafic.

En mode "slow start", la fenêtre augmente d'un paquet à chaque réception d'un acquittement. Cette croissance rapide est destinée à transmettre les paquets au plus vite lorsque la taille de la fenêtre est trop petite et que l'on risque une sous-utilisation du lien. Afin de limiter dans le temps cette phase de croissance rapide, on utilise une variable, appelée "seuil de slow start" et notée W_{th} : lorsque la valeur de la fenêtre est inférieure ou égale à cette variable ($W \leq W_{th}$), la source TCP est en mode "slow start". Au-delà, elle est en mode "congestion avoidance".

Le mode "congestion avoidance" : Lorsque la taille de fenêtre W est strictement supérieure au seuil W_{th}, son mode de croissance est ralenti : sa taille augmente d'un paquet lorsqu'elle a reçu W acquittements.

Nous remarquons que, dans le mode "slow start" comme dans le mode "congestion avoidance", la fenêtre TCP croît inexorablement. Au bout d'une certaine période qui dépend de la taille des buffers, des temps de service et des capacités des liens, cette fenêtre va dépasser la capacité du réseau et un paquet sera perdu. Cette perte de paquet est inévitable dans le cas des versions Reno et Tahoe. Le principe de TCP est précisément de s'appuyer sur ces pertes pour contrôler son débit et osciller autour de la largeur de la bande passante.

Après une perte : Le protocole TCP dispose de plusieurs méthodes pour décréter qu'un paquet a été perdu. Ces méthodes sont détaillées au paragraphe 1.2.1.2. Lorsque cette perte est déclarée, la source TCP estime qu'elle a dépassé la capacité du réseau et réduit la taille de sa fenêtre.

Dans le cas de la version TCP-Tahoe, la fenêtre retombe alors à sa valeur initiale égale à 1. La source recommence une nouvelle phase de "slow start". La valeur de W_{th} est également ramenée à la moitié de la taille de la fenêtre juste avant que la perte n'ait lieu.

Le protocole TCP-Reno poursuit la même démarche que TCP-Tahoe si la perte de paquet est détectée par l'expiration d'un compteur temporel (temporisateur ou timer). Dans le cas contraire, i.e lorsqu'une perte de paquet est détectée par la réception de trois acquittements dupliqués (voir paragraphe 1.2.1.2), la source TCP exécute le mécanisme "fast recovery" en mode "congestion avoidance" :

- le seuil de slow start est ramené à la moitié de la taille de la fenêtre au moment de la détection de la perte ;
- la fenêtre de congestion prend la valeur $W = W_{th} + 3$;
- à chaque réception d'un nouvel acquittement dupliqué, la source incrémente la fenêtre d'un paquet et transmet un nouveau paquet ;
- à la réception du premier acquittement non dupliqué, W est ramenée à W_{th} et une nouvelle phase de "congestion avoidance" démarre.

Le mécanisme "fast recovery" évite de redémarrer à une valeur de W trop basse. De plus, on s'affranchit d'une phase "slow start" trop agressive. Enfin, ce mécanisme tient compte du fait que, même si l'un des paquets a été perdu, les paquets suivant celui-ci ont pu être reçus correctement. Lorsque l'on reçoit trois acquittements dupliqués, cela signifie

que trois paquets ont été reçus, mais ne correspondent pas à celui attendu. On augmente donc la fenêtre de trois pour tenir compte de ces trois réceptions. Puis, à chaque nouvel acquittement dupliqué, la destination indique à la source qu'elle a encore reçu un nouveau paquet, mais que celui-ci ne correspond toujours pas au paquet attendu. La source TCP augmente alors encore sa fenêtre de 1 pour prendre en compte ce nouveau paquet reçu.

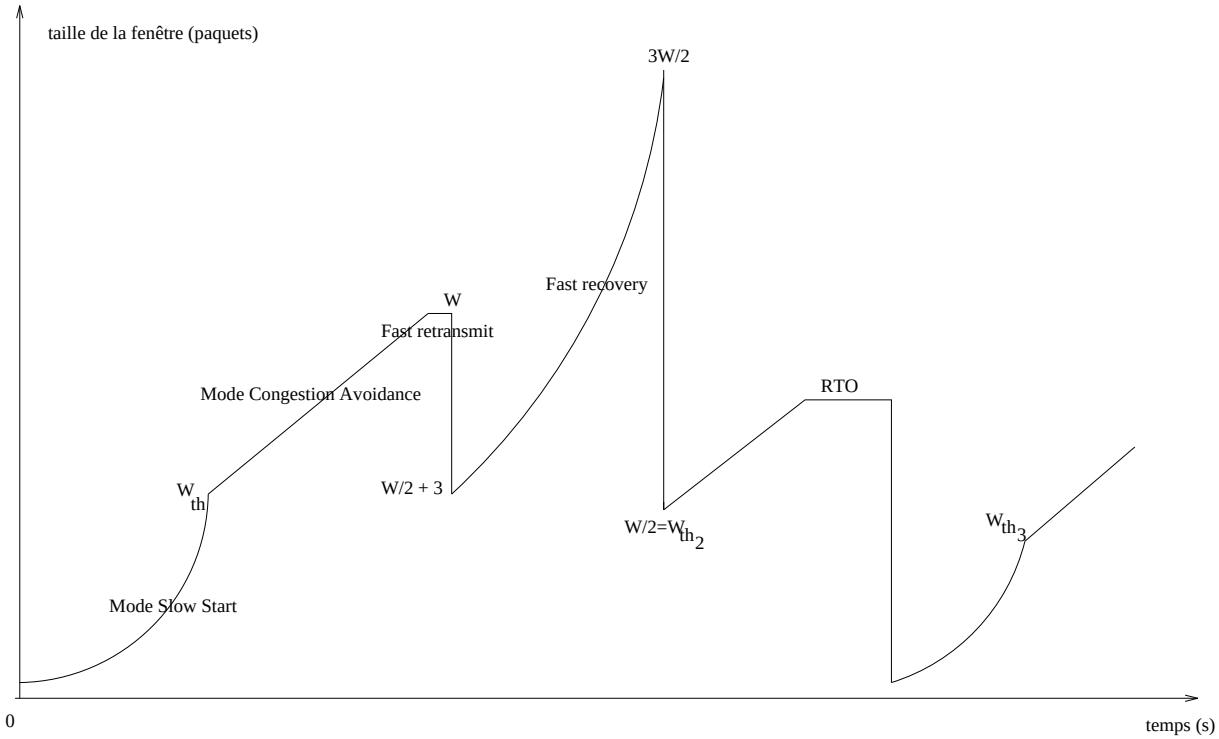


FIG. 1.1 - *Exemple d'évolution de la fenêtre TCP*

1.2.1.2 Détection d'une perte

Dans ses versions les plus courantes, le protocole TCP utilise les pertes de paquets pour diminuer la taille de la fenêtre de congestion. Ces pertes sont dues majoritairement à des débordements de buffer, à la saturation des liens et, plus généralement, à l'insuffisance des ressources qui provoquent un phénomène de congestion. A la détection d'une perte, toutes les versions de TCP proposent donc de revoir leur débit de transmission. Pourtant, comme cela est indiqué dans les documents [COS98] et [LM94], il est possible que ces pertes ne soient pas dues à des congestions dans le réseau mais, par exemple, à des reconfigurations de routeurs ou à des saturations momentanées des nœuds d'accès. Décréter une congestion pour ces exemples conduit à des réductions de débits inutiles. Cela dit, comme nous ne disposons pas d'information explicite sur la cause de la perte d'un paquet, il faut songer avant tout à limiter les conséquences d'une probable congestion. Nous nous intéressons ici aux différents moyens dont dispose la source TCP pour détecter ces pertes rapidement. Ces pertes vont délimiter ce que l'on appelle un "cycle TCP".

DÉFINITION 1.2.1
On appelle "cycle TCP" la période de temps comprise entre deux détections successives de pertes de paquets au niveau de la source TCP.

■

Le temporisateur de retransmission : La première méthode pour décréter une perte de paquet est de mesurer le temps depuis lequel le paquet a été envoyé et de vérifier si ce temps est supérieur à la valeur de la variable nommée "Retransmission Time Out" (RTO). Pour mesurer ce temps, un temporisateur appelé "temporisateur de retransmission" est déclenché par la source TCP à chaque fois qu'un paquet est envoyé. Si ce paquet n'est pas acquitté lorsque le timer expire, i.e lorsqu'il dépasse RTO, il est alors retransmis. La valeur du RTO est comprise entre 1 et 64 secondes, et sa granularité vaut généralement 500 ms. Le RTO double à chaque retransmission.

Au cours de chaque connexion, la valeur du RTO est évaluée et ajustée en tenant compte des fluctuations du trafic et du temps de transmission aller-retour (égal au RTT), c'est-à-dire de la période de temps séparant l'envoi d'un paquet de la réception de son acquittement.

L'algorithme exposé par Van Jacobson dans [JAC88] et [JAC90] pour le calcul du RTT et du RTO est le suivant :

$$\begin{aligned} Err &= M - A \\ A &\longleftarrow A + g \cdot Err \\ D &\longleftarrow D + h \cdot (|Err| - D) \\ RTO &= A + 4D \end{aligned}$$

où A est une estimation de la moyenne du RTT, M le RTT mesuré et D une estimation de l'écart type moyen. Err est la différence entre la valeur mesurée et la valeur moyenne du RTT. Le gain g pour l'estimation de la moyenne est fixé à 1/8 et le gain h à 1/4. Le calcul et la mise à jour du RTO sont renouvelés à chaque réception d'un acquittement. Néanmoins, l'algorithme de Karn spécifié dans [KP87] recommande de ne pas mettre à jour le RTO lorsque l'on renvoie un paquet ou lorsque l'on reçoit l'acquittement d'un paquet retransmis.

Acquittements dupliqués : Les problèmes de l'utilisation d'un temporisateur proviennent de l'estimation du RTT et du fait que la connexion puisse rester inactive pendant un certain temps, ne pouvant envoyer de nouveaux paquets ni retransmettre ceux qui ont été perdus. Dans un souci d'optimisation de performances de TCP, un autre principe de recouvrement de perte a été implémenté dès la version TCP-Tahoe. Il s'agit du mécanisme "fast retransmit". Le mécanisme "fast recovery" décrit auparavant a été implémenté plus tard, dans la version TCP-Reno, en complément du "fast retransmit".

Lorsqu'un paquet est reçu hors séquence, c'est-à-dire que le numéro de séquence du paquet reçu est différent du numéro attendu, la destination génère un acquittement dupliqué. La tâche des acquittements dupliqués est d'informer la source TCP que le destinataire n'a pas reçu le paquet attendu (indiqué par le numéro de séquence transporté par l'acquittement dupliqué). La source ne sait pas, à la réception de cet acquittement dupliqué, si le paquet suivant a été perdu ou bien s'il y a eu perte de l'ordonnancement des paquets. C'est pourquoi, avant de considérer qu'un paquet a été perdu, le mécanisme "fast fetransmit" recommande que la source attende d'avoir reçu trois acquittements dupliqués, soit quatre acquittements du même paquet. Elle peut alors retransmettre le paquet supposé perdu sans attendre l'expiration du temporisateur. Ceci permet d'accélérer la détection,

donc le recouvrement de perte. Lors d'une perte avec TCP-Tahoe, on recommence un nouveau cycle TCP à partir du mode "slow start" et d'une fenêtre de congestion égale à un. Dans le cas de TCP-Reno, le mécanisme "fast recovery" prend le relais afin d'éviter un gaspillage de la bande passante.

1.2.1.3 Les problèmes de TCP

Le problème majeur de TCP-Tahoe est le temps perdu avant la détection d'une perte : même si le RTO est correctement évalué, la granularité élevée du temporisateur de retransmission (500 ms) est fortement pénalisante à l'heure où les réseaux deviennent de plus en plus rapides. Aussi a-t-on introduit le mécanisme de "fast retransmit" pour détecter plus rapidement les pertes. Le mécanisme "fast recovery" propose de combler la sortie des paquets arrivés à destination (manifestée par la réception d'acquittements dupliqués) par de nouveaux paquets. Il conduit à la nouvelle version TCP-Reno.

Toutefois, les performances de TCP-Reno sont fortement dégradées dans le cas de pertes successives issues d'une même fenêtre. Ce problème est identifié par [FLO95] sous le nom de "successive fast retransmits". En effet, à chaque détection de perte par acquittements dupliqués, la source TCP-Reno réduit sa fenêtre de moitié et ne retransmet que le paquet qu'elle a pu identifier comme perdu grâce au mécanisme "fast retransmit". Si un autre paquet a été perdu, elle le découvrira lors d'un second "fast retransmit" qui conduira à une nouvelle réduction de la taille de fenêtre et du seuil de "slow start". Ceci entraîne souvent l'expiration du RTO et une nouvelle phase de "slow start" avec une valeur de W_{th} beaucoup plus basse qu'elle ne l'aurait été dans le cas de TCP-Tahoe.

Un comparatif des performances de Tahoe, Reno et New-Reno dans le cas de "successive fast retransmits" est donné dans [FF96]. Ce problème nous intéresse tout particulièrement car, dans le cas des réseaux ATM, les cas de pertes multiples sont fréquents en raison de la superposition d'autres mécanismes de contrôle de trafic, comme le lissage, qui entraînent une augmentation plus soudaine du nombre de paquets dans le réseau. Le document [MG94] montre l'inefficacité du mécanisme "fast retransmit" dans le cas de pertes multiples, et la nécessité de faire appel à un mécanisme d'acquittements sélectifs plus efficace dans un contexte haut débits, mais malheureusement peu répandu actuellement dans les versions de TCP.

Nos expériences (cf partie III) ont confirmé que le protocole TCP-Reno entraîne des conséquences catastrophiques sur ATM et qu'il est nécessaire d'implémenter une version de TCP moins pénalisante pour ce type de réseaux. Dans un souci de simplicité, nous pourrions réutiliser la version TCP-Tahoe, sans mécanisme "fast retransmit". Mais la meilleure solution serait effectivement d'utiliser l'option d'acquittements sélectifs proposée dans le RFC 1323 [JBB92] pour la version SACK-TCP. Cette option permet au récepteur d'informer l'émetteur des paquets qu'il a déjà reçus, et rend possible la retransmission de plusieurs paquets tout en conservant une taille de fenêtre plus grande, ce qui n'est pas concevable avec TCP-Reno dans le cas de pertes multiples. Le problème incontournable est qu'il est difficile, dans un premier temps, de changer la version de TCP de tous les ordinateurs de la planète. La seule solution envisageable est donc d'utiliser le mécanisme ABR qui, comme nous le montrons plus loin, peut limiter le problème des "successives fast retransmits" sur les réseaux ATM.

Le problème des "successive fast retransmits" apparaît également avec la version TCP-

Vegas, mais sous une autre forme. Il s'agit du phénomène des faux "fast recovery", qui se produit lorsque des acquittements dupliqués reviennent après l'expiration du RTO, ce qui génère des retransmissions inutiles. Néanmoins, il a été prouvé, dans [AHA96] et [AHA97], que la version TCP-Vegas permettait une utilisation maximale des liens ainsi qu'une meilleure équité que TCP-Reno. En outre, le document [AH98] montre que TCP-Vegas, s'il est correctement paramétré, interagit nettement mieux que TCP-Reno avec la capacité de transfert ABR de l'ATM Forum. Cependant, il faut relativiser cette affirmation dans le sens où, comme nous le verrons dans ce document, il est déjà délicat de paramétrer de manière adéquate le mécanisme ABR et nous pouvons nous attendre à ce que la tâche soit encore plus lourde lorsqu'il s'agit de paramétrer à la fois TCP-Vegas et ABR. Les recherches que nous avons menées nous incitent aujourd'hui à préférer la simplification et la robustesse des mécanismes plutôt que la multiplication des réglages : si ces derniers peuvent apporter un gain en performances, ils s'avèrent également difficiles à optimiser et dépendent de chaque configuration de réseau ; de plus, un mauvais paramétrage peut entraîner une baisse de performance significative, par exemple une mauvaise estimation du RTT dans le cas de TCP-Vegas. C'est sans doute pour ces raisons que la communauté Internet n'a pas validé le protocole TCP-Vegas qui apparaît plus compliqué que ses prédécesseurs.

Outre les problèmes des mécanismes "fast retransmit" et "fast recovery" mentionnés précédemment et dont les conséquences varient en fonction des versions, quelques problèmes semblent inhérents aux multiples versions développées du protocole TCP. En effet, Reno et Vegas n'apportent pas ou peu de solutions à certains problèmes de TCP-Tahoe remarqués notamment dans [SZC90].

1.2.2 Le mixage des paquets

Dans le cas d'un réseau mettant en concurrence plusieurs connexions TCP, nous constatons que les paquets de chaque connexion ne sont pas "brassés" ou mixés ensemble. En effet, lorsque la fenêtre TCP est augmentée, le paquet supplémentaire est envoyé juste après le paquet précédent. Lorsque ces paquets arrivent au commutateur, ils sont adjacents dans la file. Ces paquets adjacents vont générer, par l'intermédiaire de leurs acquittements, deux paires de paquets à nouveau adjacents. Il n'y aura donc pas de paquet provenant d'autres connexions entre les paquets d'une même connexion. Ce phénomène nuit à l'équité entre les connexions puisque les ressources sont temporairement monopolisées par les paquets adjacents d'une même connexion.

Les solutions à ce problème ne se trouvent pas dans les versions Reno et Vegas de TCP. On peut y remédier en introduisant par exemple un délai (RTT/W) entre 2 paquets d'une même connexion. Mais la méthode la plus intéressante et la plus performante est d'utiliser des procédures de gestion des nœuds du type Fair Queueing ou Round Robin pour mixer les paquets des différentes connexions dans la file d'attente du commutateur. Ce type de procédure sera détaillé au paragraphe 2.2.3 et s'applique bien dans le cas des réseaux ATM, destinés à l'intégration de services.

1.2.3 La synchronisation des sources

Lorsque de multiples connexions partagent le même chemin, ce chemin devient congestionné au même moment pour toutes les connexions. Plusieurs de ces connexions sont ainsi amenées à perdre un paquet. Leurs mécanismes de contrôle réagissent alors simultanément à la congestion : les sources se synchronisent sur la même période de congestion et font décroître la taille de leurs fenêtres de congestion au même instant. Il en résulte une mauvaise utilisation du lien, puisque toutes les sources réduisent leur débit en même temps.

Une solution partielle peut consister à utiliser les versions Reno et Vegas car la charge totale du réseau ne chute pas aussi terriblement. Mais la meilleure solution, dans le cas de sources synchronisées, est de jeter les paquets de manière préventive, avant qu'un débordement de buffer ne se produise. Cela élimine les pertes de paquets synchronisées, car seule une connexion voit un de ses paquets jeté pendant la période de congestion. C'est la solution mise en œuvre par la procédure "Random Early Detection" (cf [FJ93]) détaillée au paragraphe 4.3.2.1.

1.2.4 Limitations des mécanismes de contrôle par fenêtre dans le cadre des réseaux à hauts débits

Comme nous venons de le voir, les mécanismes de contrôle de flux par fenêtre tels que TCP présentent l'immense avantage d'être relativement simples. Du fait de leur utilisation sur les réseaux existants durant de longues années, les ingénieursont acquis une expérience significative de ces mécanismes. Malheureusement, l'utilisation de ces mécanismes pose quelques difficultés dans le cadre des réseaux hauts débits à intégration de service.

Nous avons vu certains problèmes du protocole TCP. Voyons à présent les problèmes communs à l'ensemble des mécanismes de contrôle par fenêtre dans le cadre d'une utilisation sur les réseaux ATM.

1.2.4.1 La largeur de la bande passante

Les réseaux à hauts débits offrent une largeur de bande passante plus grande que celle des réseaux "normaux" : les fibres optiques permettent de propager davantage de cellules plus rapidement, les routeurs et commutateurs ont des temps de service plus petits et des tailles de buffers plus grandes. Il en résulte que le nombre de cellules, donc de paquets, en cours de transfert dans un réseau à haut débit peut être largement supérieur à celui d'un réseau de paquets "standard" ; ce phénomène est particulièrement remarquable sur les réseaux longue distance où les délais d'acheminement des paquets sont importants. Pour que les sources soient persistantes (i.e qu'elles émettent en continu), il faut donc que leurs tailles de fenêtre soient très grandes. Mais, de grandes fenêtres équivalent à des temps de réponse ou de réaction du mécanisme de contrôle plus grands. Ce problème a été maintes fois constaté pour TCP : si la fenêtre peut envoyer 100 paquets sans recevoir d'acquittement, il va de soi que le moment où la source s'aperçoit de la congestion, c'est-à-dire le moment où elle constate que ses acquittements ont été retardés, arrive souvent trop tard.

1.2.4.2 La qualité de service

Les réseaux ATM ne sont pas simplement des réseaux à hauts débits. Ce sont également des réseaux à intégration de services. Ils doivent donc être capable de transporter tout type de flux pouvant provenir d'applications de transfert de données autant que d'applications temps réel telles que la voix ou la vidéo. Or la qualité de service requise par ces différentes applications n'est pas la même.

Délai : Le contrôle de trafic par fenêtre a été développé pour des réseaux de données, sans grande contrainte de délai ni garantie de bande passante, mais permettant un taux de perte faible. Ce genre de mécanisme ne s'applique pas ou peu dans le cas par exemple d'applications temps réel à débit fixe : il n'est pas concevable d'imaginer le débit d'une visio-conférence varier en fonction de la taille de la fenêtre de congestion, ni les cellules d'un flux temps réel rester bloquées dans le buffer d'un commutateur à cause de fenêtres trop grandes. La seule alternative que proposent les mécanismes de contrôle par fenêtre est de choisir entre leur utilisation, limitant les pertes au prix d'un délai variable, ou leur non-utilisation, au prix d'un contrôle aléatoire des pertes dans le réseau. Nous verrons par la suite comment résoudre ce problème dans l'ATM. La principale constatation dans le cas présent est qu'il est difficile d'offrir des qualités de service spécifiques lorsque le seul mécanisme de contrôle de trafic utilisé est un contrôle par fenêtre.

Pertes : Le deuxième principe des mécanismes de contrôle de flux par fenêtre est de s'appuyer sur les pertes de paquets pour réguler leur débit [2]. En fin de compte, ces mécanismes n'évitent pas les phases de congestion ni les pertes de paquets. Ils ont été conçus pour pouvoir les traiter ; c'est pour cela que l'on parle de contrôle réactif. Ce phénomène est particulièrement pénalisant dans le cas de l'ATM, qui s'appuie précisément sur des réseaux fiables, rencontrant peu de pertes, pour acheminer son trafic. La présence inévitable et fréquente de pertes sur un réseau peut être une contrainte inacceptable pour certaines applications.

Au vu des défaillances des mécanismes de contrôle par fenêtre pour une utilisation dans le cadre de l'ATM, il semble nécessaire de mettre en place, non pas un seul mécanisme de contrôle de trafic, mais un ensemble cohérent de mécanismes spécifiques aux réseaux ATM. Cet ensemble de mécanismes est détaillé dans cette partie. L'objectif de la seconde partie est d'évaluer les performances des mécanismes existants, décrits plus haut, lorsqu'ils interagissent avec les mécanismes développés pour l'ATM.

1.2.5 Les mécanismes de contrôle par crédit

Avant d'aborder la description de l'ensemble des mécanismes de contrôle de trafic déployés dans l'ATM, il est nécessaire de modérer nos propos quant aux nouveaux problèmes posés par les réseaux ATM vis-à-vis des mécanismes de contrôle existants. En effet, des solutions à ces problèmes peuvent être trouvées et il est possible, pour certains types de

2. Le protocole TCP-Vegas représente une exception à cette règle

services, d'intégrer des mécanismes de contrôle par fenêtre dans le cadre du contrôle de trafic des réseaux ATM.

Les mécanismes existants décrits précédemment implémentent un contrôle de flux de bout en bout. Pour des applications de transfert de données dans l'ATM, i.e des applications qui ne requièrent pas de bande passante garantie ni de contrainte temps réel, le problème majeur d'un contrôle de flux de bout en bout provient des délais rencontrés dans les nœuds, des temps de propagation sur les liens, et des grandes tailles de fenêtre : le délai pour obtenir la réponse du réseau est trop important. Une des solutions proposées alors pour adapter les mécanismes de contrôle par fenêtre au transfert de données dans l'ATM est d'offrir un contrôle lien par lien, à la place d'un contrôle de bout en bout.

Le principe est identique aux mécanismes de contrôle par fenêtre de bout en bout, appliqué entre deux nœuds consécutifs d'une même connexion. Le nœud i de la connexion envoie un crédit vers le nœud *i-1* quand il a pu libérer une place dans la mémoire de la connexion après l'envoi du paquet vers le nœud *i+1*. Si le lien de sortie du nœud i est saturé, il retarde l'envoi des crédits vers le nœud *i-1* qui, voyant les paquets s'accumuler dans son buffer, retarde à son tour l'envoi des crédits vers le nœud *i-2*. Le contrôle se propage alors jusqu'à la source par un phénomène dit de "backpressure". Ce mécanisme, appliqué à l'ATM, est décrit dans [KBC94].

Ce mécanisme de contrôle par crédits lien par lien a été proposé à l'ATM Forum pour la classe de service à bande passante non garantie et trafic non temps réel. Il a été démontré que ce mécanisme garantissait un taux de perte très faible, une optimisation des ressources et une équité meilleures que les mécanismes de contrôle par débit que nous décrivons plus bas. On lui a reproché d'une part de générer trop de trafic pour le contrôle des crédits et d'autre part d'être trop compliqué à implémenter, car le commutateur doit connaître le nombre de cellules de chaque connexion présentes dans ses buffers. De plus, les surplus de paquets sont stockés dans les nœuds du réseau et non plus aux extrémités : toutes les connexions sont pénalisées et il est nécessaire de prévoir davantage de place mémoire dans chacun des buffers du réseau. Ces inconvénients ont bien sûr contribué à la dévalorisation de ces mécanismes mais la principale raison pour laquelle l'approche par crédits n'a pas été retenue est économique : les mécanismes de contrôle par crédits laissent peu de libertés aux constructeurs, à l'opposé des mécanismes de contrôle par débit, décrits à la section 3.2.

Nous avons décrit dans ce premier chapitre certaines solutions classiques proposées pour contrôler le trafic de données sur les réseaux actuels. Comme nous l'avons remarqué, ces solutions ne sont pas toujours adaptées aux réseaux ATM. Nous allons voir dans le chapitre suivant quelles solutions s'offrent à nous pour contrôler efficacement le trafic avec ce mode de transfert.

Chapitre 2

Le contrat de trafic

Le contrôle de trafic dans les réseaux ATM repose sur l'établissement d'un contrat de trafic entre l'utilisateur et le réseau lors de l'ouverture d'une connexion. L'utilisateur fournit à l'opérateur les paramètres qui caractérisent le trafic qu'il présentera à l'entrée du réseau. L'utilisateur requiert en même temps une certaine qualité de service. Si le réseau accepte la connexion, cette qualité de service devra être respectée tant que durera la connexion, sous réserve que la source respecte de son côté les paramètres de trafic qu'elle a spécifiés dans le contrat. Notons que la source peut demander la renégociation du contrat de trafic en cours de connexion et qu'il appartient au réseau, d'une part de décider s'il dispose des ressources nécessaires pour accepter la connexion (rôle de la fonction CAC, cf paragraphe2.2.1), d'autre part de vérifier que la source respecte bien les paramètres du trafic auxquels elle s'est engagée (rôle de la fonction UPC, cf paragraphe 2.2.2).

2.1 Paramétrage et qualité de service

Afin de déterminer si une connexion respecte le contrat négocié avec le réseau ou si le réseau est capable d'admettre une nouvelle connexion, un certain nombre de paramètres ont été définis ; chaque paramètre spécifie un aspect particulier du trafic, tel que son débit crête, son débit moyen ou la taille maximale des rafales. L'ensemble des paramètres qui peuvent être utilisés par une source pour caractériser son trafic lors de la négociation du contrat est appelé descripteur de trafic de la source. Dans le cas des réseaux ATM, les paramètres de trafic négociables, définis par l'ATM Forum dans [ATM99], sont les suivants :

- Peak Cell Rate (PCR) : débit crête de la source.
- Minimum Cell Rate (MCR) : débit minimal de la source.
- Sustainable Cell Rate (SCR) : débit soutenu par la source. Il représente la borne supérieure du débit moyen de la connexion.
- Maximum Burst Size (MBS) : durée maximale des rafales, soit le nombre maximal de cellules que la source peut transmettre au débit crête.
- Maximum Frame Size (MFS) : introduit dans le mécanisme GFR (Guaranteed Frame Rate, cf paragraphe 3.3). C'est le nombre maximal de cellules contenues dans une

trame lorsque les cellules d'informations sont organisées en trames délimitées au niveau de la couche ATM.

- Cell Delay Variation Tolerance (CDVT) : lors du multiplexage des cellules d'une connexion avec celles des autres connexions, une variation de délai peut être générée entre les cellules, même si celles-ci ont été émises à un débit conforme au contrat. De plus, le passage des cellules sur la couche physique doit respecter le train de slots imposé par le lien. Le débit d'émission au niveau de la couche ATM n'est pas nécessairement en phase avec le débit du lien. Il s'ensuit une variation du délai entre les cellules d'une même connexion. Afin de ne pas pénaliser la source si celle-ci émet ses cellules correctement, on tient compte d'une possible variation du délai entre les cellules. CDVT représente la variation maximale que le réseau peut tolérer.

Une fois le trafic de la source caractérisé, il est possible de spécifier la qualité de service (Quality of Service, QoS) garantie tout au long de la connexion. Une partie des paramètres de qualité de service est négociable : l'utilisateur peut demander à bénéficier de cette qualité de service en échange du trafic qu'il présente à l'entrée du réseau. Ces paramètres sont les suivants :

- Maximum Cell Transfer Delay (max CTD) : le temps de propagation maximum entre la source et la destination, à partir des Interfaces Utilisateurs-Réseaux (UNI).
- Peak to Peak Cell Delay Variation (peak-to-peak CDV) : la variation maximale du temps de propagation des cellules. Le terme "peak-to-peak" se réfère au meilleur et au pire des temps de propagation.
- Cell Loss Ratio (CLR) : le taux de perte de cellules, défini comme le nombre de cellules perdues divisé par le nombre total de cellules émises.

Les autres paramètres de qualité de service seront calculés par le réseau en fonction du trafic de la source. Le réseau s'engage à les respecter, mais ces paramètres ne sont pas négociables :

- Cell Error Ratio (CER) : le taux de cellules erronées, soit le nombre de cellules erronées divisé par la somme du nombre de cellules correctement transmises et du nombre de cellules erronées.
- Severely-Errored Cell Block Ratio (SECBR) : le taux de blocs de cellules erronés, soit le nombre de blocs erronés sur le nombre de blocs transmis. Un bloc représente le nombre de cellules de données transmises entre deux cellules OAM (cellules d'opération et de maintenance).
- Cell Misinsertion Rate (CMR) : le débit de cellules insérées par erreur. Ce débit représente le nombre de cellules insérées par erreur durant un intervalle de temps (cet intervalle peut être la durée de la connexion), divisé par cet intervalle de temps.

2.2 Les fonctions de contrôle de trafic

Après avoir défini en quels termes était négocié, pour chaque connexion (Virtual Path Connection et Virtual Channel Connection), le contrat de trafic entre utilisateur et réseau,

il importe à présent de préciser quelles fonctions sont nécessaires pour contrôler ce trafic. En premier lieu, accompagnant la négociation du contrat de trafic, une fonction de contrôle d'admission de connexion (Connection Admission Control, CAC) est indispensable : son rôle sera de décider si le réseau dispose ou non des ressources nécessaires pour admettre la nouvelle connexion tout en garantissant la QoS demandée par cette connexion et par les connexions précédemment établies. Ensuite, afin de vérifier le respect des engagements du contrat de trafic, la fonction UPC (Usage Parameter Control) est chargée de détecter toute violation du contrat et de prendre les mesures appropriées. Hormis ces deux fonctions CAC et UPC, indissociables du contrat de trafic, d'autres mécanismes peuvent venir compléter le contrôle de trafic dans les réseaux ATM.

2.2.1 La fonction CAC

La fonction CAC est définie comme l'ensemble des actions prises par le réseau lors d'une demande d'établissement d'une connexion, afin de déterminer si le réseau doit accepter ou rejeter l'appel. Cette fonction peut également être sollicitée lorsqu'une source souhaite renégocier les paramètres de son trafic et la qualité de service requise. La fonction CAC établit les paramètres qui doivent être contrôlés par l'UPC.

Si le réseau fait simplement du multiplexage déterministe, la fonction CAC est simple : on alloue à chaque connexion son débit crête, dans la limite de la capacité disponible sur le lien. Mais, compte tenu de la grande hétérogénéité et de la sporadicité des trafics présentés à l'entrée d'un réseau, il paraît probable que le réseau utilise un multiplexage statistique, afin d'optimiser les ressources. Dans ce cas, la fonction CAC est plus complexe. Une approche possible est de décider si le réseau dispose de ressources suffisantes dans le pire des cas (cf [DOS93]). Cela revient bien souvent à faire du multiplexage déterministe, c'est-à-dire que l'on privilégie la qualité de service au détriment d'une utilisation optimale des ressources. Une autre approche, dans le cas de trafics variables, est présentée dans [ROB91] et consiste à déterminer la bande passante effective qu'il est nécessaire d'allouer à chaque connexion pour garantir une probabilité de perte faible tout en optimisant l'utilisation des ressources. Néanmoins, si le pire cas se présente, le réseau ne pourra pas garantir la qualité de service pour laquelle il s'est engagé. De plus, la bande passante effective est difficile à définir, donc difficile à contrôler au niveau de l'UPC.

Aucune spécification n'est indiquée par l'ATM Forum quant à la manière d'implémenter la fonction CAC. La conception de la fonction CAC reste donc à la discrétion de l'opérateur.

2.2.2 La fonction UPC

La fonction UPC (Usage Parameter Control), ou contrôle de conformité, définit les actions prises par le réseau pour gérer et contrôler le trafic aux accès du réseau. L'objectif est de détecter toute violation du contrat de trafic établi et de prendre les mesures appropriées pour protéger le réseau des applications malicieuses ou malveillantes, aussi bien que des erreurs non intentionnelles qui peuvent affecter la qualité de service des connexions établies. A la réception des cellules, l'UPC détermine, avant de les envoyer sur le réseau, si ces cellules correspondent à une connexion valide et si elles respectent le contrat de trafic négocié pour cette connexion. Dans le cas contraire, elle décide des mesures à prendre qui

peuvent consister par exemple à marquer ou à détruire les cellules non conformes.

2.2.2.1 La conformité

Un algorithme GCRA (Generic Cell Rate Algorithm) est désigné pour déterminer la conformité des cellules. Deux versions équivalentes sont proposées : le Virtual Scheduling Algorithm (VSA), représenté figure 2.1 et le Continuous State Leaky Bucket Algorithm (CS-LBA), représenté figure 2.2. Ces deux algorithmes reposent sur :

- la suite des instants d'arrivée effectifs des cellules à l'UPC ;
- la suite des instants d'arrivée théoriques des cellules, qui dépendent du débit auquel ces cellules doivent être envoyées (ce débit a été négocié dans le contrat de trafic) et de la tolérance de variation de délai.

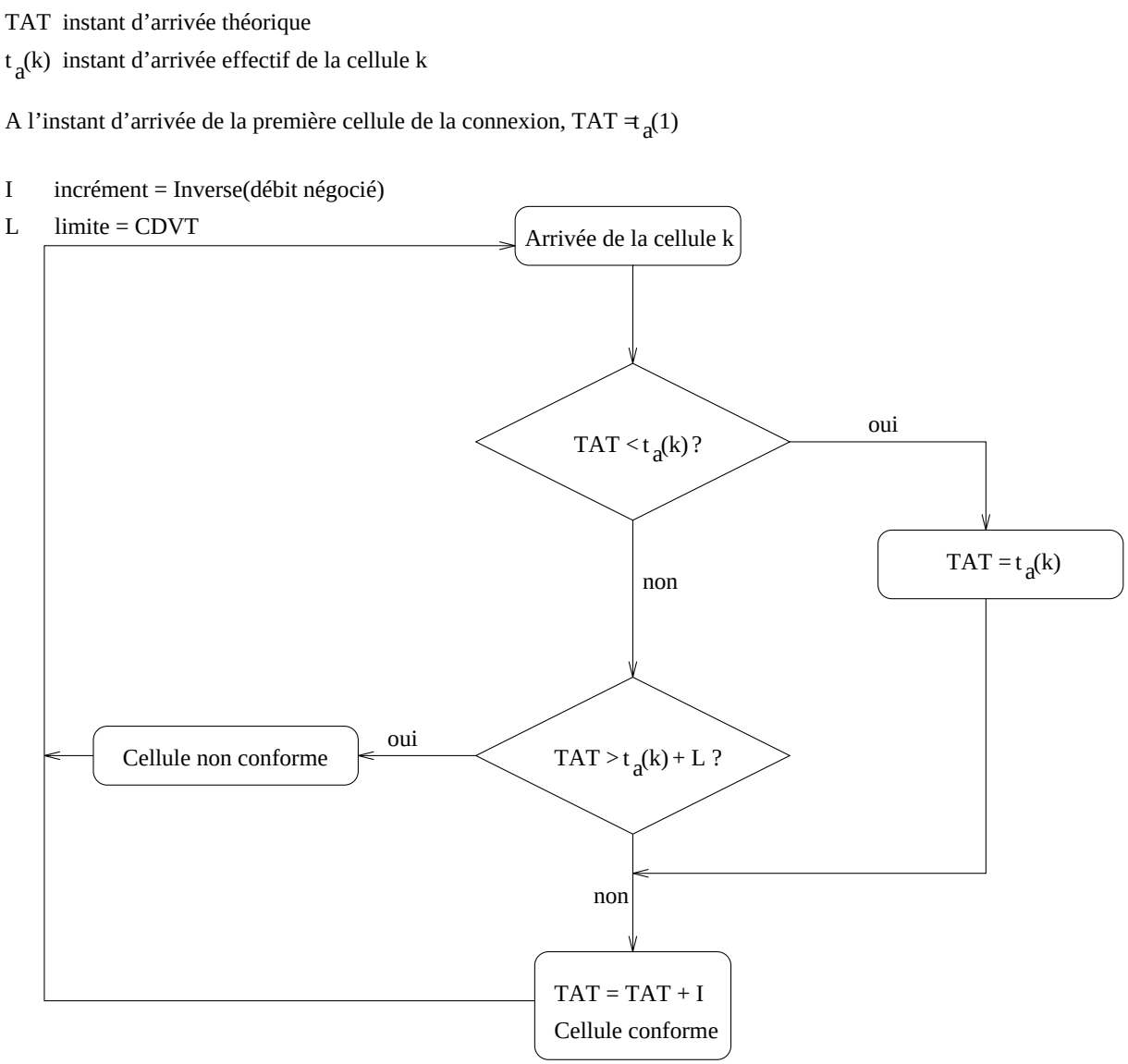


FIG. 2.1 - *Virtual Scheduling Algorithm*

LCT dernier instant de conformité

$t_a(k)$ instant d'arrivée effectif de la cellule k

X valeur du compteur du leaky bucket

X' variable auxiliaire

A l'instant d'arrivée de la première cellule de la connexion, X = 0 et LCT = $t_a(1)$

I incrément = Inverse(débit négocié)

L limite = CDVT

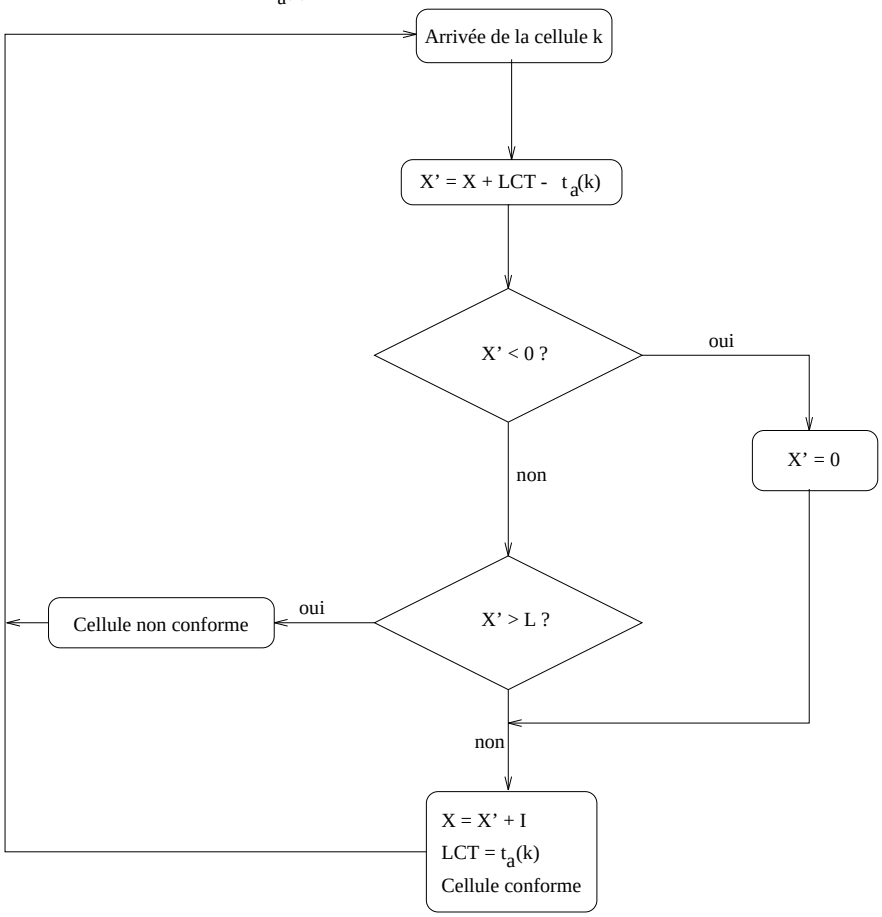


FIG. 2.2 - *Continuous State Leaky Bucket Algorithm*

ABR et la conformité : Dans le cas de la capacité de transfert ABR décrite à la section 3.2, une connexion est caractérisée par les débits MCR, PCR et ACR. Le débit ACR (Allowed Cell Rate) s'adapte dynamiquement à la capacité disponible pour la connexion. Aussi la conformité de l'ABR diffère-t-elle du GCRA classique, de par la variabilité de l'incrément I ($I = \frac{1}{ACR}$). Cet incrément varie en fonction des indications provenant du réseau et transportées par les cellules de gestion des ressources. Ces cellules RM (Ressource Management) informent d'abord l'UPC du nouveau débit alloué à la connexion, puis atteignent la source ABR. Il y a donc un problème de synchronisation entre la source et l'UPC, dans le sens où il ne faudra appliquer un changement de débit à l'UPC que lorsque la source aura pris connaissance du nouveau débit alloué et que les cellules censées respecter ce débit seront arrivées à l'interface. Ce temps correspond à un temps d'aller-retour entre l'UPC et la source (cf figure 2.3). Rappelons à ce sujet que l'UPC se trouve à l'entrée du réseau, aux interfaces UNI (User to Network Interface) ou INI (International Network Interface), tandis que la source du trafic se trouve au niveau de la couche ATM du terminal équivalent ; entre l'UPC et la source se trouvent les fonctions de la couche physique ainsi que les équipements du client tels que les multiplexeurs.

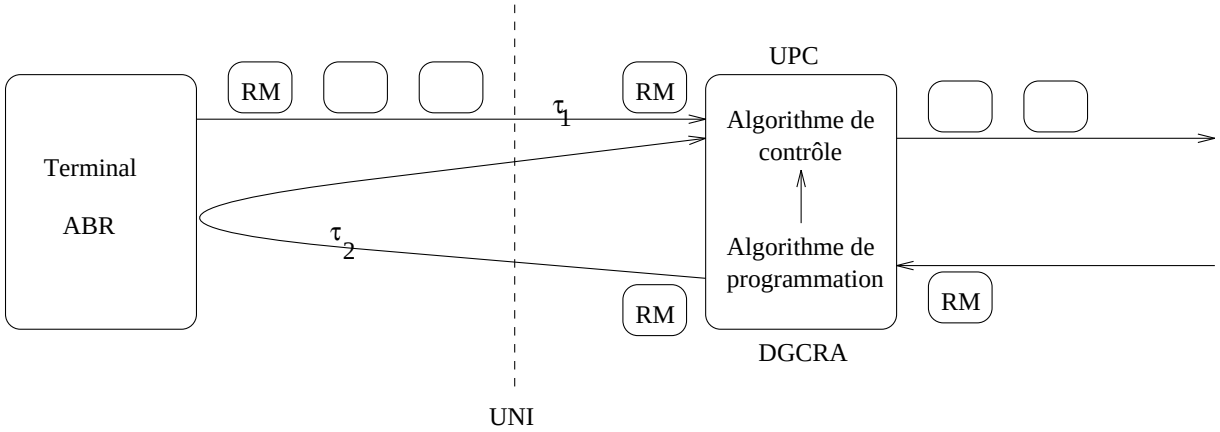


FIG. 2.3 - *Délai entre l'UPC et la source*

Entre chaque changement de débit, la connexion est autorisée à émettre au débit ACR fixe sur cette période. Par conséquent, l'algorithme GCRA classique peut s'appliquer sur l'intervalle de temps où le débit autorisé est constant. Cependant, du fait de la variabilité du débit à contrôler, le GCRA doit être étendu afin de déterminer les valeurs et les instants d'application des débits à contrôler. Le contrôle de conformité de l'ABR fait donc intervenir un nouvel algorithme dynamique appelé "Dynamic GCRA" (DGCRA). Le DGCRA est composé de deux processus :

- un algorithme de contrôle, similaire au GCRA, qui contrôle le débit auquel la source est autorisé à transmettre ;
- un algorithme de programmation qui détermine à quel moment prendre en compte, dans le contrôle de la conformité des cellules, la demande de changement de débit portée par les cellules RM. La programmation des changements de débit s'effectue sur la base des pires cas : les augmentations de débit sont programmées le plus tôt possible (en fonction de la variation de délai tolérée entre la source ABR et l'UPC) et les diminutions le plus tard possible.

De nombreux algorithmes DGCRA ont été proposés. Les documents [BBFa] et [BBFb] ont été les premiers à lever la problématique de la conformité ABR et à poser les fondements de sa définition en proposant un algorithme qui a été appelé par la suite Algorithme A. Un algorithme plus simple, appelé Algorithme B, a été présenté dans [ATM96] puis amélioré dans [RAB97] et [CC96]. Cet algorithme a été retenu à l'UIT comme spécification de la conformité ABR. Enfin, le CNET propose dans [RK98] un algorithme simple à implémenter et permettant un contrôle plus strict que l'algorithme de l'UIT. Notons que ces algorithmes s'appliquent uniquement au mode d'indication explicite du débit (Explicit Rate, ER).

2.2.2.2 Mécanismes de contrôle d'accès

A partir des algorithmes de contrôle de conformité définis dans la section précédente, il est possible de réaliser divers mécanismes de contrôle d'accès. On distingue deux classes de mécanismes de contrôle : les mécanismes par prélèvement (ou par crédits) et les "shapers", tels que le contrôleur-espaceur, qui remettent en forme le trafic. Les actions que peuvent réaliser ces mécanismes sont les suivantes :

- laisser passer la cellule ;
- détruire la cellule ;
- retarder la cellule ;
- marquer la cellule, i.e positionner le bit CLP (Cell Loss Priority) de la cellule ATM à 1 ;
- avertir la fonction CAC afin qu'elle ferme la connexion.

Les mécanismes par prélèvement : Un mécanisme par prélèvement est transparent pour le trafic conforme et repose sur les principes généraux suivants :

- il contient un ensemble de N crédits ;
- quand une cellule arrive dans le mécanisme, elle prend un crédit s'il en reste, sinon elle est déclarée comme non conforme ;
- les crédits sont engendrés suivant une règle propre au mécanisme.

Les mécanismes par prélèvement les plus référencés dans la littérature sont l'algorithme de fenêtre sautante, l'algorithme de fenêtre glissante, l'algorithme EWMA (Exponentially Weighted Moving Average) et le Leaky Bucket. Une description des propriétés de ces mécanismes peut être trouvée dans [RAT90] et [GUI99]. Les mécanismes par prélèvement diffèrent des mécanismes de fenêtre utilisés dans les réseaux existants par leur manière de prévenir les congestions au lieu d'y réagir : la génération des crédits dépend du comportement de la source et non du comportement du réseau.

Notons qu'un mécanisme par prélèvement doit observer au moins N (le nombre de crédits) cellules du flux contrôlé avant de pouvoir détecter une cellule en violation. Dès lors, plus N est grand, plus le mécanisme est inerte. Ceci conduit naturellement à prendre le nombre de

crédits comme critère de sélection : pour une marge de sécurité et une précision données, moins le mécanisme contient de crédits, plus il est efficace. Avec ce critère, il ressort de [BW95] que le Leaky Bucket est le plus efficace des mécanismes par prélèvements cités ci-dessus. C'est pourquoi nous nous y intéressons plus longuement dans le paragraphe suivant.

Le Leaky Bucket : Le Leaky Bucket a été introduit par Turner dans [TUR86] et a suscité de nombreuses études et améliorations. Dans ce mécanisme par prélèvement, les crédits sont générés périodiquement, c'est-à-dire qu'un crédit est généré toutes les T unités de temps où $1/T$ est le débit policé, appelé ici débit de fuite. Le Leaky Bucket implémente l'algorithme de conformité CSLBA décrit plus haut.

Une amélioration du Leaky Bucket, le Buffered Leaky Bucket (cf [SLCG93]), allie les fonctionnalités des mécanismes par prélèvement et des shapers. Ce mécanisme consiste en un ensemble de N crédits générés périodiquement avec une période T, et également d'une mémoire tampon pour les cellules. Si une cellule arrive alors qu'aucun crédit n'est disponible, elle est marquée puis rangée dans la mémoire tampon et ne la quitte que quand un crédit a été créé. Sinon, les cellules passent de manière transparente. Cette solution permet de lisser le trafic. Le schéma classique de ce leaky bucket est représenté figure 2.4.

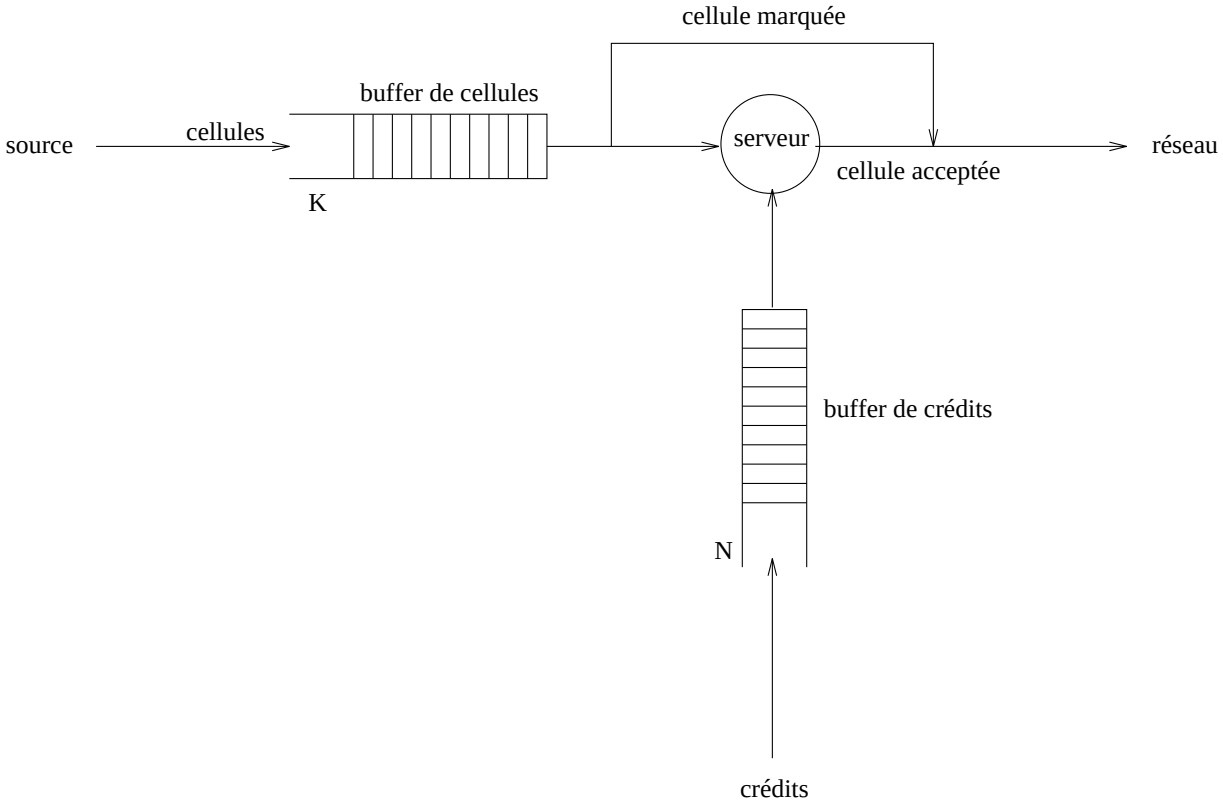


FIG. 2.4 - *Buffered Leaky Bucket*

Le document [GUI99] montre que le Leaky Bucket peut autoriser par instants des trafics abusifs ("trafics pires") sur le réseau, ce qui peut être pénalisant du point de vue des ressources et de la qualité de service offerte. Ce défaut ne peut être compensé que par l'introduction de larges buffers, ce qui a pour effet d'augmenter exagérément le délai des

cellules. C'est pour cela que le CNET a introduit dans [BRS90] la notion de contrôleur-espaceur.

Le Contrôleur-Espaceur : Outre le Buffered Leaky Bucket, le Contrôleur-Espaceur est sans doute le meilleur représentant des mécanismes à mémoires tampons. Son objectif principal est de remettre en forme le trafic avant de l'envoyer vers le réseau. Pour cela, le contrôleur-espaceur dispose de plusieurs mémoires tampons, chacune étant associée à une connexion et vidée suivant les paramètres de trafic déclarés dans le contrat. Dès lors, si une violation de contrat se produit pour la connexion, la mémoire tampon associée déborde et des cellules sont détruites.

La fonction principale du contrôleur-espaceur est de lisser, connexion par connexion, les cellules. Il implémente directement l'algorithme VSA décrit plus haut pour calculer les instants théoriques d'arrivée des cellules (compte tenu du débit négocié dans le contrat) et les comparer aux instants d'arrivée effectifs. L'idée de base est de laisser passer les cellules en retard par rapport à leur instant d'arrivée théorique et de retarder les cellules conformes mais en avance jusqu'à leur instant théorique : une date de ré-émission des cellules est donc calculée de telle sorte que la distance entre la cellule considérée et la cellule précédente soit au moins égale à la période négociée dans le contrat de trafic. Une description complète de ce mécanisme peut être trouvée dans [BSG92]. L'opérateur France Télécom déploie actuellement des contrôleurs-espaceurs sur son réseau dorsal afin de diminuer les coûts des communications et d'augmenter la souplesse des trafics offerts ; les caractéristiques des services associés à ces contrôleurs sont décrits dans le document [Fra98].

2.2.3 Autres mécanismes de contrôle de trafic

Nous venons de présenter les mécanismes de contrôle de trafic désignés pour assurer le contrôle d'admission et le contrôle de conformité des réseaux ATM. D'autres mécanismes de contrôle de trafic peuvent compléter les fonctions précédemment décrites et être implémentés à divers endroits du réseau. C'est le cas notamment des fonctions de lissage et de remise en forme du trafic (shaping). Nous avons présenté le lissage comme associé à l'UPC. Il peut aussi être réalisé dans d'autres éléments du réseau lorsque l'on souhaite reformer le trafic conformément au contrat.

Il importe cependant de distinguer les actions réalisées par l'UPC du shaping. La fonction UPC s'est peu à peu concentrée sur la fonction de contrôle. De son côté, la notion de "traffic shaping" se charge uniquement de la remise en forme du trafic et est actuellement dominée par le concept de serveur cyclique généralisé (discipline WFQ, Weighted Fair Queueing). La fonction d'espacement remplie par le contrôleur-espaceur est un cas particulier de mise en forme du trafic qui assure en même temps la fonction de contrôle de l'UPC. Après avoir étudié dans les paragraphes précédents les mécanismes de contrôle de l'ATM tels que l'UPC ou la CAC, nous nous intéressons ici simplement aux mécanismes de shaping.

En premier lieu, la discipline WFQ (Weighted Fair Queueing) a pour objectif de garantir les délais des connexions prioritaires, tout en limitant les pertes des autres connexions. Pour cela, à chaque cellule d'une connexion est associé un "poids", i.e une priorité, qui constitue en fait la portion minimale de la file associée à la connexion. En fonction des

poids des cellules de chaque connexion et du nombre de cellules arrivées par connexion, le serveur WFQ calcule le temps auquel devront être servies les cellules entrant dans la file. Le principe est de garantir la même "quantité de service" à toutes les connexions. Cette quantité de service est relative au poids accordé à la connexion. Les principaux avantages de cette discipline reposent sur l'équité garantie entre les connexions et sur le fait que les flux prioritaires rencontrent des buffers virtuels suffisamment petits pour respecter leurs contraintes de délais. On lui reproche d'être souvent trop compliquée à mettre en œuvre. Pourtant, ce mécanisme a fait l'objet de nombreuses études et améliorations, menées notamment par la communauté TCP/IP (cf [BZ96]) dans le cadre de l'implémentation du service *Diffserv* destiné à offrir différentes classes de service dans les réseaux IP (cf [MRS99]). La discipline WFQ est de ce fait utilisable dans le cadre des réseaux ATM, comme en témoigne le document [CT98], et permet de ne pas avoir à implémenter un buffer par niveau de priorité. Un modèle d'implémentation de WFQ a été réalisé par le CNET et peut être trouvé dans [BC99].

Comme le précise le document [TOU], le mécanisme de Weighted Fair Queueing assure une réservation minimale ainsi qu'un partage équitable des pertes en proportion des trafics soumis par les différents flux. Ce mécanisme peut être pourvu d'extensions permettant d'accroître le contrôle des pertes et de gérer au mieux les flux prioritaires dans les buffers. En effet, il peut se révéler nécessaire de sélectionner, parmi les paquets d'un flux présents dans la file, celui de moindre importance et qui doit être préférentiellement rejeté. C'est le cas par exemple lorsque l'ensemble des cellules d'une même connexion se compose de cellules marquées, avec le bit CLP mis à 1, et de cellules non marquées. Les compléments proposés dans ce cas en faveur de la discipline WFQ sont les mécanismes de rejet sélectif. Un descriptif et une analyse détaillés de ces mécanismes sont donnés dans le document [LEM95]. Les deux principaux mécanismes à rejet sélectif sont les suivants :

- La file d'attente à tampon partagé ou Partial Buffer Sharing (PBS) : le buffer du commutateur est constitué de N places et d'un seuil de congestion K. L'objectif étant de protéger les cellules prioritaires (avec le bit CLP à 0), lorsque le nombre de cellules présentes dans le buffer dépasse le seuil K, seules les cellules prioritaires sont acceptées, dans la limite des N places disponibles. Ainsi, dès que le nombre de cellules présentes dans la file est supérieur à K, les cellules de priorité inférieure (CLP1) sont rejetées avant d'atteindre la file. En revanche, les cellules déjà admises dans le buffer ne peuvent pas en être retirées ultérieurement.

- L'algorithme du Push Out (PO) : tant que le buffer n'est pas saturé, les cellules des deux classes (CLP0 et CLP1) sont admises dans leur ordre d'arrivée et sont servies suivant la discipline FIFO (First In First Out, Premier Arrivé Premier Servi). En revanche, dès que toutes les places du buffer sont occupées, l'arrivée d'une cellule prioritaire provoque le rejet d'une cellule ordinaire déjà présente dans la file. Cette discipline est plus délicate à implémenter que la politique PBS car il faut détruire des cellules qui sont déjà dans le buffer, tout en respectant le séquencement des cellules d'une même connexion. De plus, le document [KHBG91] montre que les performances de la politique PO sont comparables à celles de la politique PBS.

Enfin, dans le cas de trafics variables où l'on peut avoir besoin ponctuellement d'un débit crête nettement supérieur au débit moyen, il peut être préférable de réserver ces ressources avant d'envoyer les données : la source informe le réseau avant de transmettre

une nouvelle rafale, ce qui permet au réseau de préparer les ressources nécessaires au transport de la rafale. C'est la solution que propose le CNET, d'après deux protocoles de réservation rapide de ressources spécifiés dans [BT91] :

- Le FRP-DT (Fast Reservation Protocol with Delayed Transmission) : la source attend une confirmation du réseau avant de transmettre la rafale. Dans ce cas, une probabilité de perte de cellules est garantie sur les rafales acceptées. Ce service s'applique plutôt au transfert de données et à l'interconnexion de réseaux.
- Le FRP-IT (Fast Reservation Protocol with Immediate Transmission) : la source envoie la rafale tout de suite après la cellule de réservation. Dans ce cas, la rafale est perdue si le réseau ne dispose pas de ressources disponibles.

Ce mécanisme n'a pas eu le succès escompté car il pose de nouveaux problèmes en terme de caractérisation des rafales, qui viennent se rajouter à la complexité déjà importante de la fonction CAC.

Nous avons décrit dans ce chapitre les principaux mécanismes propres à assurer un contrôle de trafic dans les réseaux ATM. Comme nous l'avons souligné, ces mécanismes ont en majorité été inventés indépendamment de l'ATM. Cependant, ils s'avèrent particulièrement bien adaptés au contrôle de trafic sur ce type de réseau. Notre but n'est pas d'analyser ces précédents mécanismes dans le détail mais de proposer une solution finale à France Télécom pour le contrôle de trafic dans les réseaux ATM. Nous allons voir à présent quels services peuvent être offerts par l'opérateur en utilisant certains de ces mécanismes.

Chapitre 3

Les capacités de transfert

Les mécanismes que nous avons présentés jusqu'ici doivent permettre l'acheminement des flux en fonction de leurs priorités, paramètres et objectifs de qualité de service respectifs. La couche ATM peut donc agir différemment selon les propriétés des flux transportés. Le but de ce chapitre est de définir et de classer les différents types de flux en fonction des paramètres de trafic et de la qualité de service prévue. Les flux sont donc rassemblés en plusieurs "capacités de transfert" adaptées à différents types d'applications. A l'aide de cette classification, les mécanismes définis précédemment peuvent établir des priorités et réagir distinctement en fonction des flux. De plus, d'autres mécanismes de contrôle de trafic, propres à chaque capacité de transfert (par opposition aux mécanismes précédents, communs à toutes les capacités), précisent la méthode de contrôle de trafic adaptée à chaque flux et aux engagements pris par l'opérateur. De ce fait, un opérateur peut associer plusieurs coûts à un même service en fonction de la capacité de transfert choisie. Nous détaillons dans ce qui suit les capacités de transfert principales définies par l'ATM Forum dans [ATM96] et l'UIT dans [ITU96] ainsi que les mécanismes associés à chacune de ces capacités.

3.1 Capacités prioritaires

Les capacités de transfert prioritaires sont celles pour lesquelles la bande passante est garantie : sous la condition que la source respecte les paramètres négociés avec le réseau dans le contrat de trafic, l'opérateur garantit la qualité de service demandée tout au long de la connexion. Deux capacités de transfert sont qualifiées de "prioritaires". Ce sont les seules capacités prévues actuellement sur la dorsale ATM de France Télécom.

3.1.1 La capacité de transfert DBR

La capacité DBR (Deterministic Bit Rate) est définie par l'UIT dans le but d'assurer le transfert de trafics isochrones dont les contraintes de délai et de débit acheminé sont fortes. Son correspondant à l'ATM Forum est le service CBR (Constant Bit Rate). Le seul paramètre de trafic négocié est le débit maximum auquel la source veut transmettre (PCR, Peak Cell Rate). La qualité de service inclut la probabilité de perte de cellules (CLR), le délai maximal de bout en bout (CDT) et la variation maximale de ce délai

(CDV). Cette qualité de service est garantie durant la connexion à tout flux qui respecte le débit d'émission maximal PCR. Un seul niveau de priorité est défini : la QoS n'est garantie que pour les cellules dont le bit CLP est à 0.

Cette capacité est très simple à implémenter : c'est une émulation de circuit, tout à fait adaptée à la téléphonie ou aux applications multimédia telles que la visio-conférence. De plus, pour un trafic variable où il est difficile d'estimer d'autres paramètres que le débit crête, cette capacité peut également être utilisée. Du fait de sa simplicité, France Télécom ne présentait initialement dans son offre ATM que cette capacité. Mais il est probable que le bénéfice de l'ATM ne saura être trouvé que dans la variabilité des services et des coûts que l'opérateur pourra offrir à l'utilisateur.

3.1.2 La capacité de transfert SBR

L'arrivée de Contrôleurs-Espaceurs sur le backbone ATM de France Télécom devrait permettre (cf [Fra98]) d'offrir des services à débits variables. C'est ce que propose l'UIT avec la capacité de transfert SBR (Statistical Bit Rate), dont le correspondant à l'ATM Forum est le service VBR (Variable Bit Rate). Ce type de capacité s'adresse aux utilisateurs générant des trafics très sporadiques et qui ne peuvent pas se permettre de payer une liaison au débit crête : l'envoi de trafic au débit PCR étant ponctuelle, l'utilisation de la capacité de transfert DBR serait peu rentable dans ce cas.

La capacité de transfert SBR offre la possibilité de négocier les paramètres PCR (Peak Cell Rate), SCR (Sustainable Celle Rate) et MBS (Maximum Burst Size) pour caractériser le trafic de la source (cf chapitre 2). Si la source respecte ces paramètres, le réseau garantira le taux de pertes (CLR) ainsi que le délai (CDT) et la variation de délai (CDV) dans le cas de trafics temps réel. Pour les trafics non temps réel, seule la probabilité de perte est garantie. Si l'option marquage est utilisée, l'UPC marquera toutes les cellules en excès du contrat SBR/MBS. Les cellules marquées, dont le bit CLP est à 1, n'auront aucune garantie en terme de qualité de service. Les cellules émises à un débit supérieur à PCR seront détruites, comme pour la capacité DBR.

La capacité de transfert SBR offre une plus grande flexibilité et une plus grande souplesse que la capacité DBR. Elle permet à France Télécom de diversifier son offre et de mieux profiter des possibilités d'ATM. Néanmoins, elle montre trois inconvénients majeurs :

- Une connexion SBR peut être contrôlée par l'UPC même si le réseau dispose des ressources suffisantes pour transmettre les cellules non conformes au contrat de trafic. C'est l'inconvénient du contrôle préventif exercé par l'UPC : les cellules sont marquées aveuglément et indépendamment de l'état du réseau.

- Garantir une QoS tout au long d'une connexion peut se justifier dans le cas de trafic temps réel ; mais cela conduit bien souvent à surestimer les ressources nécessaires. En effet, la fonction CAC étant difficile à concevoir dans le cas de trafics variables, elle aboutit souvent, comme nous l'avons vu, à la réservation de ressources en fonction du pire des cas prévisibles.

- Enfin, la caractérisation d'un trafic sporadique est délicate, autant du point de vue de la source que du point de vue du réseau. Le paramètre le plus simple à estimer est toujours le débit crête.

Ces inconvénients s'avèrent particulièrement pénalisants pour des applications de transfert de données ou d'interconnexion de réseaux, qui ne nécessitent pas de garantie de bande passante et ne peuvent pas non plus évaluer le trafic qui transitera sur le réseau. Afin de palier ces inconvénients, la capacité de transfert ABR a été introduite.

3.2 La capacité de transfert ABR

La capacité de transfert ABR (Available Bit Rate) doit permettre un partage dynamique de la bande passante disponible sur le réseau entre tous les utilisateurs ayant négocié ce service. L'objectif de cette capacité est de s'adapter à l'état du réseau en augmentant ou en diminuant le débit des sources. Il s'agit d'un mécanisme de contrôle préventif par débit, en vue d'une optimisation de l'utilisation du lien et d'un taux de perte des cellules faible. Le service ABR paraît donc tout à fait adapté au transfert de données ou à l'interconnexion de réseaux. Les mécanismes de contrôle de trafic générés par cette capacité de transfert sont l'objet des études que nous avons menées dans le cadre de ce doctorat. Aussi la présentation de cette capacité est-elle plus détaillée. Pour des compléments d'informations, le lecteur pourra se reporter aux ouvrages [ATM96], [RAB98] et [KG96].

3.2.1 Principe et intérêts

Le principe de la capacité de transfert ABR est d'adapter le trafic émis aux ressources disponibles dans le réseau plutôt que d'adapter les ressources au trafic prévu (cela n'interdit pas bien sûr un dimensionnement préalable des ressources du réseau). On ne garantit donc pas une qualité de service mais une meilleure utilisation de la bande passante : le réseau informe les sources ABR des limites du trafic qu'elles peuvent générer. Cela suppose d'une part que les sources ABR puissent s'adapter à l'état du réseau et restituer, ou au contraire consommer, une partie de la bande passante si besoin est : c'est le cas du transfert de données et de l'interconnexion de réseaux locaux. D'autre part, les sources utilisant l'ABR ne doivent pas requérir de bande passante garantie, ni de contraintes temps réel. C'est pour cette raison que la capacité de transfert ABR est qualifiée de service "best effort" et considérée moins prioritaire que les trafics DBR et SBR. Les capacités de transfert sont hiérarchisées de la manière suivante (cf figure 3.1) :

- Une partie constante de la bande passante est réservée aux connexions DBR.
- Une seconde partie, variable, est réservée aux connexions SBR.
- Le reste de la bande est disponible pour les connexions ABR et UBR (Unspecified Bit Rate). Le réseau calcule donc la bande passante non utilisée par les deux services précédents et régule ainsi équitablement le débit d'émission des sources ABR. La bande passante disponible par connexion est variable en fonction de la bande passante (variable également) garantie aux connexions SBR et DBR, et du nombre de connexions ABR acceptées.

Le principal intérêt de la capacité de transfert ABR est de ne pas perdre de vue la notion de qualité de service propre aux réseaux ATM. Cela peut paraître contradictoire. En fait, le souci d'ABR est de protéger le réseau et de respecter la qualité de service des connexions

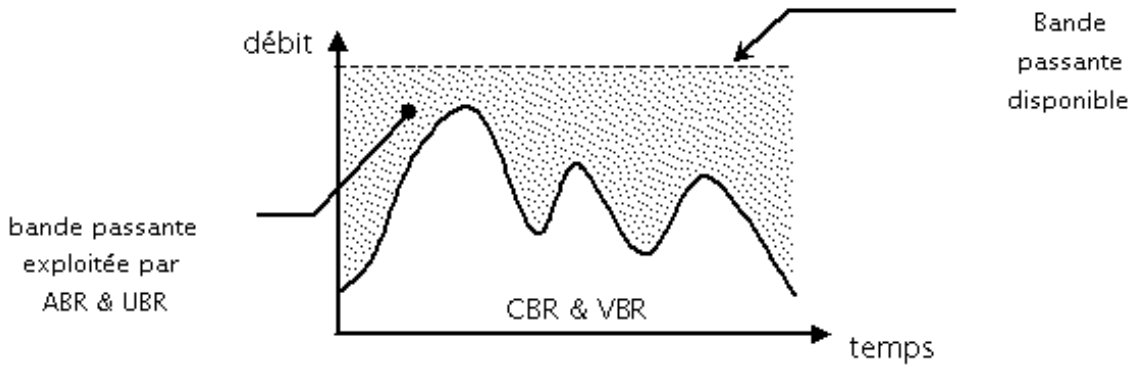


FIG. 3.1 - *Adaptation dynamique à la bande passante disponible*

prioritaires, tout en récupérant rapidement la capacité restante pour les autres connexions. La réunion de ces deux avantages n'est pas envisageable avec le protocole TCP ni avec la capacité de transfert UBR décrite au paragraphe 3.3. Dans le premier cas, TCP transmet jusqu'à la perte donc jusqu'à ce qu'il sature le buffer du réseau. Ceci peut provoquer des pertes de cellules prioritaires ou, si des mécanismes de rejet sélectif ou de priorisation sont utilisés, des pertes importantes du côté du trafic TCP. Les deux alternatives sont peu enviables. Dans le cas d'UBR, les cellules sont envoyées dans le réseau sans contrôle. Les conséquences de cet envoi sont bien souvent désastreuses (cf [Fra98]). Dans le meilleur des cas, le protocole TCP (ou un autre mécanisme de contrôle de trafic existant) est utilisé et nous revenons au problème précédent mais au moins les cellules sont retransmises. Dans le pire des cas, les cellules sont irrémédiablement perdues. Dans tous les cas, le réseau n'est pas protégé.

La capacité de transfert ABR résout ce problème en réagissant à l'état du réseau et en retenant les cellules au niveau de la source par limitation du débit d'émission de cette source. Il importe de préciser que l'ABR n'a pas pour objectif d'être interfacé à toutes les stations de travail d'un même réseau local. Pour cela, les mécanismes existants suffisent amplement. Le mécanisme ABR sera plutôt implémenté à la sortie d'un réseau local, pour contrôler et factoriser le trafic sortant et l'adapter à la capacité disponible sur le lien de sortie. Par exemple, une centaine d'applications TCP communiquant d'un même site vers l'extérieur sont difficilement contrôlables et saturent rapidement le lien de sortie. L'intérêt d'une source ABR est précisément de récolter les cellules de ces cent connexions TCP dans les buffers d'accès et de contrôler l'envoi de ces cellules sur le réseau. C'est en cela qu'ABR joue pleinement son rôle de protection du réseau et de respect de la qualité de service garantie aux autres connexions.

3.2.2 Les mécanismes de l'ABR

Comme toutes les capacités de transfert, l'utilisateur et le réseau doivent établir un contrat de trafic. Au niveau de la source, les paramètres négociés sont le débit crête (PCR), le débit minimal (MCR) et le débit initial (Initial Cell Rate, ICR). Il n'y a pas de garantie de bande passante au-delà du débit minimal MCR, qui peut être choisi nul (ce qui a pour effet de simplifier la fonction CAC, puisque toutes les connexions peuvent alors être acceptées). Le débit calculé en fonction des informations reçues de la part du réseau est le débit maximal autorisé (Allowed Cell Rate, ACR) ; il est compris entre MCR

et PCR. Les cellules qui ne respectent pas ce débit sont déclarées non conformes selon l'algorithme DGCRA décrit plus haut ; elles sont alors marquées ou détruites. En ce qui concerne la qualité de service, l'ATM Forum ne spécifie pas de garantie ferme pour l'ABR : l'utilisateur peut s'attendre à un faible taux de perte et à une relative équité entre les connexions. A l'UIT, une probabilité de perte maximale (CLR) ainsi qu'une garantie d'équité et d'isolation des connexions sont assurées aux cellules conformes.

Nous décrivons dans ce qui suit les mécanismes mis en œuvre pour la capacité de transfert ABR. L'ATM Forum et l'UIT diffèrent par certains points dans la normalisation de ces mécanismes. Nous présentons ici simplement les grandes lignes, communes aux deux spécifications.

3.2.2.1 Les cellules RM

Nous avons vu que le principe de l'ABR repose sur la réaction de la source aux informations provenant du réseau : ces informations indiquent à la source l'état de congestion du réseau et lui permettent de déduire le débit auquel elle doit transmettre. Ce retour d'information est transporté par des cellules particulières : les cellules RM (Resource Management). Le flux des cellules RM est inséré par la source dans le flux des cellules de données. Les éléments du réseau (commutateurs, destination) peuvent exceptionnellement insérer des cellules RM afin de diminuer le temps de réaction de la source. Chaque flux d'information de la source vers le destinataire est accompagné d'une boucle de contrôle comprenant deux flux de cellules RM, un dans la direction aller (forward direction, de la source vers la destination), et un dans le sens retour (backward direction, de la destination vers la source). Les cellules RM sont modifiées lors de leur transit dans le réseau et retournées vers la source par la destination afin de l'informer de la disponibilité de la bande passante, du niveau ou de l'imminence d'une congestion (cf figure 3.2).

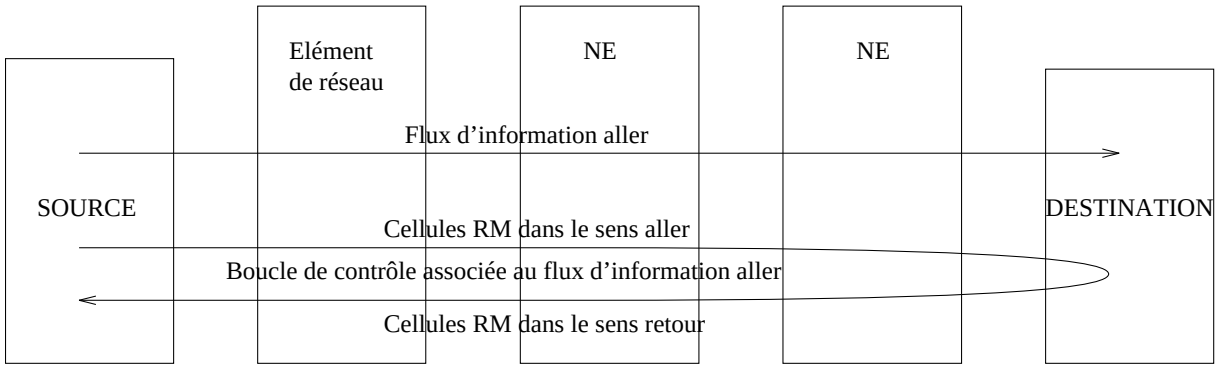


Fig. 3.2 - *Boucle de contrôle ABR (feedback loop)*

Une option future du service ABR sera de partitionner une connexion en plusieurs segments de contrôle : on raccourcira alors la boucle de contrôle en utilisant des sources et des destinations virtuelles. Cette fonctionnalité permettra un contrôle plus étroit du trafic, car les informations contenues dans les cellules RM reviendront plus rapidement aux sources virtuelles.

Le tableau 3.1 indique les champs spécifiés dans les cellules RM pour le transport des informations, c'est-à-dire le transport du message que le réseau veut faire passer à la source. Dans la plupart des cas, une cellule RM "in-rate" est insérée par la source dans

le flux d'information toutes les N_{rm} cellules de données. Si une cellule RM ne respecte pas cette contrainte, elle sera marquée et appelée cellule RM hors bande ("out-of-rate") ; comme nous l'avons précisé plus haut, cette insertion de cellule hors bande peut être nécessaire pour informer plus rapidement la source d'un état de congestion. Toutes les cellules RM empruntent le même chemin que les cellules de données.

DIR	DIRection : indique s'il s'agit d'une cellule RM "forward", circulant dans le même sens que le flux de données auquel elle est associée, ou "backward"
BN	Backward explicit congestion Notification : si le bit BN est mis à 1, cela signifie que la cellule RM a été générée par un commutateur ou une destination. Pour des cellules RM émises par la source, le bit BN est laissé à 0
CI	Congestion Indication : lorsque le bit est mis à 1, le réseau souhaite indiquer une congestion à la source. Les cellules RM "in-rate" quittent la source avec le bit CI à 0
NI	No Increase : lorsque le bit est mis à 1, le réseau recommande à la source de ne pas augmenter son débit. Les cellules RM "in-rate" quittent la source avec le bit NI à 0
ER	Explicit Rate : si les commutateurs disposent de la fonctionnalité nécessaire (voir plus loin), ils peuvent indiquer à la source le débit explicite auquel elle doit transmettre. Dans le cas contraire, le champ ER est laissé à la valeur PCR
CCR	Current Cell Rate : informe les différents éléments du réseau du débit de la source, instantané ou moyen. Pour les cellules générées par le réseau, $CCR = 0$. Ce champ peut être utilisé par les commutateurs pour calculer l'Explicit Rate
MCR	Minimum Cell Rate : informe les différents éléments du réseau du débit minimal négocié pour la source

TAB. 3.1 - *Les champs principaux de la cellule RM*

3.2.2.2 Les paramètres du services ABR

Avant de présenter les mécanismes mis en jeu pour la capacité de transfert ABR, nous souhaitons introduire dans le tableau 3.2 les nombreux paramètres implémentés dans chaque unité de contrôle ABR. Nous rappelons qu'une unité de contrôle ABR fait partie de l'offre de service de la couche ATM et se situe donc au niveau du contrôle de trafic de la couche ATM du terminal équivalent. Le rôle et l'influence des paramètres ABR est un des objets de l'étude que nous avons menée.

PCR	Peak Cell Rate : débit crête que la source ne peut excéder	exprimé en cellules par seconde
MCR	Minimum Cell Rate : débit minimal auquel la source est toujours autorisée à transmettre	cells/s
ICR	Initial Cell Rate : débit initial, au début d'une connexion ou après une période d'inactivité	cells/s
ACR	Allowed Cell Rate : débit courant auquel la source est autorisé à transmettre. Compris entre MCR et PCR	cells/s
RIF	Rate Increase Factor : contrôle l'augmentation du débit ACR à la réception d'une cellule RM	2^{-n} $n \in \{0, ..., 15\}$
RDF	Rate Decrease Factor : contrôle la diminution du débit ACR à la réception d'une cellule RM	2^{-n} $n \in \{0, ..., 15\}$
N_{rm}	Nombre maximal de cellules de données séparant l'envoi de deux cellules RM in-rate	2^n cellules $n \in \{0, ..., 8\}$ Défaut : 32 cellules
M_{rm}	Si le temps écoulé depuis l'envoi de la dernière cellule RM est supérieur à T_{rm}, la source émet une cellule RM toutes les M_{rm} cellules de données	Défaut : 2 cellules
T_{rm}	Temps maximal entre l'envoi de deux cellules RM consécutives	millisecondes 100×2^{-n} ms $n \in \{0, ..., 7\}$
CRM	Missing RM-cell Count : nombre maximal de cellules RM-forward envoyées sans avoir reçu de cellules RM-backward	cellules
CDF	Cutoff Decrease Factor : contrôle la décroissance de l'ACR, lorsque le nombre CRM est atteint	2^{-n} $n \in \{0, ..., 6\}$
ADTF	ACR Decrease Time Factor : temps maximum écoulé entre l'envoi de deux cellules RM consécutives. Une fois ce seuil dépassé, l'ACR est ramené à la valeur ICR	secondes Granularité = 10 ms $0,01 \leq ADTF \leq 10,23$
TCR	Tagged Cell Rate : débit auquel la source peut envoyer des cellules RM hors bande	Egal à 10 cells/s
TBE	Transient Buffer Exposure : nombre de cellules que la source peut envoyer lors de l'initiation ou du redémarrage d'une connexion, avant que la première cellule RM revienne	cellules
FRTT	Fixed Round Trip Time : somme des délais de	10^{-5} secondes

page suivante

	propagation aller-retour depuis la source jusqu'à la destination	Granularité = 10^{-5} s

TAB. 3.2 - *Les paramètres de la capacité de transfert ABR*

Les informations transportées dans les cellules RM, conjuguées aux valeurs de ces paramètres, induisent un certain nombre de comportements au niveau de la source ABR, que nous détaillons ci-dessous.

3.2.2.3 Comportement de la source

L'ATM Forum, dans le document [ATM99], spécifie de manière détaillée le comportement de chacun des éléments intervenant dans la boucle de contrôle de congestion (source, nœud de réseau, commutateur, destination, source et destination virtuelles). La conformité de ces comportements est vérifiée à l'entrée du réseau par l'UPC et une fonction de contrôle de débit dynamique, décrite dans le paragraphe 2.2.2.1. Certaines libertés sont néanmoins laissées à la charge du fabricant, notamment sur le choix des politiques de calcul du partage des ressources, la gestion des files d'attente et les disciplines de service. Nous nous contentons dans les trois paragraphes suivants de rappeler les principes de fonctionnement de chacun des éléments participants au contrôle.

Au niveau de la source, la valeur prépondérante est l'Allowed Cell Rate : cette valeur induit le débit auquel la source est autorisée à transmettre. Elle est calculée à partir des informations ramenées à la source par les cellules RM. L'ACR ne doit jamais dépasser le PCR ni descendre en-dessous du MCR. A l'initiation d'une connexion, l'ACR est fixé à ICR.

Avant l'émission d'une cellule RM-forward, la source doit suivre les procédures suivantes :

- Une cellule RM "in-rate" ne peut être émise qu'après l'envoi de N_{rm} cellules de données consécutives ou lorsqu'un temps T_{rm} s'est écoulé et que M_{rm} cellules de données consécutives ont été émises. Afin d'éviter des situations de famine, c'est-à-dire des situations où aucune cellule de données, donc aucune cellule RM, ne peut être envoyée, la source a la possibilité d'envoyer des cellules RM marquées au débit TCR.
- Si le temps écoulé depuis la dernière émission d'une cellule RM-forward est supérieur à ADTF, l'ACR doit être ramené à la valeur ICR.
- Si au moins CRM cellules RM-forward ont été envoyées depuis la réception de la dernière cellule RM-backward, l'ACR doit être réduit de $ACR \times CDF$.
- La nouvelle valeur de l'ACR doit être placée dans le champ CCR de la cellule RM-forward afin de pouvoir être exploitée par les autres éléments du réseau. Les cellules de données transmises à la suite de cette cellule RM doivent respecter le débit ACR.

A la réception d'une cellule RM-backward, la source doit modifier son débit ACR comme suit :

- si le champ CI est à 1, l'ACR doit être réduit de $ACR \times RDF$;
- si le champ NI est à 1, l'ACR doit rester inchangé ;

- si ces deux champs sont à 0, l'ACR peut être augmenté de $RIF \times PCR$;

- si le champ ER a été modifié par un des éléments du réseau, c'est-à-dire si sa valeur n'est plus égale à PCR, l'ACR doit être ramené au minimum de la valeur calculée lors des trois procédures précédentes et de la valeur contenue dans le champ ER.

Notons que la source interprète uniquement les informations contenues dans les cellules RM pour le calcul du débit ACR.

3.2.2.4 Comportement de la destination

Le comportement général d'une destination ABR est relativement simple. Son rôle consiste à retourner les cellules RM-forward qu'elle reçoit vers la source, en les convertissant en cellules RM-backward. Pour cela elle modifie le champ DIR de la cellule RM.

Si la dernière cellule de donnée reçue indique une congestion (par l'intermédiaire du bit EFCI, Explicit Forward Congestion Indication), la destination recevant une cellule RM-forward marque son bit CI à 1 et la retourne vers la source. Le bit EFCI de la cellule de donnée est en revanche remis à 0. Si les bits CI ou NI sont déjà mis à 1, la destination ne les modifie pas. En revanche, une destination peut informer la source de ses propres congestions en modifiant le champ ER, ou en mettant à 1 les bits CI ou NI. Elle peut également générer des cellules RM-backward, dans la bande ou hors bande (i.e à un débit inférieur à TCR et avec le bit CLP mis à 1). L'objectif principal de la destination reste toujours de faire remonter l'information vers la source.

3.2.2.5 Comportement des commutateurs

Les commutateurs représentent souvent les goulots d'étranglement des connexions, c'est-à-dire les nœuds présentant les débits de sortie les plus faibles, compte tenu des autres flux. Leurs ports de sortie disposent chacun d'au moins une file d'attente, qui présente un risque de congestion. Les mécanismes de prévention des congestions au niveau des commutateurs, et plus généralement au niveau de chacun des éléments du réseau, sont donc particulièrement stratégiques. Différents mécanismes, plus ou moins complexes et plus ou moins efficaces, peuvent être utilisés ; les constructeurs ayant un large degré de liberté concernant la conception des algorithmes, une multitude de schémas de commutateurs est actuellement proposée. Leurs modes de fonctionnement sont répertoriés selon la capacité de l'élément de réseau à envoyer, par l'intermédiaire des cellules RM, des informations pertinentes concernant son état de congestion. Toutefois, un certain nombre de qualités communes sont souhaitées :

- les ressources du réseau doivent être partagées de manière équitable (cf paragraphe 3.2.3) entre les connexions ABR ;

- les commandes du réseau doivent permettre la convergence vers un état d'équilibre, i.e elles doivent garantir la stabilité du réseau ;

- le taux de perte de cellules doit être minimal.

Nous définissons dans les paragraphes suivants les deux types de commutateurs les plus répandus actuellement. Nous représentons également en figure 3.3 un exemple de cheminement des cellules RM à travers différents éléments de réseau.

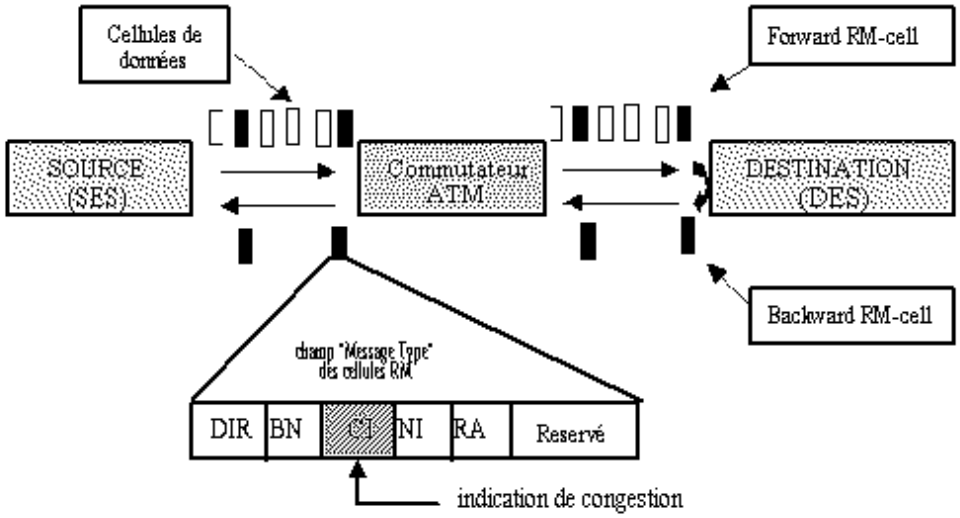


FIG. 3.3 - *Cheminement des cellules RM*

Commutateurs binaires : Dans cette catégorie sont regroupés les commutateurs basés EFCI et le "Relative Rate Marking". Dans le premier cas, le commutateur modifie le bit EFCI des cellules de données et le met à 1 lors d'une prévention de congestion. Ce bit sera interprété ensuite par la destination. Dans le second cas, le commutateur marque à 1 les bits CI ou NI des cellules RM lorsqu'un risque de congestion est détecté. Ce marquage peut être effectué à l'aller ou au retour, i.e sur les cellules RM-forward ou RM-backward. Lorsqu'il est effectué dans le sens retour, la source obtient les informations les plus récentes concernant l'état du nœud. L'élément de réseau peut lui-même générer des cellules RM-backward dans la bande (CLP=0) ou hors bande (au plus TCR cellules/s avec le bit CLP=1) pour accélérer le retour des informations à la source.

Il est également possible d'implémenter une politique d'allocation propriétaire, et de contrôler le taux d'utilisation du lien par VC (Virtual Connection), afin de garantir une meilleure équité en pénalisant les sources trop gourmandes ou inversement de récupérer pour d'autres la bande passante non utilisée par certaines connexions.

La stratégie finale adoptée dépend encore une fois du constructeur. Nous remarquons simplement que l'utilisation du bit NI est peu courante. L'étude [LR96] indique pourtant que ce bit permet de réduire la vitesse de croissance des files d'attente, ainsi que l'amplitude des oscillations des débits des sources. Cependant, lorsque l'instruction No Increase est appliquée, les sources n'ont pas le droit d'augmenter leur débit ; une nouvelle connexion ne peut donc pas augmenter (ni diminuer) son débit au-delà du débit initial et ne bénéficie pas du même débit autorisé que les autres connexions. Nous assistons alors à un phénomène d'inéquité du mécanisme No Increase à l'encontre des nouvelles connexions. Ce problème a condamné l'utilisation du bit NI dans le cas général.

Détection de congestion : Le Relative Rate Marking dépend de l'état de congestion du nœud, c'est-à-dire de la saturation d'une file d'attente d'un élément de réseau. Comme il s'agit d'un mode de fonctionnement binaire, l'élément de réseau indiquera simplement s'il y a un risque de congestion ou non. Le problème qui nous intéresse ici est de préciser les moyens dont dispose un commutateur pour décider s'il y a risque de congestion. Ces moyens sont décrits dans [RIT97a] et [CB96]. Nous présentons ici trois des schémas les plus courants :

- Le schéma basé sur la taille de la file d'attente. Lorsque la taille de la file d'attente dépasse un seuil Q_H, l'alerte est donnée et les bit EFCI et CI sont marqués. Si la taille de la file d'attente descend en-dessous du seuil Q_L, l'élément de réseau considère qu'il n'y a aucun risque de congestion dans son buffer et ne modifie pas les champs EFCI, CI ou NI. Si la taille de la file d'attente est comprise entre Q_L et Q_H et que le bit NI est utilisable, le commutateur considère être autour de son point de fonctionnement optimal, et en informe la source par l'intermédiaire du bit NI. Si le bit NI n'est pas utilisable, le commutateur continue à marquer les cellules s'il était auparavant dans un état de congestion, et ce jusqu'à ce que la file d'attente soit descendue en-dessous de Q_L. Si, à l'opposé, nous provenons d'un état où le buffer n'était pas saturé, le commutateur ne marque pas les cellules tant que la taille de la file d'attente est inférieure à Q_H.

- Le schéma basé sur le débit ACR. De la même manière, nous utilisons deux seuils Q_L et Q_H pour déceler un risque de congestion. Ces seuils ne représentent plus la longueur absolue de la file d'attente (égale à Q), mais la longueur relative au débit d'arrivée des cellules de chaque connexion active au commutateur, dont le débit de sortie est évalué à C_F :

$$Q_{ACR} = Q \cdot \frac{\sum_{\text{connexions actives}} ACR}{C_F}$$

- Le schéma basé sur la croissance de la file d'attente utilise également deux seuils Q_L et Q_H, représentant respectivement la longueur minimale et maximale de la file d'attente en fonction de la vitesse de croissance de cette file. Ils sont donc rapportés à la formule :

$$Q_{dQ} = Q \cdot \left(\frac{dQ/dt}{C_F} + 1 \right)$$

L'avantage des commutateurs binaires réside dans leur simplicité. Cependant, leurs performances dépendent beaucoup du choix des paramètres ABR et de la valeur des seuils Q_L et Q_H. Il est difficile de s'orienter vers un choix optimal dans une configuration de réseau qui peut être variable. L'objectif des parties II et III est précisément de guider l'opérateur dans ce choix.

Commutateurs basés "Explicit Rate" : Lorsque le mode de fonctionnement "Explicit Rate Marking" est implémenté, l'élément de réseau évalue la charge de son nœud, i.e le

taux de remplissage de ses buffers, et calcule une répartition équitable, appelée "Fairshare", de la bande passante entre les connexions (voir la définition de l'équité au paragraphe 3.2.3). Cette répartition notifie le débit alloué (Explicit Rate) par l'élément de réseau à chacune des connexions ABR. Le champ ER des cellules RM-forward ou RM-backward de chaque connexion est mis à jour avec cette valeur de débit. La cellule RM rapporte à la source le débit explicite le plus faible rencontré sur le réseau.

Les commutateurs à débit explicite peuvent cohabiter avec les commutateurs binaires. Le débit ACR autorisé à la source est le minimum de la valeur du champ ER de la cellule RM et de l'ACR calculé à partir des indications de congestion.

Plusieurs mécanismes d'allocation sont envisagés pour estimer la valeur du débit explicite. Nous pouvons citer l'algorithme EPRCA (Enhanced Proportional Rate Control Algorithm) décrit dans [ATM96] ou l'algorithme PRQCA (Proportional Rate and Queue Control Algorithm) décrit dans [JK96]. Même si aucune normalisation n'est effective, il semble cependant que le mécanisme le plus en vogue actuellement est l'algorithme ERICA (Explicit Rate Indication for Congestion Avoidance) présenté par Raj Jain dans [JKG+96] et [KJF+96].

ERICA : L'algorithme ERICA détermine pendant un intervalle périodique (l'Average Interval Length, AIL) le facteur de charge du lien (Overload Factor, OF), la bande passante disponible pour le service ABR (Capacité ABR), et le nombre de VC actifs. Ce nombre de VC actifs pendant l'intervalle permet de calculer le "Fair Share" (FS) :

$$FS = \frac{\text{Capacité ABR}}{\text{Nombre de VC actifs}}$$

Le facteur de charge est calculé à partir du débit entrant :

$$OF = \frac{\text{Débit ABR entrant}}{\text{Capacité ABR}}$$

La capacité ABR est fonction de l'objectif que l'on s'est fixé concernant le taux d'utilisation du lien (Target Utilization) :

Capacité ABR = Taux d'utilisation visé × Capacité du lien
- Capacité utilisée par les connexions VBR
  - Capacité utilisée par les connexions CBR

Dans le cas des commutateurs ERICA, le taux d'utilisation du lien visé est compris entre 85 % et 95 % de la capacité totale. Afin d'atteindre un taux d'utilisation maximal (100 % de la bande passante), la version améliorée ERICA+ fixe le taux d'utilisation visé à 1, et introduit la taille de la file d'attente comme paramètre de contrôle. A cet effet, l'algorithme requiert un objectif concernant la taille de la file d'attente (Target Queue Length, Q). La capacité ABR est alors révisée comme suit :

$$\text{Capacité ABR} = \text{Capacité ABR} \times \frac{b \cdot Q}{(b-1) \times \text{Taille de la file d'attente} + Q}$$

Les valeurs recommandées pour le paramètre b sont, d'après [AHAEE99] :

$$b = \begin{cases} \min\{1,15, \frac{\pi Q}{2*ER*\tau}\} & \text{si Taille de la file} > Q \\ \min\{1,05, \frac{\pi Q}{6*ER*\tau}\} & \text{sinon} \end{cases}$$

où ER est égal à la valeur du champ ER de la cellule RM et τ est égal au délai aller-retour d'une cellule.

Le débit explicite signalé à la source par l'intermédiaire des cellules RM vaut :

$$ER_{calculated} = \max\left\{FS, \frac{CCR(VC)}{OF}\right\}$$

$CCR(VC)$ est le débit entrant des cellules du VC ; il peut être estimé par le commutateur ou relevé dans le champ CCR des cellules RM.

Les commutateurs basés "Explicit Rate" permettent une meilleure utilisation et une meilleure répartition de la bande passante. La complexité du paramétrage de l'ABR dans le cas des commutateurs binaires est relâchée, au prix d'une plus grande complexité d'implémentation. En outre, le nombre de tâches exécutées par les commutateurs augmente, ce qui va à l'encontre du concept des réseaux ATM, qui est de repousser les fonctionnalités complexes à la périphérie du réseau. Toutefois, le volume des cellules RM étant réduit, l'exécution de ces fonctions devrait être possible.

L'"Explicit Rate Marking" étant à présent disponible sur le marché, il serait regrettable de s'en priver, compte tenu de ses performances honorables et de son utilisation "immédiate", dans le sens où le paramétrage est simple. Le problème réside toutefois dans la différence de prix entre un commutateur binaire et un commutateur ERICA.

3.2.3 Equité

L'un des objectifs de l'ABR est de réaliser un partage équitable de la capacité disponible entre les connexions ABR. Mais l'équité entre les connexions d'un réseau est une notion difficile à définir. En effet, comment pourrait-on parler d'un système équitable lorsqu'aucune des connexions de ce système n'est équivalente à une autre, ne serait-ce que par les différents chemins qu'elles empruntent ou par les ressources qu'elles requièrent ? Quelle serait dans ce cas la manière de définir un partage équitable de la bande passante (BP) entre les connexions ? Du point de vue d'un opérateur, l'engagement sur le partage des ressources doit simplement être suffisant pour garantir un débit et une QoS acceptables à toutes les connexions. Néanmoins, l'UIT et l'ATM Forum recommandent que l'opérateur s'assure qu'aucun élément de réseau implémentant le mécanisme ABR ne favorise ou discrimine un ensemble de connexions particulier. A cet effet, cinq politiques

de respect de l'équité ont été définies dans [ATM99]. Ces algorithmes se rapprochent du principe d'équité au sens Max-Min, décrit dans [MR98]. L'objectif de ces critères est de déterminer la bande passante allouée à chaque connexion :

- partage Max-Min : $BP(i) = BP/n$, où $BP(i)$ désigne la bande passante de la connexion i, BP la bande passante disponible à partager entre toutes les connexions non contraintes, et n le nombre de connexions actives sur le nœud ;
- partage après garantie du MCR :
  $BP(i) = MCR(i) + \frac{BP - \sum_{i=1}^{n} MCR(i)}{n}$
- allocation du maximum de MCR et du partage Max-Min :
  $BP(i) = \max\{MCR(i), BP/n\}$
- allocation pondérée, proportionnelle à MCR :
  $BP(i) = BP \times \frac{MCR(i)}{\sum_{i=1}^{n} MCR(i)}$.

Notons que le terme $BP(i)$ représente en fait l'Explicit Rate auquel chaque source devrait émettre.

Afin d'assurer le partage équitable de la bande passante entre les connexions ABR, certains commutateurs proposent également un algorithme gardant la trace des débits courants de toutes les connexions traversant le commutateur. Lorsqu'une congestion est détectée, la bande passante allouée aux connexions est partagée en utilisant la fonction d'équité suivante, définie dans [MBLCM96] :

$$f(i) = \frac{ACR(i) - MCR(i)}{PCR(i)}$$

où i représente l'indice de la connexion.

Le commutateur calcule alors la valeur moyenne de la fonction de partage équitable de la bande passante du lien parmi toutes les connexions de ce lien :

$$BP(i) = \frac{1}{n} \sum_{i=1}^{n} f(i)$$

Cette définition de l'équité s'applique tout particulièrement aux commutateurs binaires. Si un commutateur est dans une situation de congestion, le bit CI d'une cellule RM n'est mis à 1 que si, en plus, cette cellule RM appartient à une connexion pour laquelle $f(i) \geq BP(i)$. Si la cellule RM appartient à une connexion pour laquelle $f(i) < BP(i)$, alors c'est le bit NI qui est mis à 1.

Cet algorithme est nommé "Fair". Sans utiliser cet algorithme, l'obtention d'un partage équitable de la bande passante avec des commutateurs binaires n'est réalisable qu'après un choix particulièrement fin des facteurs de croissance et de décroissance du débit de la source (RIF et RDF). Dans le cas contraire, il en résulte une iniquité marquée entre les diverses connexions, et ce pour une large gamme de choix des paramètres du système. L'algorithme *Fair* autorise donc une marge de choix des paramètres RIF et RDF beaucoup plus large.

3.2.4 Récapitulatif des études menées sur la capacité ABR

Nombreuses sont les études par simulation réalisées sur le mécanisme ABR. Les phénomènes ou problèmes principalement observés sont résumés dans les documents [OMS+97] et [FCB96]. Nous souhaitons mentionner ici les plus importants :

- Les longs chemins, présentant beaucoup de nœuds à traverser, montrent des problèmes d'équité.
- De meilleurs performances sont généralement obtenues avec les commutateurs à débit explicite, au prix d'une complexité plus importante des tâches des commutateurs.
- Les contextes WAN exigent des générations fréquentes de cellules RM, afin que les informations ne soient pas désuettes à leur arrivée à la source.
- Le mécanisme ABR est très sensible aux valeurs des paramètres N_{rm}, RIF, RDF, ICR, MCR et PCR.
- La segmentation d'un réseau en Virtual Sources et Virtual Destinations (cf section 3.2.2.1) augmente l'équité.

Les études analytiques de ce mécanisme sont naturellement plus rares. L'étude de référence est certainement celle de M. Ritter (cf [RIT96] et [RIT97b]). Elle a été reprise dans [COL96] et [AHA99] et constitue un point de départ dans la seconde partie de cet ouvrage. Les études [BS96], [OMM98] et [YH94] reprennent l'approche différentielle développée dans [RIT96] et l'adaptent à un cas de dimensionnement de réseau particulier. Nous détaillerons ces analyses dans la partie II.

3.3 Les capacités de transfert UBR et GFR

La dernière capacité de transfert définie par l'ATM Forum dans [ATM96] est la capacité UBR. Avec cette capacité, aucun contrôle n'est effectué par le réseau et aucune QoS n'est garantie : la source envoie ses données à ses risques et périls. C'est le service le moins coûteux et le plus simple. UBR est parfois utile lorsqu'un mécanisme de contrôle de trafic tel que TCP est déjà implémenté et qu'il n'est donc pas nécessaire de transférer le trafic à l'aide de la capacité ABR. Toutefois, cette politique est très controversée, en raison de l'inadaptation de TCP à ATM et des meilleures performances obtenues lors d'une utilisation de TCP avec la capacité ABR (cf section 4.4). Néanmoins, compte tenu de sa simplicité, France Télécom a décidé de lancer une offre UBR.

Une amélioration de ce service a été proposée dans [ATM99] et [GH96] afin de minimiser les pertes lors de transferts à charge moyenne. Cette nouvelle capacité de transfert, nommée GFR (Guaranteed Frame Rate) requiert que les cellules de données soient organisées sous la forme de trames assemblées dans la couche ATM. Il garantit un débit minimal (Minimum Cell Rate) sous réserve que la taille maximale des trames ne dépasse pas MFS (Maximum Frame Size) cellules, et que la longueur des rafales ne dépasse pas MBS (Maximum Burst Size) cellules (voir le chapitre 2) envoyées au débit MCR. Si cette contrainte est respectée, l'utilisateur peut s'attendre à ce que ses trames soient délivrées par le réseau avec un minimum de pertes. Ce service permet d'obtenir une qualité de

service minimale lorsque le réseau est congestionné, tout en autorisant un transfert à un débit supérieur lorsque les ressources sont disponibles. Toutefois, la volonté d'assurer une garantie dure associée au MCR impose de réserver la bande passante correspondante et donc de la faire payer au client. De plus, en l'absence de retour d'informations, il est probable que l'utilisateur ne pourra pas se rendre compte qu'il doit réduire son débit, ce qui aboutira encore à des pertes de cellules nombreuses en cas de fortes charges. Le problème d'UBR n'est donc que partiellement résolu par GFR, au prix d'une complexité et d'un coût supérieurs. Ces inconvénients pourraient intervenir en la faveur du mécanisme ABR.

Après avoir défini les différents mécanismes et capacités de transfert proposées par l'ATM Forum pour le contrôle du trafic, nous avons montré lesquels s'avéraient les plus aptes à transférer les données sur ce mode transfert. Cependant, comme nous l'avons précisé, les solutions classiques développées sur les réseaux existants pour contrôler le trafic de données ne seront pas amenées à disparaître et les solutions nouvelles devront savoir composer avec ces protocoles. Nous analysons dans le chapitre suivant quels problèmes ou interactions peuvent survenir du fait de la superposition de deux générations de mécanismes sur ATM.

Chapitre 4

Interconnexion de TCP/IP et d'ATM

Nous abordons dans ce chapitre les conséquences liées à l'intégration des protocoles TCP/IP sur les réseaux ATM. En effet, la superposition de ces protocoles entraîne un certain nombre de problèmes dûs notamment à leurs architectures distinctes, aux différents formats d'adressage, à la conversion des paquets en cellules, et aux mécanismes de contrôle de trafic concurrents. Nous résumons ces problèmes dans les sections suivantes.

4.1 Intégration de TCP sur ATM

Ce paragraphe traite d'un problème indirectement lié au contrôle de trafic et qui n'est pas le sujet de notre étude. Néanmoins, il est important, dans le cadre de l'interconnexion de TCP/IP sur ATM, de mentionner les techniques liées précisément à la superposition des architectures de protocoles.

L'intégration de IP sur ATM passe par la résolution technique des problème d'encapsulation des paquets, de résolution d'adresses, de routage et de qualité de service. Le cas d'ATM n'est pas exceptionnel : pour chaque technologie sur laquelle est implémenté IP, il est nécessaire de spécifier comment le protocole IP est transcrit à la couche liaison. Les documents [DAV98] et [REK97] donnent un aperçu des techniques d'intégration proposées pour IP sur ATM. La première, nommée Classical IP, consiste simplement à rester aussi près que possible de l'architecture IP. Les routeurs effectuent uniquement des conversions d'adresses IP en adresses ATM. Cette technologie ne permet pas de profiter de la commutation ATM.

Suite aux propositions "IP Switching" et "Tag Switching" destinées à autoriser une "commutation" IP sur ATM par l'intermédiaire de labels, CISCO a standardisé dans [LF98] le protocole MPLS (Multiprotocol Label Switching) implémenté aujourd'hui par Nortel. L'objectif de ce protocole est la commutation de labels, c'est-à-dire l'accélération des temps de traitement au niveau des routeurs par l'association de labels à certains flux. Cette technique permet de profiter des techniques de commutation, sans nécessiter un changement complet des équipements du réseau. De plus, elle doit assurer une évolutivité du réseau en termes de flux, de nombre de nœuds et de routeurs. Malheureusement, la commutation proposée est une commutation de niveau 3, c'est-à-dire que les routeurs

ne disparaissent pas ; on leur associe simplement un algorithme de "forwarding" pour accélérer la distribution des paquets à partir des labels. Ces routeurs représentent toujours les goulots d'étranglement des réseaux.

L'ATM Forum a de son côté travaillé sur une intégration des protocoles existants basée sur une approche serveur. Fondé sur les principes de LAN Emulation (cf [ATM97]), ce travail a abouti à la spécification du protocole MPOA (Multi-Protocol Over ATM). Défini dans [RIT98] et [ATM98], ce protocole complexe mais astucieux, développé entre autres par Newbridge, permet de faire du routage sans routeurs. Il sépare les deux fonctions assurées par les routeurs MPLS :

- La commutation des paquets est assurée en hard par les équipements ATM, c'est-à-dire que l'on ne remonte pas au niveau IP.
- La distribution et le calcul de routes sont réalisées par un serveur de routes, qui établit la correspondance entre les adresses réseau et ATM. Ce serveur peut être implémenté sur un équipement dédié ou bien intégré à l'intérieur de routeurs ou de commutateurs.

Un circuit virtuel de niveau 2 (couche ATM) est donc établi pour chaque flux émis d'importance suffisante. Il est alors possible de s'affranchir de la fonctionnalité "routage" et ainsi de profiter pleinement de la qualité de service ATM.

Nous ne nous étendrons pas plus loin dans cette description. Il va de soi que le poids de CISCO à l'échelle mondiale a une forte influence sur l'adoption des normes ; cependant rien n'empêche les acheteurs de routeurs ou de commutateurs d'adopter des technologies concurrentes, sous réserve que ces technologies soient inter-opérables, ce qui est généralement le cas.

4.2 L'adaptation d'ATM à TCP

Le mode de transfert ATM est composé de trois couches :

- la couche physique dont la fonction principale est d'adapter le flux des cellules au support physique ;
- la couche ATM dont la fonction principale est le multiplexage et le démultiplexage des cellules de différentes VPC et VCC utilisant un même conduit de transmission. C'est dans cette couche que sont implémentés la majorité des mécanismes de contrôle de trafic de l'ATM car c'est une couche commune à tous les services ;
- la couche AAL dont la fonction principale est d'adapter le service offert par la couche ATM au service requis par la couche supérieure à la couche AAL. En particulier, c'est la couche AAL qui doit adapter le trafic TCP au mode de transfert ATM.

Dans ce paragraphe, nous nous intéressons uniquement à la couche AAL. Parmi quatre protocoles AAL normalisés, seuls deux ont été recommandés par l'UIT et l'ATM Forum pour le transfert de données sur ATM : les protocoles AAL3/4 et AAL5. Mais le protocole AAL5 s'avère plus populaire et mieux adapté que l'AAL3/4 pour transporter efficacement

les protocoles orientés connexion des couches supérieures. En particulier, si nous utilisons les protocoles TCP/IP au-dessus de mécanismes de contrôle de trafic pris en charge par l'ATM, l'AAL5 permet aux couches supérieures de gérer les connexions et à la couche ATM de garantir un nombre d'erreurs minimal. Dans ce cas, beaucoup de champs et de fonctions du protocole AAL3/4 telles que l'entrelacement des messages, la synchronisation des trames, la gestion des connexions ou l'allocation d'une taille de buffer, s'avèrent inutiles. Pour des services orientés connexion, l'AAL5 réunit ainsi les avantages suivants :

- simplicité des fonctions implémentées et réduction du temps de traitement des trames AAL5 ;
- réduction de l'en-tête des cellules ATM ; cet avantage est important pour les protocoles TCP/IP qui rajoutent déjà une en-tête conséquente lors de la transmission des paquets. L'efficacité de l'AAL5 est alors plus intéressante que l'efficacité du protocole AAL3/4.

L'AAL5 est donc recommandé pour adapter le service de TCP/IP au mode de transfert ATM. Néanmoins, TCP et AAL5 étant deux couches de transport de bout en bout, certains conflits peuvent apparaître entre des fonctions redondantes mais pas forcément complémentaires (telles que la détection d'erreurs ou la prise en compte des informations de congestion et de priorité à la perte). En particulier, nous pourrions discuter de l'intérêt d'une transmission en mode assuré par l'intermédiaire d'un protocole de la sous-couche SSCS (Service Specific Convergence Sublayer) de l'AAL, en sachant que les retransmissions des trames perdues ou erronées sont de toutes façons assurées par le protocole TCP. Cependant, nous ne souhaitons pas aller plus loin dans la description des protocoles de la couche AAL qui ne constituent pas l'objet de notre étude, et nous présentons simplement en figure 4.1 un exemple d'utilisation du protocole AAL5 sur une connexion TCP sur ATM, à laquelle vient s'ajouter le mécanisme de contrôle de flux ABR.

4.3 Procédures de gestion mémoire

Nous venons de présenter les protocoles développés pour la superposition des architectures IP et ATM. Nous nous intéressons à présent à un des problèmes liés à la conversion des paquets IP en cellules ATM. Lorsqu'un réseau ATM est congestionné, il y a risque de perte de cellules au niveau des buffers des goulots d'étranglement. Or, dans le cas de TCP/IP sur ATM, lorsque l'on perd une cellule, on perd également tout le paquet auquel appartenait cette cellule. On n'a donc aucun intérêt à conserver les autres cellules de ce paquet, qui viennent augmenter inutilement la charge sur le réseau. Ce problème de fragmentation des paquets en cellules, générant des cellules "mortes", est amplifié pour de petites tailles de buffers, des paquets TCP de grandes tailles et de grandes fenêtres de congestion. Plusieurs stratégies d'effacement de ces cellules mortes sont proposées.

4.3.1 Partial Packet Discard et Early Packet Discard

En cas de congestion du réseau ATM occasionnelle, Floyd et Romanow ont proposé dans [RF94] le mécanisme Partial Packet Discard (PPD). Lorsqu'une cellule d'un paquet a été

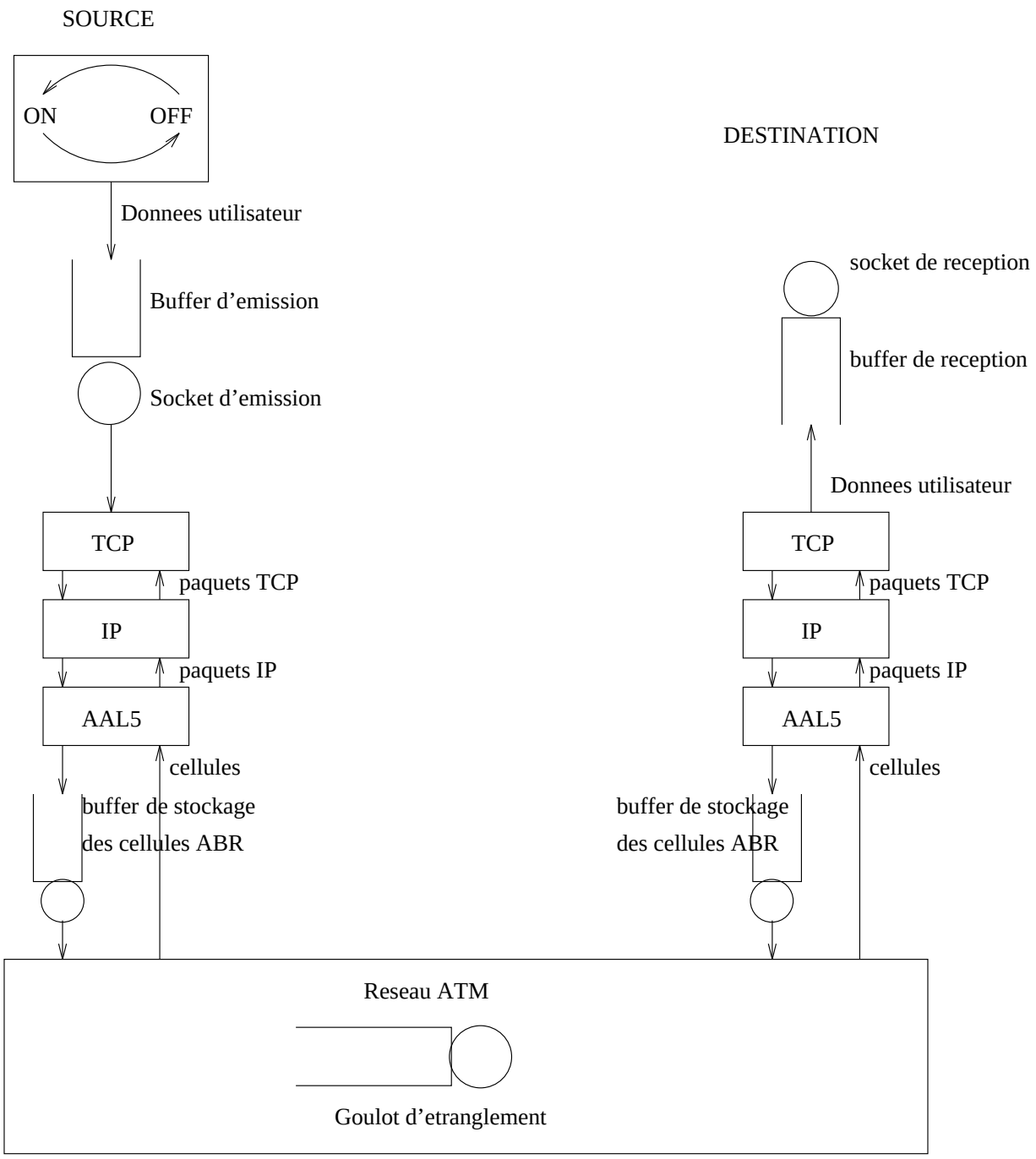


FIG. 4.1 - *Exemple d'une connexion TCP sur ABR*

perdue à cause d'un débordement de buffer, toutes les cellules mortes restantes de ce paquet, arrivées après la cellule perdue, sont jetées, afin de libérer de la place pour les cellules des autres paquets. Toutefois, il n'y a pas de recherche des cellules de ce paquet arrivées avant la cellule perdue. Ces cellules mortes sont donc conservées inutilement dans le buffer et gaspillent la place que l'on pourrait accorder aux cellules d'autres paquets valides.

Floyd et Romanow proposent donc un autre mécanisme plus efficace : Early Packet Discard (EPD). L'astuce consiste à ne pas attendre que le buffer déborde pour jeter un paquet et libérer ainsi de la place pour les autres paquets. Lorsqu'un certain seuil de congestion du buffer est atteint, le commutateur prend la décision de jeter la première cellule d'un paquet et toutes les cellules suivantes de ce même paquet. Ceci permet de maximiser le débit car le commutateur élimine le paquet tout entier (c'est-à-dire toutes les cellules de la trame AAL 5) et libère de la place pour les autres paquets avant le débordement du buffer. Le commutateur prévient donc le phénomène dû à la fragmentation des paquets en cellules et se comporte en fait comme un commutateur de paquets, puisqu'il jette des paquets complets et non plus seulement des cellules. Le fait de ne pas attendre le débordement du buffer pour effacer des cellules permet également de prévenir et réagir à la congestion du commutateur plus tôt ; EPD indique plus tôt à la source de ne pas transmettre immédiatement des données inutiles ou qui ne pourront pas être traitées. Cette technique est particulièrement efficace dans le cas de TCP, puisque ce protocole réagit aux informations fournies par le réseau. Plus le délai de réaction est court, plus la fenêtre TCP est ajustée de manière adéquate. Nous avons donc intérêt, dans le cas de TCP, à privilégier des files d'attente dont la longueur moyenne se rapproche le plus possible de zéro, ce que permettent les mécanismes EPD et, dans une plus grande mesure, RED (voir paragraphe 4.3.2.1).

4.3.2 La stratégie "Drop from Front"

Proposée par Lakshman, Neidhardt et Ott dans [LNO94], cette stratégie consiste à jeter les cellules du buffer du goulot d'étranglement les plus proches d'être servies. Elle présente le double avantage de créer de la place pour les nouvelles cellules arrivant au buffer et permet, dans le cas de TCP, d'envoyer les acquittements dupliqués plus tôt que si l'on jetait les cellules en bout de file. De même que pour EPD, en informant plus tôt la source TCP de la congestion du réseau, la stratégie "Drop from Front" permet à la source de réagir plus rapidement à cette congestion.

De plus, habituellement, les connexions présentant les round trip times les plus courts sont avantagées. Avec la stratégie "Drop from Front", cet avantage est minimisé, ce qui rend le système plus équitable et nécessite des tailles de buffer moindres.

Une application de cette stratégie, la procédure "Partial Frame Drop at the Front" fonctionne de la manière suivante : lorsqu'une cellule au début du buffer (c'est-à-dire juste avant d'être traitée) est jetée, toutes les cellules restantes de la trame sont jetées lorsqu'elles arrivent également au début du buffer. Jusqu'à ce moment, elles occupent l'espace mémoire du buffer (aucune recherche n'est effectuée). Notons que ce mécanisme peut être couplé avec la procédure EPD, et permettre ainsi un rejet préventif des cellules, ce qui accélère encore la réaction de TCP.

4.3.2.1 Le mécanisme RED

La stratégie "Drop from Front" (et toutes les autres procédures de gestion mémoire) peut être combinée au mécanisme proposé par Jacobson et Floyd dans [FJ93] : cette technique "Random Early Detection" (RED) permet, comme EPD, de ne pas attendre que la perte d'un paquet soit inévitable pour jeter ce paquet. En fonction de l'occupation des buffers, chaque cellule dispose d'une probabilité p_a d'être rejetée. Lorsque la première cellule d'un paquet arrive au buffer, la décision de jeter ou de conserver cette cellule est prise selon la probabilité p_a. Combiné à la stratégie "Drop from Front", ce mécanisme permet, lorsque l'on choisit de jeter une cellule, de jeter la trame entière. Ceci autorise d'une part à ne pas attendre la perte inévitable d'une trame et d'autre part, en partant de la première cellule, à libérer davantage de place dans le buffer pour les nouvelles cellules à venir.

RED est plus tolérant qu'EPD concernant les pertes de cellules occasionnelles. Cet avantage peut être utile pour des commutateurs destinés à supporter aussi bien un trafic WAN qu'un trafic LAN. En effet, le seuil EPD, prévu assez élevé pour toutes les connexions, est utilisé afin que le commutateur puisse jeter des paquets de cellules avant d'être forcé par un débordement de buffer à n'effacer que partiellement ces paquets de cellules. RED introduit également un seuil, plus petit, destiné à minimiser le remplissage moyen de la file d'attente. Lorsque ce remplissage indique la persistance d'une congestion, le commutateur demande aux sources d'utiliser leur mécanisme de contrôle de flux de bout en bout afin de réduire la charge sur le réseau. Le mécanisme RED est donc moins radical qu'EPD et apparaît plutôt comme une combinaison des avantages de ce mécanisme et des mécanismes de contrôle déjà prévus au niveau ATM. Nous pouvons trouver une analyse de ce mécanisme dans [GOU99].

La communauté IP, dans le cadre de ses recherches sur la qualité de service IP et de son groupe de travail "Diffserv", propose également, en complément de WFQ (cf 2.2.3), l'algorithme WRED (Weighted Random Early Detection). L'objectif de ce mécanisme est de proposer des priorités à certains flux lors de leur passage dans les files d'attente. Les paquets IP de faible priorité sont ainsi rejetés avec une plus grande probabilité que les paquets de forte priorité. Cet algorithme est implémenté dans les routeurs CISCO. Enfin, le document [KL98] introduit l'algorithme FB-RED (Fair-Buffering Random Early Detection) destiné spécifiquement aux réseaux ATM. FB-RED utilise le mécanisme ABR pour optimiser les caractéristiques de RED dans le cadre d'une superposition équitable de TCP/IP sur ATM.

4.4 Les problèmes de TCP sur ABR

Après avoir présenté les méthodes d'interconnexion du protocole TCP sur les réseaux ATM, il importe maintenant de revenir à la notion de contrôle de trafic. A ce titre, l'intégration de TCP pose de nouveaux problèmes. En effet, le protocole de transport TCP propose un mécanisme de contrôle largement utilisé sur l'Internet. Mais, comme nous l'avons remarqué au paragraphe 1.2.4, ce mécanisme ne convient pas à la qualité de service requise dans l'ATM ou est inadapté à la vitesse de transmission des réseaux à hauts débits. Aussi propose-t-on parfois de coupler ce protocole à la capacité de transfert ABR, dont le mécanisme de contrôle de trafic vient suppléer le protocole TCP. Néanmoins, ce couplage peut être néfaste si les deux mécanismes de contrôle de flux agissent dans des directions

opposées. Nous recensons brièvement dans cette section les inconvénients majeurs de l'interaction de TCP sur ABR. Le but de cet ouvrage est précisément d'analyser et de résoudre ces problèmes ; en préliminaire de la partie II, nous essayons donc de prévoir certaines caractéristiques de l'interfonctionnement de TCP sur ABR en présentant les conclusions émises par plusieurs études.

4.4.1 Les besoins en buffer

Les mesures de performances, effectuées par Raj Jain dans [KJF+96] sur le commutateur ERICA, montrent que l'utilisation de buffers de tailles limitées peut entraîner une chute du débit importante, même si le taux de perte des cellules (Cell Loss Ratio) reste faible. Les expériences réalisées ont conduit aux conclusions suivantes :

- Une mémoire tampon supérieure ou égale à

  $[(3 \cdot RTT + c \times \text{feedback delay}) \times \text{bande passante du lien}]$

  garantit aucune perte pour TCP (c est une constante du commutateur et de ses paramètres).

- La même quantité de mémoire tampon peut supporter un grand nombre de sources ABR ou TCP.

- Dans beaucoup de cas, ($RTT \times$ bande passante du lien) suffit.

- Les politiques de destruction de cellules augmentent le débit effectif mais sont moins efficaces qu'un bon algorithme de commutateur.

- Une bonne approximation de la longueur maximale de la file d'attente du commutateur est :

  $$Q_{max} = (a \cdot RTT + b \cdot AIL + c \times \text{feeback delay}) \times \text{bande passante du lien}$$

  où AIL est l'Average Interval Length décrit dans le paragraphe 3.2.2.5.

  Si l'AIL est supérieur au feedback delay, son effet est alors prépondérant. D'autre part, si le feedback delay est petit, les files d'attente restent petites, même dans le cas de grands RTT, car le commutateur peut limiter le débit des sources avant qu'elles ne surchargent le réseau.

Cette étude permet de donner une approximation grossière de la taille maximale que peuvent atteindre les files d'attente lors d'un fonctionnement normal. La valeur fournie ne permet pas une optimisation des tailles de mémoires mais peut être destinée à limiter les pertes sur un réseau.

4.4.1.1 Les trafics prioritaires sur ABR

L'étude de Raj Jain n'envisage pas la perturbation du trafic ABR par les trafics de plus haute priorité tels que CBR et VBR, qui introduisent une certaine variance sur la charge

et la capacité du trafic ABR. Dans le cas des commutateurs explicite, la version ERICA+ est particulièrement intéressante dans ce cas et permet de s'adapter plus rapidement à la capacité disponible.

Dans le cas de commutateurs binaires, les performances de TCP dépendent beaucoup du trafic CBR ou VBR avec lesquels le trafic TCP partage les canaux de transmission. En particulier si ce trafic a lieu par rafale, le partage équitable de la bande passante entre les connexions revêt une importance non négligeable et exige un choix attentionné des paramètres ABR, comme le montre l'étude [MBLCM96] réalisée par le Politecnico di Torino. Ce paramétrage est précisément l'objet de l'analyse menée dans les parties II et III.

4.4.2 ABR contre UBR

Les expériences réalisées dans [MLB97] comparent le mécanisme ABR avec le mécanisme UBR. Le premier rajoute un mécanisme de contrôle de flux en-dessous du protocole TCP, tandis que le second laisse TCP gérer les congestions et se charge d'envoyer toutes les cellules qu'il peut, sans savoir si elles traverseront ou non le réseau ATM. Le document [MAN97a] discerne les problèmes suivants pour ABR :

- Les cellules RM engendrent une surcharge du réseau.
- ABR introduit une nouvelle file d'attente aux extrémités du réseau ATM, qui peut augmenter le délai de bout en bout. Lorsque la charge du réseau est importante, le délai est dominé par le temps d'attente dans cette nouvelle file. En effet, la boucle de contrôle ABR est prévue pour ralentir le débit de la source en cas de congestion du réseau ATM. Mais ABR n'est pas prévu pour signaler au réel émetteur du trafic qu'il doit réduire son débit. Ceci signifie qu'ABR peut éviter la congestion dans le réseau ATM, mais non pas pour le trafic TCP de bout en bout. En fait, ABR ne fait que délocaliser la congestion et les pertes en dehors de la boucle de contrôle ABR. Nous présentons en détail ce problème dans la partie II.
- La complexité d'ABR fait que de nombreux paramètres doivent être réglés correctement pour que TCP puisse fonctionner.

Néanmoins, comme nous l'avons déjà souligné aux paragraphes 3.2 et 3.3 et comme le confirment l'étude précédente et l'analyse [SK98], l'interaction entre TCP et ABR apparaît meilleure que l'interaction entre TCP et UBR. Les avantages d'ABR vis-à-vis d'UBR ressortent particulièrement dans le cas de fortes charges, lorsque par exemple la taille des buffers dans le réseau ATM devient petite comparée à la taille des buffers aux extrémités du réseau. En effet, les connexions UBR génèrent davantage de pertes si les buffers du réseau ne sont pas adaptés pour recevoir tous les paquets envoyés par les sources (cela se produit dès que le nombre de connexions augmente). De plus, le mécanisme ABR permet de mieux contrôler l'équité entre les connexions. L'étude [OA97] de T. J. Ott démontre en conclusion que l'ABR permet de répartir équitablement la bande passante entre les connexions et de garantir un remplissage plus ou moins constant du goulot d'étranglement, c'est-à-dire de minimiser le délai et les pertes dans le réseau. Ces avantages sont précisément les caractéristiques attendues d'un mécanisme de contrôle de trafic et d'optimisation des ressources. Toutefois, T. J. Ott ne semble par convaincu que

les performances d'ABR dans le transfert de trafics TCP soient suffisantes pour motiver le choix de cette capacité complexe au détriment de la capacité UBR.

Cette première partie propose une exploration générale des différents mécanismes de contrôle de trafic essentiellement rencontrés dans l'ATM ; cette recherche est complétée par le rappel du protocole TCP. Nous avons effectué une description de la mécanique élémentaire de TCP, puis une présentation des principales fonctions, mécanismes et services développés dans l'ATM ; le cœur de cette présentation concerne la capacité de transfert ABR.

Ce dernier chapitre aborde les conséquences liées à l'intégration des protocoles TCP/IP sur ATM. Les inconvénients ou points d'arrêt majeurs présentés dans cette section sont une ouverture vers l'analyse du protocole TCP couplé avec la capacité de transfert ABR. Nous allons voir dans ce qui suit comment le protocole TCP peut être interfacé avec ABR de manière à ce que leur interaction soit bénéfique à l'ensemble du trafic.

Deuxième partie

Analyse d'une connexion TCP sur ABR

Chapitre 1

Introduction

Le principal intérêt de la capacité de transfert ABR pour un opérateur réside dans la possibilité d'assurer un multiplexage statistique de la bande passante entre les utilisateurs, tout en protégeant chaque utilisateur des autres. Cette protection consiste à garantir à chacun une part équitable de la bande passante, et à éviter autant que possible les pertes de cellules sur les supports partagés. Tandis que TCP utilise les pertes de paquets pour réguler son débit, l'objectif d'ABR est de protéger le réseau ATM et ses utilisateurs en délocalisant les pertes de cellules sur des supports non partagés.

Afin d'atteindre cet objectif, nous abordons dans cette partie l'analyse difficile du comportement de TCP sur ABR et du paramétrage d'un réseau ABR et de ses connexions. Nous considérons que l'opérateur gère les unités de contrôle ABR de la connexion ATM. Cela lui permet d'assumer le contrôle de la congestion dans son réseau en réduisant lui-même le débit lors des indications de congestion générées par les équipements traversés. L'autre choix consistant à faire varier les paramètres du policing et à demander à l'utilisateur de réguler son débit peut aboutir à des performances très mauvaises, avec des rejets partiels de trames, des problèmes d'ajustement au temps de réaction des sources et la contrainte pour les clients d'être équipés en cartes ABR. C'est pourquoi nous considérons qu'il est plus pertinent de réaliser cette étude d'un point de vue opérateur.

1.1 Simulations et analyses

Le service ABR a fait l'objet de nombreuses études par simulation dans des cas de configurations particuliers. Quelques études analytiques du service ABR isolé ont également été réalisées, dont les plus intéressantes sont détaillées dans [RIT96], [RIT97b] et [AHA99]. Cependant, ce service ayant été conçu pour le transport de données, un des problèmes majeurs que les opérateurs seront amenés à rencontrer lors de son implémentation proviendra vraisemblablement des interactions avec d'autres protocoles de transfert de données déjà existants, dans le monde Internet notamment, et utilisant un mécanisme de contrôle de congestion différent de celui utilisé par le service ABR. Typiquement, le protocole de transfert de données le plus utilisé actuellement est le protocole TCP.

Nous analysons donc dans cette partie les performances de la catégorie de service ABR dans le transport de cellules ATM résultant de la segmentation de paquets TCP. Le comportement du protocole TCP sur ABR a déjà fait l'objet d'études par simulation, menées

notamment par Raj Jain dans [KJF+96] ou l'EPFL dans [MLB97]. Cependant, comme tout exemple numérique, ces simulations analysent des configurations particulières et il paraît difficile de généraliser leurs résultats. Néanmoins, elles permettent d'avoir un éventail assez large des problèmes rencontrés lorsque TCP et ABR interagissent ; certains problèmes sont décrits dans la partie I. Parallèlement, si les deux mécanismes TCP et ABR ont déjà été analysés chacun séparément, très peu d'études analytiques de modèles de connexions TCP sur ABR ont été abordées : les principales sont exposées dans les documents [AH98] et [SK98]. Le but de cette partie est précisément de proposer un nouveau modèle analytique d'une connexion TCP sur ABR, en présence de flux exogène. Nous verrons que cette étude permet non seulement de calculer des formules d'évaluation de performances explicites pour notre modèle, mais également de comprendre et d'expliquer le comportement observé lors de simulations et éventuellement de déceler de nouveaux phénomènes, inconnus jusqu'alors. Cette étude apparaît donc comme un excellent point de départ pour formuler des recommandations pertinentes quant au choix des paramètres lors de l'implémentation du service ABR sur un réseau existant qui utilise déjà le protocole TCP pour le transfert de ses données.

1.2 Problèmes et motivations

Nous avons mentionné en première partie les caractéristiques des protocoles TCP et ABR, ainsi que leurs avantages et leur complémentarité. Nous avons également indiqué certains inconvénients dûs aux interactions entre les deux mécanismes. En effet, TCP et ABR poursuivent le même but, à savoir le contrôle de trafic, mais de deux manières différentes. Ces différences peuvent être source de complémentarité mais également source d'antagonismes. Nous reprenons dans ce paragraphe les problèmes prépondérants.

Premièrement, grâce à son mécanisme de fenêtre dynamique, le protocole TCP est relativement simple et s'adapte à un grand nombre de configurations de réseaux sans nécessiter de réglage particulier. A l'opposé, à cause de la multiplicité de ses paramètres, la capacité de transfert ABR est plus compliquée et difficile à configurer : les performances dépendent fortement du choix des paramètres et des mécanismes de congestion utilisés par les divers éléments du réseau. L'un des objectifs de cette étude est précisément d'aider l'opérateur à choisir correctement les paramètres d'ABR. La robustesse de ce paramétrage permettra de garantir des performances optimales pour de multiples configurations de réseaux.

Deuxièmement, TCP et ABR utilisent deux mécanismes de contrôle de congestion différents : tandis que TCP réduit son débit en fonction des pertes de paquets, ABR utilise les informations contenues dans les cellules RM pour contrôler l'ACR. De ce fait, la source ABR ne doit pas attendre la perte d'une cellule pour réduire son débit.

Enfin, en cas de congestion, la boucle de contrôle ABR est responsable de la limitation du débit de la source ABR au niveau de l'unité de contrôle ABR. Mais dans le cas d'une connexion TCP communiquant au-dessus d'une connexion virtuelle (CV) ATM-ABR, la boucle de contrôle ABR n'est pas capable d'indiquer à la réelle origine du trafic l'arrivée d'une congestion sur le réseau et le débit auquel cette source devrait émettre ses paquets. En effet, bien que la source ABR soit consciente de l'état de congestion du réseau, elle ne peut pas communiquer avec la source TCP, et par conséquent ne peut pas l'informer ni

l'obliger à réduire son débit[1]. La source TCP envoie alors des paquets jusqu'à la perte. Cette impossibilité de communiquer entre TCP et ABR implique que la congestion peut être évitée dans la partie ATM du réseau, grâce à l'unité de contrôle ABR, mais pas sur le trafic TCP de bout en bout. La congestion est donc simplement délocalisée : en cas de congestion, les cellules ne sont plus stockées dans le réseau au niveau du goulot d'étranglement (puisque la source ABR réduit son débit de manière à ne pas saturer le réseau) mais elles sont stockées au niveau de la source, dans le buffer de l'unité de contrôle ABR. La source TCP continue d'envoyer des paquets à la source ABR qui ne peut plus les envoyer dans le réseau puisque son débit ACR est réduit ; l'unité de contrôle ABR stocke alors ces paquets dans le buffer d'accès au réseau, que nous nommons ici "ingress buffer". Cette délocalisation des congestions présente l'intérêt d'éviter que les pertes de paquets TCP se produisent sur les supports partagés du réseau.

L'objectif de cette étude est d'analyser, en terme de performances, les interactions entre les mécanismes de contrôle de flux TCP et ABR. Dans cette partie, nous présentons une approche analytique pour décrire l'évolution dynamique en fonction du temps du débit ACR et du remplissage du goulot d'étranglement, c'est-à-dire du nœud présentant le débit le plus faible sur le réseau du point de vue de notre connexion TCP sur ABR. Nous décrivons également l'évolution de la fenêtre TCP, donc du débit d'entrée des paquets TCP dans le réseau, ainsi que l'évolution de la file d'attente de l'ingress buffer, c'est-à-dire du buffer d'accès dans lequel sont stockées les cellules ATM avant d'être envoyées dans le réseau par l'unité de contrôle ABR. Nous étudions les corrélations entre ces variables et indiquons les différents états que parcourt notre système durant un cycle de contrôle TCP (la définition d'un cycle TCP est donnée dans la partie I, définition 1.2.1). Nous cherchons à comprendre comment fonctionne notre connexion et à observer les phénomènes qui influencent ses performances.

1.3 Plan de l'analyse

Dans cette partie, nous adaptons les analyses fluides de [RIT96] et [ABN+95] pour étudier l'impact d'une connexion TCP communiquant sur un VC ATM-ABR, combiné avec un trafic exogène indépendant de notre connexion mais qui l'influence en augmentant le flux d'arrivée au goulot d'étranglement. Nous présentons notre modèle dans une première section, puis nous rappelons les principaux résultats établis et qui serviront de base à notre étude. Ensuite, nous résolvons notre modèle phase par phase. Des résultats explicites sont fournis : nous calculons pour chaque état de notre système les expressions en fonction du temps du débit ACR, de la fenêtre de congestion W, et de la longueur des files d'attente mentionnées plus haut. Nous résumons également l'ensemble des états par lesquels peut transiter notre système à l'aide de diagrammes d'états présentés en annexe A.

1. S. Floyd a proposé dans [FLO94] d'utiliser une "Notification Explicite de Congestion" pour informer la source TCP de la congestion du réseau avant d'atteindre la perte d'un paquet. Ce mécanisme n'a pas encore été implémenté.

Chapitre 2

Analyse d'une connexion TCP sur ABR

2.1 Modèle et notations

Nous modélisons une connexion TCP communiquant sur un réseau ATM à travers un CV ABR et partageant le goulot d'étranglement du réseau avec un trafic exogène fluide supposé constant, égal à E paquets TCP par seconde. Le modèle proposé est composé de deux files d'attente : une file représentant l'ingress buffer de la source ABR et une autre file représentant le buffer du goulot d'étranglement du réseau ATM. Nous avons choisi de modéliser le réseau par une seule file d'attente[1] que nous appelons le "goulot d'étranglement", i.e le nœud présentant le débit de sortie le plus faible, compte tenu des autres flux. Ce goulot est destiné à suivre l'état de congestion sur le réseau ATM, tandis que l'ingress buffer doit permettre de surveiller la délocalisation de cette congestion au niveau de la source ABR. Ces deux files d'attente sont servies de manière FIFO (First In, First Out). Le schéma de notre modèle est présenté en figure 2.1. Les notations utilisées dans notre analyse sont récapitulées dans le paragraphe 2.1.2.

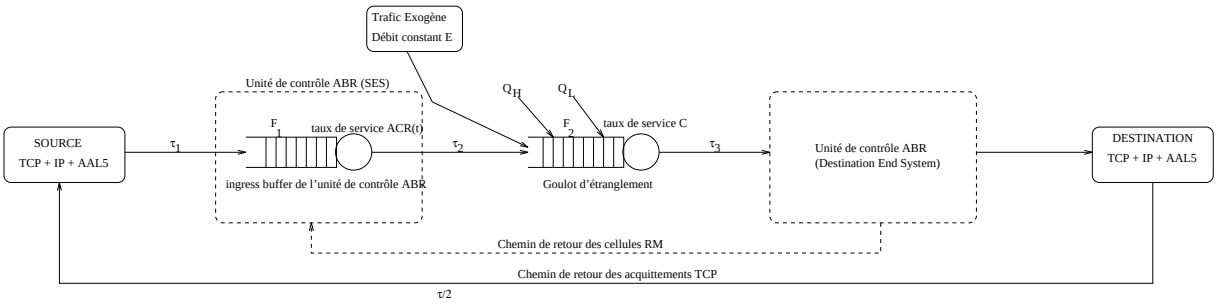


FIG. 2.1 - *Modèle de notre connexion TCP sur ABR*

La source TCP est une source infinie : il y a toujours un paquet à émettre. De plus, nous considérons uniquement les phases de transfert d'informations et n'analysons pas le comportement à l'ouverture ou à la fermeture de la connexion.

1. Les études [BA99] ou [BES99] proposent de modéliser les liens traversés par plusieurs files, ce qui permet de tenir compte de plusieurs sources de saturations.

Le type de marquage des cellules RM utilisé est le "Relative Rate Marking" défini au paragraphe 3.2.2.5. Le goulot d'étranglement correspond donc à un commutateur binaire, où seul le bit CI des cellules RM est marqué. Le bit NI n'est pas utilisé pour les raisons exposées au paragraphe 3.2.2.5.

Le schéma de détection de congestion utilisé au niveau de la file d'attente du goulot d'étranglement est basé sur la longueur de cette file. Ce schéma introduit deux seuils au niveau du goulot d'étranglement : un seuil supérieur Q_H et un seuil inférieur Q_L. Lorsque l'occupation du buffer du goulot d'étranglement dépasse Q_H, une indication de congestion est retournée à la source ABR par l'intermédiaire des cellules RM. Cette indication est matérialisée par la mise à 1 du bit CI des cellules RM. A la réception de ces cellules RM, la source ABR commence à faire décroître l'ACR tant qu'on ne l'a pas informée que l'occupation du buffer est redescendue en-dessous de Q_L. L'occupation du buffer du goulot d'étranglement oscille donc entre Q_L et Q_H. Ce schéma de détection de congestion simple est utilisé sur la plupart des commutateurs ATM. Il est très facile de l'implémenter. C'est pour cela que nous pensons qu'il est plus judicieux de s'intéresser au "Relative Rate Marking" qu'à "l'Explicit Rate Marking": de par sa simplicité d'utilisation et d'implémentation, le "Relative Rate Marking" sera vraisemblablement plus répandu que "l'Explicit Rate Marking". Cette technique de commutation simple et rapide correspond davantage à l'esprit d'ATM, qui consiste à repousser les fonctions complexes à la périphérie du réseau. De plus, la complexité du paramétrage importe surtout pour le "Relative Rate Marking". Nous nous plaçons donc dans une situation de "pire cas" : l'utilisation de mécanismes de détection de congestion plus évolués tels que l'"Explicit Rate Marking" permettra d'espérer des performances encore meilleures que celles que nous obtenons ici.

2.1.1 Autres hypothèses de modélisation

Afin de simplifier l'analyse du comportement de TCP, les débits et remplissages des files d'attente sont supposés fluides et exprimés en paquets TCP par seconde pour les débits, et en paquets pour l'occupation des files. A ce sujet, il importe de préciser que la source TCP envoie des paquets, tandis que la source ABR envoie des cellules ATM. Un paquet TCP contient MTU (Maximum Transmission Unit) octets, et une cellule ATM contient 53 octets. Afin d'exprimer toutes les grandeurs avec l'unité "paquets TCP", il est nécessaire d'établir une correspondance entre un paquet TCP et une cellule ATM. Cette correspondance est obtenue à l'aide du facteur n_1, qui représente le nombre de cellules ATM correspondant à un paquet TCP segmenté. Cette variable n_1 dépend grossièrement de la taille d'un paquet TCP et de la longueur du champ d'information des cellules ATM (48 octets pour l'AAL5). Une approximation de n_1 nous conduit donc à fixer sa valeur à $\frac{MTU}{48}$. Faire varier n_1 permet d'observer l'influence de la taille du MTU sur les performances d'une connexion TCP sur ABR.

Le taux de service de l'ingress buffer de l'unité de contrôle ABR, égal au débit de transmission autorisé à la source ABR soit ACR cellules ATM par seconde, est donc équivalent à $\frac{ACR}{n_1}$ paquets TCP par seconde. Le taux de service du goulot d'étranglement, supposé constant, vaut quant à lui C paquets TCP par seconde.

D'autre part, pour éviter de redémarrer une connexion ABR au débit ICR à cause d'un retard des paquets ou d'une perte, ce qui aurait pour conséquence de ne pas tenir compte du passé pendant lequel le réseau a cherché à s'adapter à la bande passante disponible,

nous prenons pour hypothèse :

$$ADTF \geq RTO$$

D'une manière générale, il est conseillé de choisir ADTF égal à la valeur maximale $10,23$ secondes. Ainsi l'ACR ne reviendra pas à la valeur ICR, sauf dans le cas d'un arrêt des transmissions.

De son côté, le temps de propagation aller-retour sur la connexion, excluant le temps d'attente et le temps de service dans les files, vaut τ. Le temps de séjour d'un paquet dans un système vide est donc égal à $T = \tau + \frac{n_1}{ACR} + \frac{1}{C}$. Nous choisissons ici de nous placer dans un contexte de réseau WAN pour les raisons qui sont décrites dans le paragraphe 3.2 de la partie I. Sous cette hypothèse, les termes $\frac{n_1}{ACR}$ et $\frac{1}{C}$ peuvent être négligés. D'autres délais, comme le délai induit par la paquetisation des cellules ATM, sont également négligés dans l'expression du temps de séjour. Nous remarquons toutefois que si nous sortons du contexte WAN, le délai de paquetisation n'est pas forcément négligeable, particulièrement lorsqu'ACR est petit puisqu'il faut attendre que toutes les cellules ATM soient revenues à la source avant de pouvoir réassembler le paquet IP.

Par ailleurs, nous savons que le trafic ABR est un trafic non prioritaire par rapport aux flux CBR ou VBR qui peuvent arriver au goulot d'étranglement. La capacité C correspond donc à la capacité restante pour le trafic ABR, après avoir écoulé les trafics prioritaires CBR et VBR. De plus, nous ne nous intéressons qu'au trafic aller et nous ne considèrons pas la possibilité d'une congestion sur le chemin de retour emprunté uniquement par les acquittements TCP et les cellules RM.

Enfin, notre étude analytique ne cherche à dimensionner que les paramètres prépondérants de l'ABR, c'est-à-dire ceux qui ont une influence notoire sur le comportement de la connexion TCP sur ABR en régime stationnaire : RIF, RDF, Q_L et Q_H. Les paramètres "mineurs" font l'objet d'une étude par simulation, décrite dans la partie III.

2.1.1.1 Légitimité de l'analyse fluide

L'analyse fluide, pourtant régulièrement utilisée lors de précédentes études, ne donne pas un modèle aussi exact que l'analyse discrète, particulièrement dans le cas du protocole TCP qui comptabilise les paquets un par un pour faire augmenter sa fenêtre. Une analyse fluide serait plus exacte si le traitement des données dans la réalité se faisait bit à bit. Mais ce traitement s'effectue paquet par paquet. Ceci signifie que l'écoulement des paquets n'est pas un flux continu fonction du temps, mais plutôt un échantillonage de paquets le long du lien. C'est pourquoi il est plus juste d'étudier un modèle discret plutôt qu'un modèle fluide.

Toutefois, dans le cas d'un réseau ATM, le découpage des paquets en cellules plus petites rend l'analyse fluide plus pertinente qu'elle ne l'est dans le cas d'une étude en mode paquets. En outre, l'analyse fluide présente l'avantage de simplifier les calculs par rapport à l'analyse discrète : comme nous pouvons le voir dans le document [BC96], l'analyse discrète d'un modèle à une seule file est lourde et complexe, et cette complexité est amplifiée dans le cas de notre modèle de connexion TCP sur ABR à 2 files d'attente. L'avantage d'une modélisation fluide est de permettre de raisonner sur des fonctions continues, et ainsi de

représenter l'évolution des variables d'état sous la forme d'équations différentielles faciles à résoudre.

De plus, bien que cette approximation puisse paraître grossière, l'objectif de cette étude n'est pas d'obtenir un modèle analytique parfaitement exact, mais plutôt de comprendre et d'expliquer les phénomènes et mécanismes qui entrent en jeu, ainsi que les interactions entre TCP et ABR. Nous souhaitons découvrir les types de comportements que peut révéler une connexion TCP sur ABR avec flux exogène et avoir une bonne vision d'ensemble de l'évolution en fonction du temps des différentes grandeurs de cette connexion. L'analyse de cette "connexion type" permet d'exploiter et de comprendre l'enchaînement des comportements dans d'autres types de réseaux plus complexes, réels ou simulés, et de résoudre les problèmes qui surviennent par la connaissance de leur cause. D'autre part, comme nous le verrons lors de la validation de cette analyse dans la partie III, les résultats obtenus analytiquement restent très proches de ceux obtenus par simulation et décrivent assez précisément l'évolution du système. Les hypothèses proposées dans cette étude permettent donc de simplifier légitimement une analyse qui, comme nous allons le découvrir, est déjà très complexe.

2.1.2 Notations

Les notations utilisées dans ce document sont indiquées dans le tableau 2.1.

W	taille de la fenêtre de congestion de la source TCP au temps t (en paquets)
ACR	allowed cell rate, débit d'émission des cellules de la source ABR, au temps t
thp_{in}	débit des paquets TCP à la sortie de la source TCP contrôlée, c'est-à-dire à l'entrée du réseau ATM
thp^r_{in}	débit imposé par le réseau et le remplissage des files d'attente, lorsque la fenêtre W dispose de paquets en nombre suffisant pour maintenir ce débit
thp_{out}	débit des paquets TCP, originaires de la source TCP contrôlée, à leur sortie de la file d'attente du goulot d'étranglement, c'est-à-dire à leur arrivée à destination
E_{out}	débit du flux exogène en sortie de goulot d'étranglement
F_1	file d'attente de l'ingress buffer de la source ABR
F_2	file d'attente du goulot d'étranglement
Q_1	nombre de paquets dans le buffer de la file 1
Q_2	nombre de paquets dans le buffer de la file 2
W_{th}	seuil de la phase de slow start
	page suivante

W_{max}	taille maximale de la fenêtre W, calculée au paragraphe 5.5
a	nombre de paquets acquittés par la réception d'un acquittement
β	équivalent de la pente de dW/dt en phase de slow-start (cf paragraphe 2.2.2)
C	taux de service constant du goulot d'étranglement (en paquets/s)
E	débit constant du flux exogène arrivant au buffer du goulot d'étranglement (en paquets/s)
n_1	nombre de cellules correspondant à un paquet TCP segmenté
PCR	débit crête de la source ABR
ICR	débit initial de la source ABR
$ADTF$	temps autorisé entre deux cellules RM avant que l'ACR ne redémarre depuis la valeur ICR
RIF	Rate Increase Factor de l'unité de contrôle ABR
RDF	Rate Decrease Factor de l'unité de contrôle ABR
N_{rm}	nombre de cellules de données séparant 2 cellules RM
τ	délai de propagation aller-retour des paquets lorsque les files d'attente sont vides, sans inclure les temps de service. $\tau = 2(\tau_1+\tau_2+\tau_3)$
τ_1	délai de propagation de la source jusqu'à l'ingress buffer de l'unité de contrôle ABR
τ_2	délai de propagation de l'unité de contrôle ABR jusqu'au buffer du goulot d'étranglement
τ_3	délai de propagation du goulot d'étranglement jusqu'à la destination TCP
T	temps de séjour d'un paquet dans un système où les files d'attente sont vides. $T = \tau + \frac{n_1}{ACR} + \frac{1}{C}$
T'	temps de séjour d'une cellule RM lorsque le goulot d'étranglement est vide. $T' = 2(\tau_2 + \tau_3) + \frac{1}{n_1 C}$
B_1	taille (en paquets) de l'ingress buffer
B_2	taille (en paquets) du goulot d'étranglement
Q_H	seuil supérieur haut du goulot d'étranglement
Q_L	seuil supérieur bas du goulot d'étranglement

TAB. 2.1 - *Notations adoptées pour l'étude générale*

2.2 Fondements de l'étude : résultats établis

Dans cette section, nous reprenons l'analyse fluide de [RIT96] réalisée sur un modèle de connexion ABR seule, et celle de [NAB+97] réalisée sur une connexion TCP avec flux exogène.

2.2.1 Analyse de l'ABR

Dans l'article [RIT96], en modélisant le flux de cellules provenant des sources ABR par un fluide (cf figure 2.2), les évolutions de l'ACR et de la longueur de la file d'attente

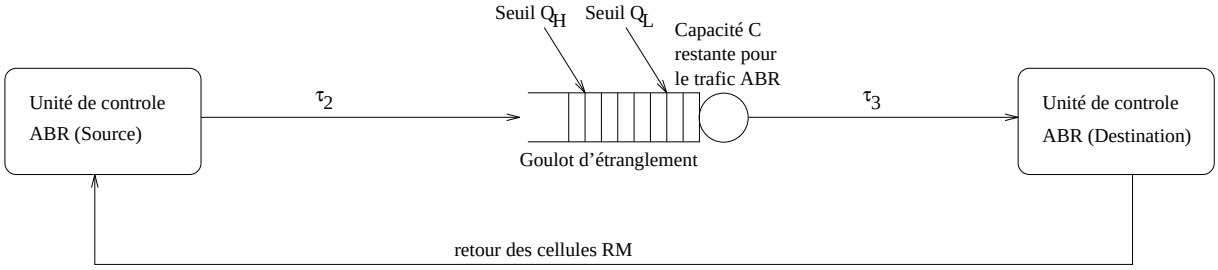


FIG. 2.2 - *Modèle de connexion ABR avec goulot d'étranglement*

du goulot d'étranglement sont calculées en résolvant des équations différentielles. Ces équations différentielles sont obtenues en divisant le contrôle ABR en trois phases :

- Phase 1 : la file d'attente du goulot d'étranglement est non vide et le nœud correspondant n'est pas congestionné. L'ACR est donc augmenté de $RIF \times PCR$ à chaque réception d'une cellule RM, tant que l'ACR n'a pas atteint PCR. Comme le buffer est non vide, ces arrivées sont régulées par un débit constant, égal à $\frac{n_1 C}{N_{rm}}$, $n_1 C$ étant la capacité restante au trafic ABR lors de son passage dans le goulot d'étranglement. La croissance de l'ACR durant cette phase est alors exprimée par l'équation différentielle suivante :

$$\frac{dACR(t)}{dt} = \frac{n_1 RIF \cdot PCR \cdot C}{N_{rm}} \tag{2.2.1}$$

  L'ACR croît de manière linéaire.

- Phase 2 : la file d'attente du goulot d'étranglement est non vide, mais le nœud est congestionné. L'ACR décroît à présent, de $RDF \times ACR$ à chaque retour d'une cellule RM, tant que l'ACR n'est pas descendu en-dessous de MCR. Le taux d'arrivée des cellules RM est toujours égal à $\frac{n_1 C}{N_{rm}}$ puisque le goulot d'étranglement est saturé. Nous obtenons donc l'équation différentielle :

$$\frac{dACR(t)}{dt} = -\frac{n_1 RDF \cdot ACR(t) \cdot C}{N_{rm}} \tag{2.2.2}$$

  L'ACR décroît de manière exponentielle.

- Phase 3 : la file d'attente du goulot d'étranglement est vide et le nœud correspondant à ce goulot d'étranglement n'est pas congestionné. L'ACR croît de $RIF \times PCR$ à chaque réception d'une cellule RM, tant que l'ACR n'a pas dépassé PCR. Mais comme le buffer est vide, le taux d'arrivée des cellules RM dépend du débit auquel elles ont été envoyées un aller-retour plus tôt, soit $\frac{ACR(t-T')}{N_{rm}}$. L'ACR est donc déterminé à partir de cette équation différentielle à retard :

$$\frac{dACR(t)}{dt} = \frac{RIF \cdot PCR \cdot ACR(t-T')}{N_{rm}} \tag{2.2.3}$$

L'ACR croît de manière exponentielle.

Remarquons que nous ne considèrons pas dans cette étude l'influence d'un flux exogène sur le CV ABR : nous soustrayons simplement de la capacité totale du lien le taux d'arrivée du trafic prioritaire (tel que le flux CBR ou VBR) et nous considèrons que la capacité restante (égale à C) représente la capacité allouée au trafic ABR.

Comme pour notre modèle, l'état de congestion est déterminé dans ce modèle par le dépassement d'un seuil supérieur Q_H, au niveau de l'occupation du goulot d'étranglement. Le retour à la normale est décrété lorsque l'occupation du buffer redescend en-dessous d'un seuil Q_L.

2.2.2 Analyse de TCP : modèle fluide à une file

Les rapports [BCA+96] et [COS96] décrivent de leur côté l'analyse fluide d'une connexion TCP partageant un goulot d'étranglement avec un flux exogène (cf figure 2.3).

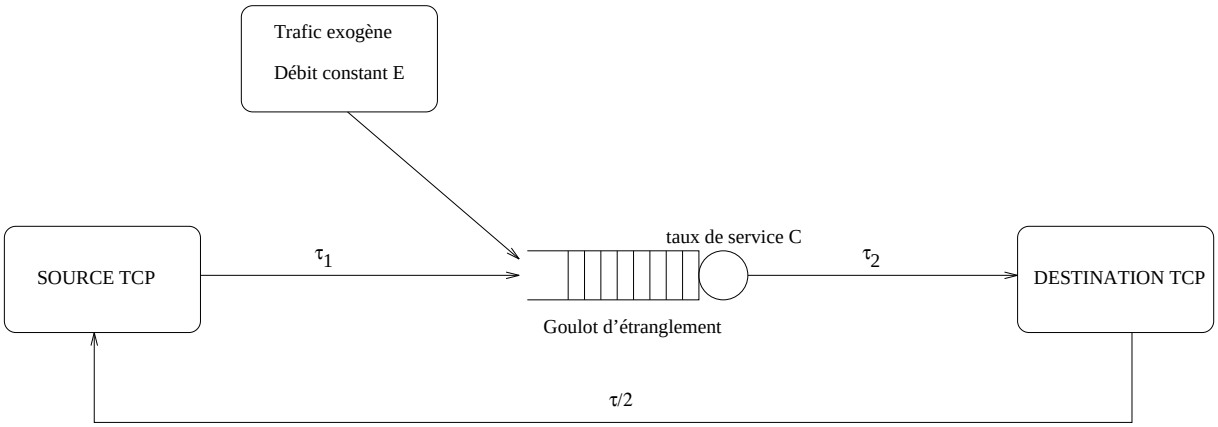


FIG. 2.3 - *Modèle d'une connexion TCP avec flux exogène et une seule file d'attente*

Ils montrent l'existence de trois phases d'évolution de la fenêtre de congestion W en fonction du temps :

- une croissance exponentielle de la taille de la fenêtre tant que cette fenêtre est inférieure à $\min\{W_{th}, T(C-E)\}$, c'est-à-dire tant que nous sommes en mode "slow start" et que la file d'attente est vide ;
- une croissance linéaire lorsque la taille de la fenêtre est telle que :

$$\min\{W_{th}, T(C-E)\} \leq W \leq \max\{W_{th}, T(C-E)\}$$

- une croissance en racine dès que la fenêtre est supérieure à $\max\{W_{th}, T(C-E)\}$, c'est-à-dire dès que nous sommes en mode "congestion avoidance" et que la file d'attente est non vide.

Pour prouver ces phases d'évolutions de W, nous observons que la vitesse de croissance de la fenêtre est telle que :

$$\frac{dW}{dt} = \frac{dW}{dack} \cdot \frac{dack(t)}{dt} = \frac{dW}{dack} \cdot \frac{thp_{out}(t)}{a}$$

où $ack(t)$ est le nombre total d'acquittements reçus à la date t. Il y a émission d'un acquittement pour a paquets TCP parvenus à destination.

Or, par définition des modes de croissance de la fenêtre, à savoir les modes "slow start" (si $W \leq W_{th}$) et "congestion avoidance[2]" (si $W > W_{th}$), nous avons :

$$\frac{dW}{dack} = \begin{cases} 1 & \text{si } W \leq W_{th} \\ W^{-1} & \text{si } W > W_{th} \end{cases} \tag{2.2.4}$$

D'où

$$\boxed{\frac{dW(t)}{dt} = \begin{cases} \frac{thp_{out}(t)}{a} & \text{si } W(t) \leq W_{th} \\ \frac{thp_{out}(t)}{a \cdot W} & \text{si } W(t) > W_{th} \end{cases}} \tag{2.2.5}$$

Par ailleurs, tant que la file d'attente du goulot d'étranglement est vide, nous avons :

$$thp_{in}(t) = thp_{out}(t) = \frac{W(t)}{T} \tag{2.2.6}$$

Lorsque le buffer est non vide, les débits de sortie du flux TCP et du flux exogène hors de la file d'attente sont inférieurs aux débits d'entrée. Nous supposerons que les débits de sortie sont proportionnels aux débits d'entrée. D'où :

$$thp_{out}(t) = C \cdot \frac{thp_{in}(t)}{thp_{in}(t) + E} \quad \text{et} \quad E_{out}(t) = C \cdot \frac{E}{thp_{in}(t) + E} \tag{2.2.7}$$

D'autre part, comme le buffer du goulot d'étranglement n'est pas vide, nous avons :

$$thp_{out}(t) + E_{out}(t) = C \tag{2.2.8}$$

2. Dans le cas de la version TCP Reno, le mode "congestion avoidance" a été modifié afin que l'augmentation de la fenêtre soit accélérée. Nous avons effectivement :

$$\frac{dW}{dack} = \begin{cases} 1 & \text{si } W \leq W_{th} \\ W^{-1} + \frac{1}{8} & \text{si } W > W_{th} \end{cases}$$

La source augmente donc sa fenêtre de 1/8 paquet à chaque acquittement reçu. Mais, d'après [STE97], cet ajout devrait être abandonné dans les versions ultérieures pour des questions de stabilité. C'est pour cette raison que nous ne considérerons pas cet aspect dans notre étude. Nous pouvons cependant en trouver une analyse fluide précise dans le document [COL98].

Les égalités 2.2.7 et 2.2.8 permettent d'obtenir une première expression de thp_{in} en fonction de thp_{out} :

$$thp_{in}(t) = thp_{out}(t)\frac{E}{C - thp_{out}(t)} \tag{2.2.9}$$

Le débit d'entrée de TCP dans le réseau étant égal au nombre de paquets acquittés plus l'augmentation de la fenêtre, nous avons également :

$$\begin{aligned} thp_{in}(t) &= thp_{out}(t) + \frac{dW(t)}{dt} \\ &= thp_{out}(t) + \frac{dW}{dack} \cdot \frac{dack(t)}{dt} \\ &= thp_{out}(t) + \frac{dW}{dack} \cdot \frac{thp_{out}(t)}{a} \end{aligned}$$

d'où

$$\boxed{thp_{in}(t) \quad = thp_{out}(t) \cdot \left(1 + \tfrac{1}{a} \cdot \tfrac{dW}{dack}\right) \qquad (2.2.10)}$$

où $\frac{dW}{dack}$ a pour expression 2.2.4.

A partir des équations 2.2.9 et 2.2.10, il vient l'expression de thp_{out} :

$$\boxed{thp_{out}(t) = C - \frac{E}{1 + \frac{1}{a} \cdot \frac{dW}{dack}} \qquad (2.2.11)}$$

Notons qu'en mode "congestion avoidance", le terme $\frac{1}{a} \cdot \frac{dW}{dack}$, égal à $\frac{1}{a \cdot W}$, peut être négligé en supposant que $W^{-1} \ll 1$.

Cette expression de thp_{out} permet de déduire l'expression de $\frac{dW}{dt}$, puis de W. Par ailleurs, concernant le remplissage $Q(t)$ de la file d'attente du goulot d'étranglement, nous avons :

$$W(t) - \tau_1 \cdot thp_{in}(t) - \left(\tau_2 + \frac{\tau}{2}\right) \cdot thp_{out}(t) - 1 = Q(t)\frac{thp_{in}(t)}{thp_{in}(t) + E}$$

Nous utiliserons l'ensemble des résultats démontrés pour un modèle à une seule file d'attente dans tous les cas de notre analyse où une seule des deux files d'attente sera non vide.

2.2.2.1 Remarques concernant les retards

Au cours de l'analyse fluide de TCP présentée dans le paragraphe précédent, nous n'avons pas tenu compte du retard dû à la propagation des paquets ou des acquittements sur les liens et au temps d'attente dans la file. Or, entre le temps auquel les paquets sortent du goulot au débit thp_{out} et le temps auquel les acquittements de ces paquets parviennent à la source, il s'écoule une durée $\tau_2 + \frac{\tau}{2}$. Ceci modifie les expressions de thp_{in}, thp_{out} et W, en ajoutant un retard supplémentaire dans les équations. Ainsi, l'équation 2.2.10 devient :

$$thp_{in}(t) = thp_{out}(t - (\tau_2 + \frac{\tau}{2})) \cdot \left(1 + \frac{1}{a} \cdot \frac{dW}{dack}\right)$$

Cependant, comme il est précisé dans [NAB+97], il est raisonnable de supposer que les débits de sortie sont proportionnels aux débits d'entrée tant que ceux-ci évoluent "lentement". Cela est généralement le cas, sauf dans les conditions présentées ci-dessous.

File d'attente vide et mode "slow start" : Néanmoins, lorsque nous sommes en mode "slow start", et que la file d'attente est vide, la fenêtre TCP croît très rapidement. Dans ce cas, nous ne pouvons pas faire l'approximation du paragraphe 2.2.2, qui comparait le débit thp_{out}, au temps t, avec le débit thp_{in}, au même instant. Nous sommes obligés de considérer un retard entre le départ de ces paquets depuis la source, leur arrivée à destination, et le retour des acquittements de ces paquets à la source . En effet, compte tenu de la vitesse de croissance de la fenêtre, les intervalles de temps séparant ces événements ne peuvent plus être considérés comme courts, et la fenêtre peut évoluer, voire doubler durant ces périodes.

Ainsi, le goulot d'étranglement étant vide, l'expression du débit d'arrivée des paquets TCP à leur destination s'écrit en tenant compte du retard entre l'émission des paquets TCP et l'instant où ces paquets sortent du goulot d'étranglement :

$$thp_{out}(t) = thp_{in}(t - \tau/2) = \frac{W(t - \tau/2)}{T}$$

Or, en tenant compte du retard entre l'arrivée des paquets à destination et le retour des acquittements correspondant à la source, on obtient :

$$\frac{dW(t)}{dt} = \frac{thp_{out}\,(t - (\tau_3 + \tau/2))}{a} \quad \text{si } W(t) \leq W_{th}$$

D'où, en combinant les deux équations précédentes :

$$\frac{dW(t)}{dt} = \frac{W(t - \tau)}{a \cdot T} \quad \text{si } W(t) \leq W_{th}$$

En suivant les recommandations de [BC63], nous posons $\frac{dW(t)}{dt} = \beta \frac{W(t)}{a \cdot T}$ et nous calculons β tel que $\frac{dW(t)}{dt} = \frac{W(t-\tau)}{a \cdot T}$. Nous avons alors :

$$\frac{dW(t)}{dt} = \beta \frac{W(t)}{a \cdot T} \quad \text{si } W(t) \leq W_{th} \tag{2.2.12}$$

Le coefficient β représente l'équivalent de la pente de $\frac{dW}{dt}$ en mode "slow start" [3]. L'erreur introduite par cette approche est faible car la phase considérée (mode "slow start" et file d'attente vide) est généralement courte et n'intervient que faiblement sur les performances globales.

2.3 Approche personnelle pour l'analyse de TCP sur ABR

Dans notre étude, nous combinons les deux approches précédentes et essayons d'adapter leurs cheminements à notre modèle : nous utilisons également une approche par équations différentielles pour notre analyse fluide et indiquons les différents états du système durant un cycle de contrôle TCP. Néanmoins, analyser une connexion TCP sur ABR avec flux exogène est plus complexe qu'analyser le protocole TCP ou le mécanisme ABR séparément. Premièrement, les états du système sont plus nombreux, car nous considérons cette fois deux files d'attente [4] et analysons ensemble les deux mécanismes de contrôle de congestion. Deuxièmement, nous mettons en évidence dans cette étude de nouveaux comportements, dûs aux interactions entre TCP et ABR : l'évolution de W dépend de l'évolution d'ACR et vice versa, ce qui entraîne des évolutions inhabituelles de ces variables en fonction du temps. Nous n'avons plus simplement des croissances et décroissances linéaires ou exponentielles : nous observons par exemple une croissance parabolique pour $ACR(t)$, une croissance logarithmique pour $W(t)$, ou encore une décroissance en argument de tangente hyperbolique pour $Q_1(t)$.

2.3.1 Méthode adoptée

A partir du modèle représenté et décrit au début de ce chapitre, nous abordons cette étude en différenciant plusieurs états dépendant de l'occupation des deux files d'attente, ainsi que de la monotonie des variables ACR et W. Dans chacun de ces états, nous cherchons à exprimer W, ACR, Q_1 et Q_2 en fonction des paramètres du réseau et des valeurs initiales de l'état. Pour cela, il nous suffit de connaître les débits entrant et sortant de chaque file d'attente et de reprendre les équations 2.2.5, 2.2.10 et 2.2.11, ainsi que celles du paragraphe 2.2.1. L'interaction de TCP et d'ABR conduit à combiner ces équations pour résoudre le système.

---

3. Pour tenir compte du retard en mode "slow start", lorsque les deux files sont vides, Collange, dans [COL94], encadre la pente de W par deux bornes :

$$\frac{W(t)}{2 \cdot a \cdot T} \leq \frac{dW(t)}{dt} \leq \frac{W(t)}{a \cdot T}$$

4. A. Legout dans [LEG96] présentait un modèle avec deux files d'attente, mais les deux taux de service étaient constants. Dans le cas de notre analyse, le taux de service de l'ingress buffer est variable, égal à $ACR(t)$.

Ainsi, le débit entrant dans l'ingress buffer est égal à thp_{in} et dépend de la taille de la fenêtre et du débit de sortie du goulot d'étranglement thp_{out}. De même, le débit de sortie de l'ingress buffer est égal à ACR lorsque ce buffer est non vide ; il dépend également du débit de sortie du goulot d'étranglement, qui dépend lui-même du flux exogène et du trafic contrôlé entrant dans ce goulot, à savoir ACR. Il s'ensuit toute une série d'imbrications entre les variables thp_{in}, thp_{out}, W, ACR, Q_1 et Q_2. De manière générale, les expressions des variables dans chaque état sont calculées en résolvant un système d'équations différentielles à six inconnues. Ce qui diffère pour chacun des états est la façon dont sont imbriquées ces variables. Ainsi, nous trouvons des états où ACR dépend de W et d'autres états où W dépend d'ACR. La méthode que nous suivons ici consiste à expliquer, pour chaque état, comment sont imbriquées les variables. Nous déterminons ensuite la variable qui nous permet de déduire toutes les autres. Cette imbrication des variables est un reflet des interactions entre les mécanismes TCP et ABR d'une part, entre le trafic contrôlé et le trafic exogène d'autre part.

2.3.2 Caractérisation des états

Pour analyser notre modèle et observer ses comportements, nous distinguons plusieurs phases traversées par notre connexion et définies ainsi :

Définition 2.3.1
On appelle "état" ou phase" toute période traversée par une connexion TCP sur ABR et caractérisée par :

- *la présence ou l'absence de paquets dans l'ingress buffer (notées respectivement $\bar{F}_1$ ou F_1)*
- *la présence ou l'absence de paquets dans le buffer du goulot d'étranglement (notées respectivement $\bar{F}_2$ ou F_2)*
- *la croissance ou la décroissance de l'occupation de l'ingress buffer (notées respectivement $\bar{F}_1^r$ ou $\bar{F}_1^v$)*
- *la croissance ou la décroissance de l'ACR (notées respectivement $\bar{F}_2^r$ ou $\bar{F}_2^v$)*
- *les modes "slow start" ou "congestion avoidance" (notés respectivement SS ou CA)*

Par abus de langage, on parle également "d'état" ou de "phase" dans le cas d'un regroupement de plusieurs états présentant une caractéristique commune, comme par exemple le cas où une file est vide et l'autre non vide.

■

La combinaison de tous les états conduit aux diagrammes présentés en annexe A, modélisant le comportement d'une connexion TCP sur ABR après élimination des états impossibles. Dans chaque phase, nous étudions les évolutions en fonction du temps des variables $ACR(t)$, $W(t)$, $Q_1(t)$ et $Q_2(t)$, et déterminons les transitions entre les états. Ces variables sont exprimées en fonction du temps t relatif à la date d'entrée dans l'état dans lequel nous nous trouvons, et non pas en fonction du temps absolu.

Ainsi, on note X_0 la valeur initiale de la variable X au début de chaque phase. De même, $t_{k_p}^{mn}$ est la date de transition vers l'état suivant : $k \in [0,3]$ (0 représente les états de la famille F_1F_2, 1 les états $\bar{F}_1F_2$, 2 les états $F_1\bar{F}_2$ et 3 les états $\bar{F}_1\bar{F}_2$), $p \in \{SS, CA\}$ indique si nous nous trouvons en mode "slow start" ou "congestion avoidance", $m \in \{r, v\}$ indique, dans le cas où F_1 est non vide, si cette file se remplit ou se vide, et $n \in \{r, v\}$ indique, dans le cas où F_2 est non vide, si l'ACR croît ou décroît. $t_{k_p}^{mn}$ est donc la date relative de transition vers l'état caractérisé par {k,p,m,n}, depuis l'état dans lequel nous nous trouvons. Il correspond à la durée de séjour dans l'état présent.

Exemple de notations : Nous sommes dans l'état $\bar{F}_1^r F_{2_{SS}}$: le buffer du goulot d'étranglement est vide, l'ingress buffer est non vide et se remplit. Nous sommes en mode "slow start". Lorsque nous arrivons dans cet état, $t = 0$ et $ACR = ACR_0$. Nous transitons vers l'état $\bar{F}_1^r \bar{F}_{2_{SS}}^r$ (où, en mode "slow start", l'ingress buffer est non vide et se remplit, le goulot d'étranglement est non vide et l'ACR croît) au temps relatif $t = t_{3_{SS}}^{rr}$ qui correspond donc à la durée de l'état $\bar{F}_1^r F_{2_{SS}}$.

L'étude de chacun des états du système constitue le corps de notre analyse d'une connexion TCP sur ABR, avec flux exogène.

2.4 Cas F_1 et F_2 vides : F_1F_2

L'ingress buffer et la file d'attente du goulot d'étranglement sont vides. Nous sommes dans l'état F_1F_2. Pour demeurer dans cet état, aucune des deux files ne doit se remplir, ce qui signifie que les taux d'arrivée des cellules dans ces deux files doivent rester inférieurs aux taux de service de ces files. D'autre part, comme l'ingress buffer est vide, le débit de sortie de la file 1 est égal au débit d'entrée dans cette file, soit thp_{in}. Or, le débit d'entrée dans la file 2 est égal au débit de sortie de la file 1, augmenté du taux d'arrivée du trafic exogène. Ceci conduit aux conditions suivantes :

1. $thp_{in}(t) < \frac{ACR(t)}{n_1}$
2. $thp_{in}(t) + E < C$

2.4.1 Le débit TCP et la fenêtre de congestion

Nous calculons dans cette section le débit d'entrée du flux TCP dans le réseau, ainsi que l'évolution en fonction du temps de la fenêtre de congestion W. Nous expliciterons clairement l'enchaînement des transitions et l'imbrication des mécanismes en fonction des grandeurs paramétrant le système.

A partir de l'équation 2.2.5, de son complément 2.2.12 valable en mode "slow start" pour des files d'attente vides, et de l'expression 2.2.6 de thp_{out} lorsque les files d'attente sont vides, nous obtenons :

$$\frac{dW(t)}{dt} = \begin{cases} \beta \frac{W(t)}{a \cdot T} & \text{si } W(t) \leq W_{th} \\ \frac{1}{a \cdot T} & \text{si } W(t) > W_{th} \end{cases}$$

Nous déterminons alors l'expression en fonction du temps de la fenêtre de congestion W :

$$W(t) = \begin{cases} W_0 \cdot \exp\left[\beta \frac{t}{a \cdot T}\right] & \text{si } W(t) \leq W_{th} \\ W_c + \frac{t - t_c}{a \cdot T} & \text{si } W(t) > W_{th} \end{cases} \quad (2.4.1)$$

avec

$$W_c = \begin{cases} W_0 & \text{si } W_0 > W_{th} \\ W_{th} & \text{si } W_0 \leq W_{th} \end{cases} \quad (2.4.2)$$

et

$$t_c = \begin{cases} 0 & \text{si } W_0 > W_{th} \\ t_{th} & \text{si } W_0 \leq W_{th} \end{cases} \quad (2.4.3)$$

t_{th} est l'instant auquel nous passons en mode "congestion avoidance", dans l'état F_1F_2. Ce passage a lieu lorsque $W(t_{th}) = W_{th}$, soit au temps :

$$t_{th} = \frac{a \cdot T}{\beta} \ln\left(\frac{W_{th}}{W_0}\right)$$

La valeur maximale de W_{th} est calculée dans la section 5.5, équation 5.5.2.

2.4.2 Evolution de l'Allowed Cell Rate

Nous cherchons à décrire ici la courbe d'évolution de l'ACR en fonction du temps. L'ACR peut être modifié à chaque réception d'une cellule RM. Or, les cellules RM sont émises toutes les N_{rm} cellules de données. Lorsque les deux files d'attente sont vides, le débit d'émission de ces cellules de données est égal au débit d'entrée des cellules de la source contrôlée dans le réseau, c'est-à-dire à thp_{in}. Le taux d'arrivée des cellules RM à la source ABR est donc égal au débit d'émission des cellules de données un RTT plus tôt, divisé par N_{rm}.

En reprenant le raisonnement suivi pour l'équation 2.2.3, nous obtenons l'équation différentielle :

$$\frac{dACR(t)}{dt} = n_1 \cdot \frac{thp_{in}(t - T')}{N_{rm}} \cdot RIF \cdot PCR$$

soit

$$\frac{dACR(t)}{dt} = n_1 \cdot \frac{W(t - T')}{T \cdot N_{rm}} \cdot RIF \cdot PCR$$

Bien entendu, les deux files d'attente étant vides, le réseau est dans un état non congestionné, et l'ACR ne peut que croître dans l'état F_1F_2. Pour des commutateurs ne modifiant que le bit CI des cellules RM, l'ACR est alors augmenté de $RIF \times PCR$ à chaque réception d'une cellule RM-backward, jusqu'à ce que le Peak Cell Rate (PCR) soit atteint. A partir de l'équation différentielle précédente et de 2.4.1, nous obtenons :

$$ACR^r(t) = \begin{cases} \text{si } W(t) \leq W_{th} \\ ACR_0 + \frac{n_1 \cdot a \cdot W_0 \cdot RIF \cdot PCR}{\beta \cdot N_{rm}} \left[\exp\left(\beta \frac{t-T'}{a \cdot T}\right) - \exp\left(-\beta \frac{T'}{a \cdot T}\right)\right] \\ \text{si } W(t) > W_{th} \\ ACR_c + \frac{n_1 \cdot RIF \cdot PCR}{T \cdot N_{rm}} \left[\left(W_c + \frac{t-2(t_c+T')}{2aT}\right) \cdot t \right. \\ \quad \left. - \left(W_c - \frac{t_c+2T'}{2aT}\right) \cdot t_c\right] \end{cases} \quad (2.4.4)$$

avec, de même,

$$ACR_c = \begin{cases} ACR_0 & \text{si } W_0 > W_{th} \\ ACR_0 & \\ \quad + \frac{n_1 \cdot a \cdot W_0 \cdot RIF \cdot PCR}{\beta \cdot N_{rm}} \left[\exp\left(\beta \frac{t_{th}-T'}{a \cdot T}\right) - \exp\left(-\beta \frac{T'}{a \cdot T}\right)\right] & \text{si } W_0 \leq W_{th} \end{cases}$$

Comme l'ACR ne doit pas dépasser le PCR, nous obtenons l'expression de l'ACR en fonction du temps suivante :

$$ACR(t) = \min\{PCR, ACR^r(t)\} \quad (2.4.5)$$

Remarquons que, dans l'état F_1F_2, l'évolution de l'ACR dépend de l'évolution de W car, F_1 étant vide, la source ABR ne peut pas être persistente.

2.4.3 Transitions possibles depuis l'état F_1F_2

Dans l'état F_1F_2, les deux files d'attente sont vides. Les transitions envisageables depuis cet état sont donc matérialisées par le remplissage de l'une de ces deux files. Nous identifions dans ce paragraphe lequel, de l'ingress buffer ou du goulot d'étranglement, se remplit en premier, afin de déterminer vers quel état s'effectue la transition. La réponse à cette question dépend des vitesses de croissance de l'ACR et de W. Nous déterminons également quand cette transition a lieu et si elle a lieu en mode "slow start" ou en mode "congestion avoidance".

2.4.3.1 Vers l'état $\bar{F}_1^r F_2$

Pour que la file F_1 de l'ingress buffer se remplisse, il faut que le débit d'entrée des paquets TCP dans le réseau, égal à thp_{in}, devienne supérieur au taux de service de l'ingress buffer, à savoir $\frac{ACR}{n_1}$. Ceci se produit à l'instant t_1^r tel que :

$$\frac{W(t_1^r)}{T} = \frac{ACR(t_1^r)}{n_1}$$

D'après les équations 2.4.1 et 2.4.4, si cet événement a lieu en mode "slow start", il se produit à l'instant :

$$t^r_{1_{SS}} = \frac{aT}{\beta} \ln \left[\frac{\frac{ACR_0}{n_1} - \frac{a \cdot W_0 \cdot RIF \cdot PCR}{\beta \cdot N_{rm} \cdot \exp \frac{\beta T'}{aT}}}{\frac{W_0}{T} - \frac{a \cdot W_0 \cdot RIF \cdot PCR}{\beta \cdot N_{rm} \cdot \exp \frac{\beta T'}{aT}}} \right]$$

La transition s'effectue alors vers l'état $\bar{F}^r_1 F_{2_{SS}}$.

Si la transition a lieu en mode "congestion avoidance" vers l'état $\bar{F}^r_1 F_{2_{CA}}$, elle se produit à l'instant $t^r_{1_{CA}}$, solution de l'équation du second degré :

$$\frac{RIF \cdot PCR}{2aT^2 \cdot N_{rm}} t^2_{1_{CA}} - \left[\frac{RIF \cdot PCR}{T \cdot N_{rm}} \left(\frac{t_c + T'}{aT} - W_c \right) + \frac{1}{aT^2} \right] t_{1_{CA}} + \frac{ACR_c - \frac{n_1 \cdot RIF \cdot PCR \cdot t_c}{T \cdot N_{rm}} \left(W_c - \frac{t_c + 2T'}{2aT} \right)}{n_1} + \frac{t_c}{aT^2} - \frac{W_c}{T} = 0$$

soit, d'après 2.4.1 et 2.4.4 :

$$t^r_{1_{CA}} = \frac{N_{rm}}{RIF \cdot PCR} + T' + t_c - a \cdot T \cdot W_c - \frac{1}{RIF \cdot PCR} \left\{ N^2_{rm} + 2N_{rm} \cdot RIF \cdot PCR \left(T' - aT^2 \frac{ACR_c}{n_1} \right) + \left[RIF^2 \cdot PCR^2 \left(T' - aT \cdot W_c \right) \right]^2 \right\}^{1/2}$$

2.4.3.2 Vers l'état $F_1 \bar{F}^r_2$

Pour que la file F_2 du goulot d'étranglement se remplisse alors que F_1 reste vide, il faut que le taux d'arrivée des cellules du flux contrôlé, à savoir thp_{in}, devienne supérieur à la capacité du lien diminuée du taux d'arrivée du trafic exogène sur ce lien. L'instant t^r_1 auquel se produit cette transition est donc tel que :

$$\frac{W(t^r_1)}{T} = C - E$$

D'après l'équation 2.4.1, si cet événement survient durant le mode "slow start", il se produit à l'instant :

$$t^r_{2_{SS}} = \frac{aT}{\beta} \ln \left(\frac{T \cdot (C - E)}{W_0} \right)$$

Nous passons alors dans l'état $F_1 \bar{F}^r_{2_{SS}}$.

D'après l'équation 2.4.1, si la transition a lieu en mode "congestion avoidance", elle se produit à l'instant :

$$t^r_{2_{CA}} = a \cdot T \cdot [T \cdot (C - E) - W_c] + t_c$$

Nous passons alors dans l'état $F_1\bar{F}^r_{2_{CA}}$.

Après calculs de ces quatre temps de transition qui représentent les quatre transitions envisageables depuis l'état F_1F_2, le prochain état est celui présentant le délai de transition le plus court. La durée de l'état F_1F_2 est égale à ce délai de transition. Toutefois, il faut tenir compte du mode dans lequel se trouve la source au début et à la fin de l'état F_1F_2. En effet, si la source est en mode "congestion avoidance", il est impossible de repasser en mode "slow start". De même, si au début de l'état F_1F_2, la source est en mode "slow start", il est impossible de transiter vers un mode "congestion avoidance" si le délai de transition est inférieur à t_{th}, de même qu'il est impossible de rester en mode "slow start" si le délai de transition est supérieur à t_{th}. La transition vers l'état suivant s'effectue donc :

- vers l'état $\bar{F}^r_1F_{2_{SS}}$ si :
  - $W_0 \leq W_{th}$: nous sommes en mode "slow start" au commencement de la phase F_1F_2)
  - $t^r_{1_{SS}} < t^r_{2_{SS}}$
  - $0 < t^r_{1_{SS}} \leq t_{th}$.

  Dans ce cas, la file F_1 de l'ingress buffer est saturée en premier, i.e le débit thp_{in} atteint le taux de service $\frac{ACR}{n_1}$ durant le mode "slow start", avant d'avoir atteint $C - E$.

- vers l'état $F_1\bar{F}^r_{2_{SS}}$ si :
  - $W_0 \leq W_{th}$
  - $t^r_{2_{SS}} \leq t^r_{1_{SS}}$ ou $t^r_{1_{SS}}$ n'existe pas (i.e $t^r_{1_{SS}} < 0$)
  - $0 < t^r_{2_{SS}} \leq t_{th}$.

  Dans ce cas, la file F_2 du goulot d'étranglement est saturée en premier, i.e le débit thp_{in} atteint la capacité restante du lien $C - E$ durant le mode "slow start", avant d'avoir atteint le taux de service $\frac{ACR}{n_1}$.

Si la transition ne se produit pas vers l'un des deux états précédents, c'est-à-dire si la transition n'a pas lieu en mode "slow start", le prochain état est alors :

- l'état $\bar{F}^r_1F_{2_{CA}}$ si $t_c \leq t^r_{1_{CA}} < t^r_{2_{CA}}$
- l'état $F_1\bar{F}^r_{2_{CA}}$ si $t_c \leq t^r_{2_{CA}} \leq t^r_{1_{CA}}$ ou $t^r_{1_{CA}}$ n'existe pas (i.e $t^r_{1_{CA}} < 0$).

Une fois déterminé le prochain état, nous savons lequel, de l'ingress buffer ou du goulot d'étranglement, se remplit en premier, et nous sommes capables de poursuivre l'observation du comportement de la connexion en étudiant à présent l'évolution des variables lorsque l'une ou l'autre des deux files se remplit.

2.5 Cas F_1 vide, F_2 non vide : $F_1\bar{F}_2$

Nous sommes ici dans l'état $F_1\bar{F}_2$. Afin de rester dans cet état, la file F_1 ne doit pas se remplir, ce qui impose que le débit d'entrée dans F_1 soit inférieur au taux de service de cette file. D'où la condition :

1. $\frac{ACR(t)}{n_1} > thp_{in}(t)$

D'autre part, la file F_2 doit être saturée, ce qui implique que le débit d'entrée dans la file F_2 soit supérieur au taux de service de cette file[5]. D'où la condition :

2. $thp_{in}(t) > C - E$

2.5.1 Evolution de l'ACR

A la différence de l'étude [RIT96], nous tenons compte ici du flux exogène et de la répartition des cellules dans le buffer du goulot d'étranglement. Les taux de sortie des paquets du flux contrôlé et des paquets du flux exogène sont en effet fonctions de cette répartition, comme le montrent les équations 2.2.7. L'influence du flux exogène génère alors des expressions de l'ACR particulières.

Dans cet état, deux phases d'évolution de l'Allowed Cell Rate en fonction du temps sont envisageables, selon que le seuil limite Q_H du goulot d'étranglement a été dépassé et que l'unité de contrôle de la source ABR a été informée de ce dépassement, ou non.

2.5.1.1 L'ACR est croissant

Le seuil limite Q_H du goulot d'étranglement n'a pas été dépassé ou bien la source ABR n'en a pas encore été informée. Le débit de retour des cellules RM est limité par le plus petit débit du lien accordé au trafic de notre connexion TCP sur ABR, à savoir par hypothèse, le débit de sortie des cellules RM hors du goulot d'étranglement, soit

---

5. Notre intuition pourrait nous amener à croire que nécessairement, pour que la file F_2 soit non vide, il faut que le débit thp_{in} ait été supérieur à la capacité du goulot d'étranglement. Ceci prouverait que la file F_2 est toujours croissante dans l'état $F_1\bar{F}_2$. Toutefois, il subsiste un cas particulier dont les étapes sont les suivantes :

   1. Dans un premier temps, la file F_1 se remplit lorsque ACR/n_1 devient inférieur à thp_{in}
   2. puis, ACR/n_1 redevient supérieur à thp_{in} et F_1 se vide
   3. ACR/n_1 continue de croître et dépasse $C - E$, la file F_2 se remplit, tandis que F_1 continue de se vider
   4. ACR se met à décroître, redescend en-dessous de $n_1(C - E)$, la file F_2 décroît, tandis que F_1 se vide toujours
   5. F_1 s'est totalement vidée et F_2 continue de décroître, car à aucun moment thp_{in} n'est devenu supérieur à $C - E$. Nous sommes alors dans l'état $F_1\bar{F}_2$, où $thp_{in} < C - E$ et F_2 décroît. Ceci est contraire à notre intuition qui affirmait que thp_{in} devait devenir supérieur à $C - E$ pour parvenir dans l'état $F_1\bar{F}_2$.

   En fait, nous verrons par la suite que notre intuition était bonne et que la phase 2 de ce cas particulier ne peut pas se réaliser.

$n_1 \cdot thp_{out}$. Par conséquent, les cellules RM reviennent au débit imposé par la file F_2. D'après l'équation 2.2.1, nous avons alors :

$$\frac{dACR(t)}{dt} = \frac{n_1 \cdot RIF \cdot PCR \cdot thp_{out}}{N_{rm}}$$

Or l'équation 2.2.11 fournit l'expression de thp_{out} :

$$thp_{out} = \begin{cases} C - \frac{a \cdot E}{1+a} & \text{si } W(t) \leq W_{th} \\ C - E & \text{si } W(t) > W_{th} \end{cases} \tag{2.5.1}$$

d'où

$$ACR(t) = \begin{cases} \min\left\{PCR, ACR_0 + \frac{n_1 \cdot RIF \cdot PCR \cdot \left(C - \frac{a \cdot E}{1+a}\right)}{N_{rm}} t\right\} & \text{si } W(t) \leq W_{th} \\ \min\left\{PCR, ACR_c + \frac{n_1 \cdot RIF \cdot PCR \cdot (C-E)}{N_{rm}} t\right\} & \text{si } W(t) > W_{th} \end{cases} \tag{2.5.2}$$

2.5.1.2 L'ACR est décroissant

La limite supérieure Q_H de la file F_2 a été dépassée et la source ABR en a été informée, au bout d'un temps τ correspondant au temps de retour d'une cellule RM. L'ACR décroît, à une vitesse toujours imposée par le débit de retour des cellules RM, à savoir $n_1 \cdot thp_{out}$. L'équation 2.2.2 permet alors d'écrire l'équation différentielle suivante :

$$\frac{dACR(t)}{dt} = -n_1 \cdot ACR(t) \frac{RDF \cdot thp_{out}}{N_{rm}}$$

d'où

$$ACR(t) = \begin{cases} \max\left\{MCR, ACR_0 \exp\left[-\frac{n_1 \cdot RDF \cdot \left(C - \frac{a \cdot E}{1+a}\right)}{N_{rm}} t\right]\right\} & \text{si } W(t) \leq W_{th} \\ \max\left\{MCR, ACR_c \exp\left[-\frac{n_1 \cdot RDF \cdot (C-E)}{N_{rm}} t\right]\right\} & \text{si } W(t) > W_{th} \end{cases} \tag{2.5.3}$$

2.5.2 Evolution de la fenêtre

A partir des équations 2.2.5 et 2.5.1 exprimant respectivement $\frac{dW}{dt}$ et thp_{out}, nous déterminons l'expression de l'évolution de la fenêtre de congestion en fonction du temps :

$$W(t) = \begin{cases} \left(\frac{C}{a} - \frac{E}{1+a}\right) t + W_0 & \text{si } W(t) \leq W_{th} \\ \sqrt{\frac{2}{a}(C-E)(t-t_c) + W_c^2} & \text{si } W(t) > W_{th} \end{cases} \tag{2.5.4}$$

W_c et t_c ont été donnés dans la section 2.4, équations 2.4.2 et 2.4.3. Par ailleurs, nous avons :

$$t_{th} = \frac{W_{th} - W_0}{\frac{C}{a} - \frac{E}{1+a}}$$

2.5.3 Le débit d'entrée thp_{in}

Comme nous l'avons vu dans le paragraphe 2.2.2, le débit d'entrée des paquets TCP dans le réseau, thp_{in}, est imposé par le débit d'arrivée des paquets à destination, thp_{out}, qui induit le débit de retour des acquittements à la source. L'équation 2.2.10 est donc a priori vérifiée. Toutefois, à cause des hypothèses restrictives de l'analyse fluide et de la non considération des retards dans les équations, il se peut que, lors d'un changement de phase, la fenêtre de congestion TCP ne soit pas en mesure d'assurer le débit calculé à partir de cette formule. Dans ce cas, le débit d'entrée est imposé non plus par le réseau, mais par la fenêtre W.

Lorsque notre système provient par exemple de l'état $F_1F_{2_{SS}}$, le passage dans l'état $F_1\bar{F}_{2_{SS}}$ entraîne une discontinuité du débit thp_{in}. En effet, juste avant le passage dans l'état $F_1\bar{F}_{2_{SS}}$, à l'instant $t^-_{2_{SS}}$, nous avons :

$$thp_{in}(t) = \frac{W(t)}{RTT(t)} = C - E$$

où RTT est le temps de transmission aller-retour d'un paquet, i.e la période délimitée par l'envoi d'un paquet et la réception de l'acquittement de ce paquet :

$$RTT(t) = \tau + n_1\frac{Q_1(t)+1}{ACR(t)} + \frac{Q_2(t)+1}{C}$$

Lorsque les deux files d'attente sont vides, $RTT = T$ et nous retrouvons l'équation 2.2.6.

Mais, juste après le passage dans l'état $F_1\bar{F}_{2_{SS}}$, à l'instant $t^+_{2_{SS}}$, nous avons, d'après les équations 2.5.1 et 2.2.10 :

$$thp_{in}(t) = (1+\frac{1}{a})C - E$$

Ce point de discontinuité de thp_{in} implique que cette variable est nécessairement supérieure à $\frac{W}{RTT}$ et à $C-E$ au passage dans l'état $F_1\bar{F}_{2_{SS}}$. La fenêtre W n'est donc pas encore suffisamment grande pour assurer le débit thp_{in} imposé par la file F_2. Par conséquent, le débit thp_{in} vaut en réalité :

$$thp_{in}(t) = \frac{W(t)}{RTT(t)} \text{ après la transition de } F_1F_{2_{SS}} \text{ vers } F_1\bar{F}_{2_{SS}}$$

Comme W est croissante alors que thp_{in} est constant, il se peut que par la suite $\frac{W}{RTT}$ devienne supérieur à $(1+\frac{1}{a})C-E$, auquel cas le débit thp_{in} sera de nouveau imposé par le réseau, c'est-à-dire par le retour des acquittements à la source. Ceci nous amène à la définition suivante :

DÉFINITION 2.5.1
Le débit d'émission des paquets TCP, thp_{in}, est limité par le retour des acquittements à la source, et par la taille de la fenêtre de congestion. Le débit d'entrée imposé par le

retour des acquittements, c'est-à-dire imposé par le réseau, est noté thp^r_{in} *et calculé grâce aux formules 2.2.10 ou 2.2.9. Le débit d'entrée imposé par la fenêtre de congestion vaut* $\frac{W}{RTT}$. *Nous avons alors :*

$$thp_{in}(t) = \min\left\{thp^r_{in}(t), \frac{W(t)}{RTT(t)}\right\}$$

■

Afin de simplifier les calculs, nous choisissons :

$$RTT = \tau + n_1\frac{Q_1(0)+1}{ACR(0)} + \frac{Q_2(0)+1}{C} \tag{2.5.5}$$

ce qui revient à négliger simplement les variations du RTT et à conserver sa valeur absolue. Cette approximation est d'autant plus valable que la période pendant laquelle $thp_{in} = \frac{W}{RTT}$ est courte : les files d'attente évoluent peu, autrement dit leur tendance à grossir ou à diminuer est négligeable par rapport à la taille globale. Nous vérifierons cette hypothèse par la suite.

D'autre part, l'expression de thp^r_{in} calculée à partir des formules 2.2.10 ou 2.2.9 est née de l'idée que, pour un système stable en boucle fermée, il est raisonnable de concevoir un équilibre où, à tout moment, les débits d'entrée sont approximativement proportionnels aux débits de sortie. En effet, dans le cas de TCP, le nombre de paquets à l'intèrieur du réseau se conserve pendant un RTT et n'augmente qu'au bout d'un RTT. Par conséquent, la période pendant laquelle $thp_{in} = \frac{W}{RTT}$ doit être considérée comme une période de convergence vers un état d'équilibre où $thp_{in} = thp^r_{in}$. D'où la définition :

DÉFINITION 2.5.2
On dit qu'une phase est "normale" lorsque le débit d'entrée des paquets TCP, thp_{in}, *est égal à* thp^r_{in}.

On dit qu'une phase est "transitoire" lorsque le débit d'entrée des paquets TCP est égal à $\frac{W}{RTT}$.

■

A partir de cette définition, nous énonçons le théorème suivant :

THÉORÈME 2.5.1
En mode "congestion avoidance", quel que soit l'état dans lequel nous nous trouvons, la phase "transitoire" est négligeable.

Preuve : En mode "congestion avoidance", l'accroissement de thp^r_{in} est de l'ordre de 1 paquet tous les RTT, ce qui est négligeable par rapport à la taille de la fenêtre. Les débits sont donc constants sur la connexion à l'ordre $o(\frac{1}{RTT})$. Aussi n'y a-t-il pas de point de discontinuité de thp^r_{in}, donc pas de phase transitoire.

■

Dans toute notre étude, nous considérerons donc que $thp_{in} = thp^r_{in}$ en mode "congestion avoidance". Ce théorème permet de démontrer la propriété suivante, valable pour l'état $F_1\bar{F}_{2_{CA}}$:

COROLLAIRE 2.5.1
En mode "congestion avoidance", lorsque F_1 est vide et F_2 est non vide, l'occupation Q_2 de la file d'attente du goulot d'étranglement est une fonction monotone croissante.

Preuve : D'après les équations 2.2.10 et 2.5.1, nous avons :

$$thp^r_{in}(t) = \begin{cases} (1+\frac{1}{a})C - E & \text{si } W(t) \leq W_{th} \\ (1+\frac{1}{aW(t)})C - E & \text{si } W(t) > W_{th} \end{cases} \quad (2.5.6)$$

Donc, lorsque $W \geq W_{th}$, $thp_{in} > C - E$, et la file d'attente du goulot d'étranglement ne peut que croître.

■

Nous remarquons que, pour calculer les débits en mode "congestion avoidance", nous ne tenons pas compte du terme en $\frac{1}{aW}$, négligeable par rapport à la taille de la fenêtre. Comme nous l'avons précisé plus haut, nous considérons que, dans un système stationnaire, les débits sont approximativement proportionnels et constants. En revanche, pour calculer l'accroissement des files d'attente et de la fenêtre TCP, nous devons tenir compte de cette légère augmentation d'un paquet dans le système tous les RTT. Nous utilisons dans ce cas le développement limité à l'ordre $o(\frac{1}{aW})$.

En mode "slow start", dans le cas général, il convient, au commencement de la phase $F_1\bar{F}_{2_{SS}}$, de comparer la valeur de $\frac{W_0}{RTT}$ avec $thp^r_{in} = \left(1+\frac{1}{a}\right)C - E$, afin de connaître le minimum de ces deux valeurs, et ainsi l'expression initiale de thp_{in}. Il est alors possible de compléter le corollaire 2.5.1 :

COROLLAIRE 2.5.2
Lorsque F_1 est vide et F_2 est non vide, si la condition 2.6.6 est satisfaite, le débit thp_{in} est toujours supérieur ou égal à la capacité du goulot d'étranglement $C - E$, quel que soit le mode de croissance de la fenêtre TCP, "slow start" ou "congestion avoidance".

La file d'attente Q_2 du goulot d'étranglement est donc toujours croissante dans l'état $F_1\bar{F}_2$.

Preuve : Cette propriété a été démontrée en mode "congestion avoidance" par le corollaire 2.5.1.

En mode "slow start", nous pouvons parvenir à l'état $F_1\bar{F}_{2_{SS}}$ depuis deux états : $F_1F_{2_{SS}}$ ou $\bar{F}_1\bar{F}_{2_{SS}}$. Dans le cas de la première transition (cf section 2.4), il faut que $\frac{W}{RTT} \geq C - E$. A cet instant, nous passons dans l'état $F_1\bar{F}_{2_{SS}}$. Le débit thp^r_{in}, de par son expression, sera toujours supérieur à $C - E$ dans cet état. Quant à $\frac{W}{RTT}$, dès le moment où il a été supérieur à $C - E$, il le restera durant tout le cycle TCP puisque $C - E$ est constant tandis que W est croissante. Ceci prouve que F_2 ne peut que croître si nous provenons de l'état $F_1F_{2_{SS}}$.

Passons à présent au cas particulier qui survient lorsque nous arrivons pour la première fois dans l'état $F_1\bar{F}_{2_{SS}}$ depuis l'état $\bar{F}_1\bar{F}_{2_{SS}}$ (cf section 2.7). La variable qui pose problème est $\frac{W}{RTT}$. Est-elle supérieure ou non à la capacité du goulot $C - E$ au moment de la

transition ? Remontons les étapes. Pour parvenir à l'état $\bar{F}_1\bar{F}_{2_{SS}}$ sans passer par l'état $F_1\bar{F}_{2_{SS}}$, il a fallu transiter par l'état $\bar{F}_1F_{2_{SS}}$ où $\frac{W}{RTT} \geq \frac{ACR}{n_1}$. Or, d'après les propriétés démontrées dans la section 2.6 en mode "slow start", si la condition 2.6.6 est satisfaite, $\frac{W}{RTT}$ croît plus rapidement que $\frac{ACR}{n_1}$. Ceci signifie que, lorsque nous passons dans l'état $\bar{F}_1\bar{F}_{2_{SS}}$, c'est-à-dire lorsque $\frac{ACR}{n_1}$ dépasse $C-E$, $\frac{W}{RTT}$ le dépasse également. Comme W est croissante et $C-E$ constant, $\frac{W}{RTT}$ sera toujours supérieur à $C-E$ lors du passage dans l'état $F_1\bar{F}_{2_{SS}}$. Le débit thp_{in}^r étant nécessairement supérieur à $C-E$ dans cet état, nous en déduisons le corollaire précédent.

■

La file F_2 ne peut donc pas décroître dans l'état $F_1\bar{F}_2$. Il n'en est pas de même d'ACR, qui peut parfaitement décroître lorsque le seuil Q_H du goulot d'étranglement est dépassé. Nous passons alors dans un état $F_1\bar{F}_2^v$ où la file F_2 continue de croître tandis qu'ACR décroît.

La file d'attente F_2 ne peut commencer à décroître que lorsque le débit d'entrée dans le goulot d'étranglement est devenu inférieur à $C-E$. Ceci ne peut se produire que si la source ABR, et non plus la source TCP, régule le débit des paquets qui parviennent au goulot. Pour cela, il faut que l'ingress buffer soit saturé ; nous ne sommes alors plus dans l'état $F_1\bar{F}_2$ mais dans l'état $\bar{F}_1\bar{F}_2$.

Exception : La file d'attente F_2 du goulot d'étranglement peut être amenée à décroître dans le cas où un paquet TCP est perdu. Nous considérons dans ce chapitre uniquement les transitions survenant à l'intérieur d'un même cycle TCP. Pour connaître l'enchaînement des événements dans le cas d'une perte de paquet, le lecteur est invité à se reporter au chapitre 5.

2.5.4 Etat transitoire : $\frac{W_0}{RTT} < thp_{in}^r$

Rappelons que ce cas ne nous préoccupe que si $W < W_{th}$, c'est-à-dire si nous sommes dans le mode "slow start". En effet, en mode "congestion avoidance", le fait de négliger le terme en $\frac{1}{aW}$ permet de ramener l'étude au cas $thp_{in} = thp_{in}^r$, que nous traitons dans la section suivante.

Dans le cas transitoire qui nous intéresse, nous avons :

$$thp_{in}(t) = \frac{W(t)}{RTT}$$

où, puisque F_1 est vide, $RTT = \tau + \frac{n_1}{ACR_0} + \frac{Q_{2_0}+1}{C}$.

Par intégration du débit entrant dans la file F_2, égal à $thp_{in} + E$ puisque F_1 est vide, moins le débit sortant, égal à C, nous obtenons le remplissage de la file d'attente F_2 :

$$Q_2(t) = Q_{2_0} + \int_0^t (thp_{in}(u) + E - C)\, du$$

Comme $thp_{in} = \frac{W}{RTT}$, d'après l'expression de W indiquée dans 2.5.4, nous trouvons :

$$Q_2(t) = \frac{1}{2RTT}\left(\frac{C}{a} - \frac{E}{1+a}\right)t^2 + \left(\frac{W_0}{RTT} - C + E\right)t + Q_{2_0} \qquad (2.5.7)$$

2.5.4.1 Transitions dans le cas où l'ACR croît

L'état qui nous intéresse ici est donc $F_1\bar{F}^r_{2_{SS}}$ transitoire. Nous cherchons les transitions possibles à partir de cet état.

La première transition envisageable est le passage en mode "congestion avoidance". Dans ce cas, nous passons dans un état $F_1\bar{F}_{2_{CA}}$ normal. Ceci se produit au temps t_{th} tel que :

$$W(t_{th}) = \left(\frac{C}{a} - \frac{E}{1+a}\right)t_{th} + W_0 = W_{th}$$

soit

$$t_{th} = \frac{W_{th} - W_0}{\frac{C}{a} - \frac{E}{1+a}}$$

Pour repasser dans un état $F_1\bar{F}_{2_{SS}}$ normal, il faut que $\frac{W}{RTT}$ atteigne thp^r_{in} en mode "slow start". Dès ce moment, la source TCP a une fenêtre W suffisamment grande pour maintenir le débit imposé par le réseau. Ceci se produit au temps $t^r_{2_{SS}}$ tel que :

$$\frac{W(t^r_{2_{SS}})}{RTT} = thp^r_{in}(t^r_{2_{SS}}) = \left(1 + \frac{1}{a}\right)C - E$$

soit, d'après les équations 2.5.4 et 2.5.6 :

$$t^r_{2_{SS}} = (1+a)RTT - \frac{(1+a)\cdot a \cdot W_0}{(1+a)C - aE}$$

Par ailleurs, durant cet état transitoire, thp_{in} n'est pas constant et il se peut que $\frac{W}{RTT}$ rejoigne $\frac{ACR}{n_1}$. Nous passons alors dans l'état $\bar{F}^r_1\bar{F}^r_{2_{SS}}$ transitoire au temps $t^{rr}_{3_{SS}}$ tel que :

$$\frac{W(t^{rr}_{3_{SS}})}{RTT} = \frac{ACR(t^{rr}_{3_{SS}})}{n_1}$$

soit, d'après les équations 2.5.4 et 2.5.2 :

$$t^{rr}_{3_{SS}} = \frac{\frac{ACR_0}{n_1} - \frac{W_0}{RTT}}{\frac{1}{RTT}\left(\frac{C}{a} - \frac{E}{1+a}\right) - \frac{RIF\cdot PCR\cdot\left(C - \frac{a\cdot E}{1+a}\right)}{N_{rm}}}$$

Enfin, le remplissage de la file F_2, Q_2, peut atteindre le seuil limite Q_H en phase transitoire si le goulot d'étranglement se remplit suffisamment vite. L'ACR commence alors à décroître en tenant compte du temps de retour des informations égal à τ. Nous passons dans l'état $F_1\bar{F}^v_{2_{SS}}$ transitoire au temps $t^v_{2_{SS}}$ tel que :

$$Q_2(t^v_{2_{SS}}) = Q_H$$

soit, d'après l'équation 2.5.7 :

$$t^v_{2_{SS}} = \tau + \frac{a(1+a)\left[(C-E)RTT - W_0\right]}{C + a(C-E)} + \frac{\sqrt{a(1+a)\left[-2(C+(C-E)a)\left(Q_{2_0} - Q_H\right)RTT + (1+a)\left(W_0 - (C-E)RTT\right)^2\right]}}{C + a(C-E)}$$

La transition s'effectue vers l'état présentant le délai de transition le plus court après calcul de ces quatre temps de transitions.

2.5.4.2 Transitions dans le cas où l'ACR décroît

De la même manière, à partir de l'état $F_1\bar{F}^v_{2_{SS}}$ transitoire, nous envisageons la possibilité d'une transition vers le mode "congestion avoidance", plus précisément vers l'état $F_1\bar{F}^v_{2_{CA}}$ normal. Ceci se produit au temps t_{th} identique au cas précédent puisque W présente la même expression dans les deux cas.

Le débit d'entrée imposé par le réseau, thp^r_{in}, a également la même expression lorsque l'ACR décroît. Par conséquent, l'instant $t^v_{2_{SS}}$ auquel $\frac{W}{RTT}$ rejoint thp^r_{in} est égal au temps $t^r_{2_{SS}}$ calculé dans le cas précédent. Cet instant correspond à la transition vers l'état $F_1\bar{F}^v_{2_{SS}}$ normal.

En revanche, l'ACR est à présent une fonction décroissante du temps. Ceci implique que le débit d'entrée dans le réseau, $thp_{in} = \frac{W}{RTT}$, rejoint plus rapidement le taux de service de la file F_1, $\frac{ACR}{n_1}$. La transition vers l'état $\bar{F}^r_1\bar{F}^v_{2_{SS}}$ transitoire se produit à l'instant $t^{rv}_{3_{SS}}$ tel que, d'après les équations 2.5.4 et 2.5.3 :

$$\frac{ACR_0}{n_1}\exp\left(-\frac{n_1 \cdot RDF \cdot \left(C - \frac{a\cdot E}{1+a}\right)}{N_{rm}} t^{rv}_{3_{SS}}\right) = \frac{1}{RTT}\left[\left(\frac{C}{a} - \frac{E}{1+a}\right)t^{rv}_{3_{SS}} + W_0\right]$$

La transition s'effectue vers l'état présentant le délai de transition le plus court après calculs de ces trois temps de transitions.

D'après le corollaire 2.5.2, la file F_2 ne peut décroître dans l'état $F_1\bar{F}_2$. Seul ACR est susceptible de décroître et le moment où ACR sera devenu inférieur à thp_{in} occasionne le passage dans un autre état où la file F_2 pourra alors éventuellement décroître lorsque l'ACR sera devenu inférieur à $C - E$.

2.5.5 Etat normal : $\frac{W_0}{RTT} \geq thp_{in}^r$

Nous sommes à présent dans un état $F_1\bar{F}_2$ normal où le débit d'entrée thp_{in} est imposé par le réseau. Comme $\frac{W_0}{RTT} \geq thp_{in}^r$, et que W croît tandis que thp_{in}^r est constant, nous restons dans une phase normale tant que dure l'état $F_1\bar{F}_2$. Nous avons donc :

$$thp_{in}(t) = thp_{in}^r(t) = \begin{cases} (1+\frac{1}{a})C - E & \text{si } W(t) \leq W_{th} \\ (1+\frac{1}{aW(t)})C - E & \text{si } W(t) > W_{th} \end{cases}$$

Rappelons qu'en mode "congestion avoidance", nous considérons que thp_{in} est toujours égal à thp_{in}^r.

En intégrant toujours le débit entrant dans la file F_2 moins le débit sortant, nous obtenons à présent :

$$Q_2(t) = Q_{2_0} + \int_0^t (thp_{in}^r(u) + E - C)\, du$$

soit, d'après l'équation 2.5.6 :

$$Q_2(t) = \begin{cases} Q_{2_0} + \frac{C \cdot t}{a} & \text{si } W(t) \leq W_{th} \\ Q_{2_c} + \frac{C}{C-E}\left(\sqrt{\frac{2}{a}(C-E)(t-t_c) + W_c^2} - W_c\right) & \text{si } W(t) > W_{th} \end{cases} \quad (2.5.8)$$

avec

$$Q_{2_c} = \begin{cases} Q_{2_0} & \text{si } W_0 > W_{th} \\ Q_{2_{th}} = Q_2(t_{th}) & \text{si } W_0 \leq W_{th} \end{cases}$$

Nous vérifions, comme démontré au théorème 2.5.2, que la file d'attente du goulot d'étranglement est effectivement toujours croissante.

2.5.5.1 Transitions dans le cas où l'ACR croît

Nous sommes dans l'état $F_1\bar{F}_2^r$ normal. Les expressions $W(t)$, $ACR(t)$ et $Q_2(t)$ sont toutes des fonctions croissantes du temps. Le débit $thp_{in} = thp_{in}^r$ quant à lui est constant, $\frac{W}{RTT} > thp_{in}^r$ et $\frac{ACR}{n_1} > thp_{in}$ par hypothèse. Par conséquent, la courbe du débit thp_{in} ne rejoindra jamais la courbe de l'ACR tant que l'ACR croît, et ne rejoindra pas non plus la courbe de $\frac{W}{RTT}$. La seule transition possible est donc vers l'état $F_1\bar{F}_2^v$, lorsque la file F_2 a atteint sa limite supérieure Q_H et que la source ABR en a été informée. L'ACR commence alors à décroître. Il reste à déterminer si cette transition a lieu dans le mode "slow start" ou "congestion avoidance".

Lorsque Q_2 atteint le seuil Q_H, le commutateur associé au nœud du goulot d'étranglement modifie le bit CI d'une cellule RM, notifiant ainsi à la source ABR qu'elle doit réduire son débit d'émission ACR. Il faut cependant attendre un temps $2\tau_3 + \tau_2$ (temps

de retour de la cellule RM) avant de passer dans l'état suivant et avant que l'ACR ne commence à décroître. L'instant t_2^v auquel se produit la transition est donc la somme du temps t_H tel que $Q_2(t_H) = Q_H$, plus le temps de retour des informations égal à $2\tau_3 + \tau_2$. D'après l'équation 2.5.8, la transition vers l'état $F_1\bar{F}^v_{2_{SS}}$ se produit en mode "slow start" à l'instant :

$$t^v_{2_{SS}} = 2\tau_3 + \tau_2 + \frac{a}{C}\left(Q_H - Q_{2_0}\right)$$

D'après cette même équation 2.5.8, la transition vers l'état $F_1\bar{F}^v_{2_{CA}}$ se produit en mode "congestion avoidance" à l'instant :

$$t^v_{2_{CA}} = 2\tau_3 + \tau_2 + t_c + \frac{a}{2(C-E)}\left[\left(\frac{C-E}{C}\left(Q_H - Q_{2_c}\right) + W_c\right)^2 - W_c^2\right]$$

En exceptant le cas où le buffer du goulot d'étranglement déborde, la transition s'effectue vers l'état $F_1\bar{F}^v_{2_{SS}}$ si $t^v_{2_{SS}} < t_{th}$, et vers l'état $F_1\bar{F}^v_{2_{CA}}$ sinon.

2.5.5.2 Transitions dans le cas où l'ACR décroît

Dans l'état $F_1\bar{F}^v_2$ normal, la file d'attente F_2 a dépassé le seuil Q_H et la source ABR en a été informée par l'intermédiaire d'une cellule RM. L'ACR décroît mais la fonction Q_2 est toujours croissante comme nous l'avons démontré plus haut. Par conséquent, tant que nous sommes toujours dans l'état $F_1\bar{F}_2$ et que l'ACR n'est pas devenu inférieur à $C - E$, le goulot d'étranglement peut toujours déborder.

Hormis cette possibilité d'un débordement du buffer du goulot d'étranglement, l'ACR peut également décroître suffisamment pour atteindre le débit thp_{in} qui est constant. Dans l'état $F_1\bar{F}_2$ normal, $\frac{ACR}{n_1}$ atteint toujours thp_{in} avant $C - E$. La file F_1 commence alors à se remplir et nous passons dans l'état $\bar{F}^r_1\bar{F}^v_2$.

Cette transition se produit en mode "slow start", vers l'état $\bar{F}^r_1\bar{F}^v_{2_{SS}}$, à l'instant $t^{rv}_{3_{SS}}$ tel que :

$$\frac{ACR(t^{rv}_{3_{SS}})}{n_1} = thp^r_{in}(t^{rv}_{3_{SS}})$$

soit, d'après les équations 2.5.3 et 2.5.6 :

$$t^{rv}_{3_{SS}} = -\frac{N_{rm}}{n_1 \cdot RDF \cdot \left(C - \frac{a \cdot E}{1+a}\right)} \ln\left(\frac{n_1\left((1+\frac{1}{a})C - E\right)}{ACR_0}\right)$$

La transition se produit en mode "congestion avoidance", vers l'état $\bar{F}^r_1\bar{F}^v_{2_{CA}}$, à l'instant $t^{rv}_{3_{CA}}$ tel que :

$$\frac{ACR(t^{rv}_{3_{CA}})}{n_1} = thp^r_{in}(t^{rv}_{3_{SS}})$$

soit, d'après ces mêmes équations 2.5.3 et 2.5.6 :

$$t^{rv}_{3_{CA}} = -\frac{N_{rm}}{n_1 \cdot RDF \cdot (C-E)} \ln\left(\frac{n_1(C-E)}{ACR_0}\right)$$

en négligeant le terme en $\frac{1}{aW}$.

En exceptant le cas où le buffer du goulot d'étranglement déborde, la transition s'effectue vers l'état $\bar{F}^r_1 \bar{F}^v_{2_{SS}}$ si $t^{rv}_{3_{SS}} \leq t_{th}$, et vers l'état $\bar{F}^r_1 \bar{F}^v_{2_{CA}}$ sinon.

Au passage dans l'état $\bar{F}^r_1 \bar{F}^v_2$, l'ACR va continuer à décroître jusqu'à atteindre la capacité $C-E$ du lien pour notre trafic TCP sur ABR. A cet instant, la file F_2 commence à décroître. Remarquons que thp_{in} étant voisin de $C-E$, l'ACR atteint rapidement $C-E$ dès que thp_{in} a été atteint. La file F_2 peut alors décroître.

2.6 Cas F_1 non vide et F_2 vide : $\bar{F}_1F_2$

Nous sommes dans l'état $\bar{F}_1F_2$. Afin de rester dans cet état, la file F_1 doit être saturée, ce qui implique que le débit d'entrée dans cette file soit supérieur au taux de service. D'où la condition :

1. $thp_{in}(t) > \frac{ACR(t)}{n_1}$

D'autre part, la file F_2 ne doit pas se remplir, ce qui impose que le débit d'entrée dans F_2, égal au débit de sortie de F_1 (soit $\frac{ACR(t)}{n_1}$ puisque F_1 est non vide) augmenté du taux d'arrivée du flux exogène E, reste inférieur au taux de service C de la file F_2. D'où la condition :

2. $C > \frac{ACR(t)}{n_1} + E$

2.6.1 Evolution de l'ACR

La file d'attente F_2 du goulot d'étranglement étant vide dans l'état $\bar{F}_1F_2$, le débit "Allowed Cell Rate" doit croître puisque nous n'avons pas atteint le seuil de limite supérieure Q_H de la file F_2. Nous n'avons donc à priori qu'une seule phase d'évolution de l'ACR.

Néanmoins, lorsque le système passe de l'état $\bar{F}^r_1 \bar{F}^v_2$ à l'état $\bar{F}_1F_2$, il est possible que l'ACR soit toujours en train de décroître. En effet, l'ACR se met à croître de nouveau à partir du moment où la source ABR a été informée que la file d'attente du goulot d'étranglement est descendue en-dessous du seuil de limite inférieure, Q_L. Dans le cas où nous passons de l'état $\bar{F}^r_1 \bar{F}^r_2$ à l'état $\bar{F}_1F_2$, le seuil Q_L a nécessairement été franchi (puisque le buffer du goulot d'étranglement est devenu vide), mais il se peut que la source ABR n'en ait pas encore été informée à cause du délai de retour des cellules RM. Dans ce cas, l'ACR continue à décroître jusqu'à ce que la cellule RM de retour des informations soit parvenue à l'unité de contrôle de la source ABR.

Nous pouvons choisir le seuil Q_L de manière à nous affranchir de ce phénomène. Il suffit que le temps pour écouler Q_L paquets dans le goulot d'étranglement soit supérieur au temps de retour des informations contenues dans les cellules RM, égal à $2\tau_3 + \tau_2$. Dans ce cas, la source ABR va être informée de la fin de la congestion (matérialisée par le dépassement du seuil inférieur Q_L) avant que le buffer du goulot d'étranglement ne se soit complètement vidé. L'ACR recommencera donc à croître avant que F_2 ne soit vide. La condition pour que le débit ACR ne puisse que croître dans l'état $\bar{F}_1 F_2$ est donc :

$$\frac{Q_L}{C} > 2\tau_3 + \tau_2 \tag{2.6.1}$$

Dans la suite de cette section, nous prendrons pour hypothèse que la condition précédente est vérifiée, c'est-à-dire que le seuil Q_L est choisi convenablement de manière à pouvoir ne considérer qu'une seule phase d'évolution de l'ACR, à savoir un ACR croissant dans l'état $\bar{F}_1 F_2$.

Le buffer du goulot d'étranglement étant vide et la source ABR étant persistente (puisque l'ingress buffer n'est pas vide), le débit de retour des cellules RM-backward est limité par le plus petit débit du lien accordé au trafic de notre connexion TCP sur ABR, à savoir par hypothèse, ACR. Le débit de retour des cellules RM au temps t dépend donc du débit ACR à l'époque où ces cellules ont été envoyées : cet événement s'est réalisé T' unités de temps plus tôt, c'est-à-dire au temps t moins le temps qu'il a fallu aux cellules pour parcourir le réseau. A l'époque où ces cellules RM ont été envoyées, le débit ACR valait donc $ACR(t - T')$. Par conséquent, d'après l'équation 2.2.3, nous avons :

$$\frac{dACR(t)}{dt} = ACR(t - T')\frac{RIF \cdot PCR}{N_{rm}}$$

Comme il est décrit dans le document [RIT96], nous proposons comme solution à cette équation :

$$\boxed{ACR(t) = \min\{PCR, ACR_0 \cdot \exp(\gamma \cdot t)\}} \tag{2.6.2}$$

où γ est solution de l'équation :

$$\gamma = \frac{RIF \cdot PCR}{N_{rm}} . \exp(-\gamma \cdot T)$$

2.6.2 Evolution de la fenêtre

Le débit thp_{out} des paquets TCP à la sortie du goulot d'étranglement est égal au débit des paquets TCP à l'entrée du goulot d'étranglement, puisque le buffer de ce goulot est vide. Nous avons donc :

$$thp_{out}(t) = \frac{ACR(t)}{n_1} \tag{2.6.3}$$

En effet, le débit des paquets TCP à l'entrée de la file F_2 est égal au débit à la sortie de F_1, soit $\frac{ACR(t)}{n_1}$ puisque l'ingress buffer n'est pas vide.

Nous en déduisons, d'après l'équation 2.2.5 :

$$\frac{dW(t)}{dt} = \begin{cases} \frac{ACR(t)}{a \cdot n_1} & \text{si } W(t) \leq W_{th} \\ \frac{ACR(t)}{a \cdot n_1 W(t)} & \text{si } W(t) > W_{th} \end{cases}$$

Notons que, contrairement au cas où F_1 et F_2 sont toutes les deux vides, c'est à présent W qui dépend d'ACR. L'expression de l'évolution de la fenêtre de congestion en fonction du temps est donc la suivante :

$$W(t) = \begin{cases} \frac{ACR_0}{a \cdot n_1 \cdot \gamma} \left[\exp(\gamma \cdot t) - 1\right] + W_0 & \text{si } W(t) \leq W_{th} \\ \sqrt{2 \frac{ACR_0}{a \cdot n_1 \cdot \gamma} \left[\exp(\gamma \cdot t) - \exp(\gamma \cdot t_c)\right] + W_c^2} & \text{si } W(t) > W_{th} \end{cases} \quad (2.6.4)$$

W_c et t_c ont été donnés dans la section 2.4 ; nous avons ici :

$$t_{th} = \frac{1}{\gamma} \ln \left[\frac{a \cdot n_1 \cdot \gamma}{ACR_0} \left(W_{th} - W_0 + \frac{ACR_0}{a \cdot n_1 \cdot \gamma} \right) \right]$$

2.6.3 Le débit d'entrée thp_{in}

De même que dans la section 2.5, nous avons :

$$thp_{in}(t) = \min \left\{ thp_{in}^r(t), \frac{W(t)}{RTT} \right\}$$

avec, ici, $RTT = \tau + n_1 \frac{Q_{1_0}+1}{ACR_0} + \frac{1}{C}$.

D'après les équations 2.2.10 et 2.6.3, nous trouvons :

$$thp_{in}^r(t) = \begin{cases} \left(1 + \frac{1}{a}\right) \frac{ACR(t)}{n_1} & \text{si } W(t) \leq W_{th} \\ \left(1 + \frac{1}{aW(t)}\right) \frac{ACR(t)}{n_1} & \text{si } W(t) > W_{th} \end{cases} \quad (2.6.5)$$

Comme dans l'état $F_1\bar{F}_2$, il se peut que, lors d'un changement de phase, par exemple lors du remplissage de F_1, la fenêtre TCP ne soit pas suffisamment grande pour assurer le nouveau débit thp_{in}^r qui lui est imposé par le réseau (en l'occurrence, par le remplissage de la file F_1). Ce phénomène est, comme nous l'avons souligné dans la section 2.5, dû à la discontinuité du débit thp_{in}^r lors de l'arrivée du système dans l'état $\bar{F}_1F_2$. Nous rentrons alors dans une phase transitoire où le débit d'entrée des paquets TCP dans le réseau, thp_{in}, est imposé par la taille de la fenêtre, et vaut donc W/RTT. D'après le théorème 2.5.1, cet état transitoire est envisagé uniquement en mode "slow start".

2.6.3.1 La phase transitoire

Nous allons comparer à présent les vitesses de croissance de thp_{in} et de $\frac{ACR}{n_1}$ en phase transitoire, c'est-à-dire lorsque $thp_{in} = \frac{W}{RTT}$. Nous en déduirons des règles d'évolution de Q_1 pour l'état $\bar{F}_1 F_{2_{SS}}$. Commençons par citer le théorème suivant :

THÉORÈME 2.6.1
Dans l'état $\bar{F}_1 F_{2_{SS}}$ transitoire, lorsque la condition 2.6.1 est respectée, les valeurs de $\frac{W}{RTT}$ et $\frac{ACR}{n_1}$ sont telles que :

$$\begin{array}{lrcl} & \dfrac{dW(t)}{RTTdt} & \geq & \dfrac{dACR(t)}{n_1 dt} \geq 0 \\ \textit{si et seulement si} & \dfrac{1}{a \cdot RTT} & \geq & \dfrac{RIF \cdot PCR}{N_{rm}} \end{array} \qquad (2.6.6)$$

Preuve : La condition 2.6.1 permet de garantir que le débit ACR est toujours croissant dans l'état $\bar{F}_1 F_{2_{SS}}$. Par ailleurs, d'après l'équation 2.2.5, nous avons, en mode "slow start" :

$$\begin{aligned} \frac{dW(t)}{dt} &= \frac{thp_{out}(t-T)}{a} \\ &= \frac{ACR(t-T)}{a \cdot n_1} \end{aligned}$$

d'où

$$\begin{aligned} \frac{dW(t)}{dt} &\geq T\frac{dACR(t)}{n_1 dt} \\ \Leftrightarrow \frac{ACR(t-T)}{a \cdot n_1} &\geq T\frac{ACR(t-T)}{n_1}\frac{RIF \cdot PCR}{N_{rm}} \\ \Leftrightarrow \frac{1}{a} &\geq T\frac{RIF \cdot PCR}{N_{rm}} \end{aligned}$$

■

Par expérience, la condition 2.6.6 est généralement respectée. Ceci signifie que la vitesse de croissance de thp_{in} dans l'état $\bar{F}_1 F_{2_{SS}}$ transitoire est en général supérieure à la vitesse de croissance d'$\frac{ACR}{n_1}$. Par la suite, nous supposerons que cette condition est satisfaite. De là découle le corollaire :

COROLLAIRE 2.6.1
Dans l'état $\bar{F}_1 F_{2_{SS}}$ transitoire, lorsque la condition 2.6.6 est remplie, si les conditions initiales de l'état sont telles que $\frac{W_0}{RTT} \geq \frac{ACR_0}{n_1}$, alors la file F_1 se remplit pendant toute la durée de l'état.

Preuve : Lorsque la condition 2.6.6 est remplie, $\frac{W}{RTT}$ croît plus vite qu'$\frac{ACR}{n_1}$. Donc, si $\frac{W_0}{RTT} \geq \frac{ACR_0}{n_1}$, le débit thp_{in}, égal à $\frac{W}{RTT}$ pendant l'état $\bar{F}_1 F_{2_{SS}}$ transitoire, restera supérieur à $\frac{ACR}{n_1}$ pendant toute la durée de l'état, ce qui implique que F_1 se remplit.

■

2.6.3.2 La phase normale

Nous considérons ici l'état $\bar{F}_1F_{2_{SS}}$ normal, ainsi que l'état $\bar{F}_1F_{2_{CA}}$ puisqu'il n'y a pas de phase transitoire en mode "congestion avoidance" d'après le théorème 2.5.1. Nous avons alors la propriété suivante :

THÉORÈME 2.6.2
Dans l'état $\bar{F}_1F_{2_{SS}}$ normal et dans l'état $\bar{F}_1F_{2_{CA}}$, la file d'attente de l'ingress buffer, F_1, est croissante. On ne parle alors plus que de l'état $\bar{F}_1^rF_{2_{SS}}$ normal et de l'état $\bar{F}_1^rF_{2_{CA}}$.

Preuve : En phase normale, $thp_{in} = thp_{in}^r$. Or, d'après l'équation 2.6.5, thp_{in}^r est supérieur à $\frac{ACR}{n_1}$ quel que soit le mode et à tout instant de l'état $\bar{F}_1F_2$ normal. Le débit entrant étant alors supérieur au débit sortant, l'occupation de F_1 augmente.

■

Etudions à présent les transitions envisageables depuis les phases transitoires ou normales.

2.6.4 Etat transitoire : $\frac{W_0}{RTT} < thp_{in}^r$

Il s'agit ici d'étudier l'état $\bar{F}_1F_{2_{SS}}$ transitoire. Dans ce cas, d'après l'équation 2.6.4, nous avons :

$$thp_{in}(t) = \frac{W(t)}{RTT} = \frac{1}{RTT}\left[\frac{ACR_0}{a \cdot n_1 \cdot \gamma}\left(\exp(\gamma \cdot t) - 1\right) + W_0\right]$$

En intégrant la différence entre le débit entrant dans la file F_1, thp_{in}, et le débit sortant, $\frac{ACR}{n_1}$, nous obtenons le remplissage de la file d'attente F_1 :

$$Q_1(t) = Q_{1_0} + \int_0^t \left(thp_{in} - \frac{ACR(u)}{n_1}\right) du$$

soit

$$\boxed{\begin{aligned} Q_1(t) = Q_{1_0} + \frac{ACR_0}{\gamma \cdot n_1}\left(\frac{1}{a \cdot \gamma \cdot RTT} - 1\right) \cdot \left(\exp(\gamma \cdot t) - 1\right) \\ + \left(\frac{W_0}{RTT} - \frac{ACR_0}{a \cdot n_1 \cdot \gamma \cdot RTT}\right) t \end{aligned}} \qquad (2.6.7)$$

Remarque : A l'instar de l'étude [RIT96], lors du calcul de Q_1, nous considèrons l'ACR préconisé par le commutateur du goulot d'étranglement et non l'ACR réellement respecté par la source. Ceci signifie que nous ne tenons pas compte des limitations de l'ACR par le MCR ou le PCR lors du calcul de Q_1. Ces bornes ne sont considérées que dans l'expression finale de $ACR(t)$. Ceci simplifie la description du modèle et conduit aux mêmes résultats.

2.6.4.1 Transitions possibles

Nous cherchons maintenant à déterminer les transitions depuis l'état $\bar{F}_1 F_{2_{SS}}$ transitoire.

La première transition envisageable est le passage en mode "congestion avoidance". Dans ce cas, nous passons dans un état $\bar{F}_1^r F_{2_{CA}}$ normal, comme nous l'avons démontré au théorème 2.6.2. Ceci se produit au temps t_{th} tel que $W(t_{th}) = W_{th}$; t_{th} a été calculé au paragraphe 2.6.2.

Il est également possible de revenir à un état $\bar{F}_1^r F_{2_{SS}}$ normal. Pour cela, il faut que $\frac{W}{RTT}$ atteigne thp_{in}^r en mode "slow start". Dès ce moment, la source TCP a une fenêtre W suffisamment grande pour maintenir le débit imposé par le réseau. Ceci se produit au temps $t_{1_{SS}}^r$ tel que :

$$\frac{W(t_{1_{SS}}^r)}{RTT} = thp_{in}^r(t_{1_{SS}}^r) = \left(1 + \frac{1}{a}\right) \frac{ACR(t_{1_{SS}}^r)}{n_1}$$

soit, d'après les expressions 2.6.4 et 2.6.5 :

$$t_{1_{SS}}^r = \frac{1}{\gamma} \ln \left(\frac{n_1}{ACR_0} \cdot \frac{\frac{ACR_0}{a \cdot n_1 \cdot \gamma \cdot RTT} - \frac{W_0}{RTT}}{\frac{1}{a \cdot \gamma \cdot RTT} - \left(1 + \frac{1}{a}\right)} \right) \tag{2.6.8}$$

D'autre part, lorsque la file d'attente F_1 décroît ce qui, comme nous l'avons vu, n'est pas exclu en phase transitoire, il se peut que l'ingress buffer se vide complètement auquel cas le prochain état est $F_1 F_{2_{SS}}$. Ceci se produit au temps $t_{0_{SS}}$ tel que :

$$Q_1(t_{0_{SS}}) = 0$$

Cette équation se résout numériquement, d'après l'expression de Q_1 (cf équation 2.6.7).

De même, depuis l'état $\bar{F}_1^v F_{2_{SS}}$ transitoire, il est possible de transiter vers l'état $\bar{F}_1^r F_{2_{SS}}$ transitoire lorsque le flux d'entrée dans F_1 devient supérieur au flux de sortie. Ceci se produit au temps $t_{1_{SS}}^{r,t}$ tel que :

$$\frac{W(t_{1_{SS}}^{r,t})}{RTT} = \frac{ACR(t_{1_{SS}}^{r,t})}{n_1}$$

soit, d'après les expressions 2.6.4 et 2.6.2 :

$$t_{1_{SS}}^{r,t} = \frac{1}{\gamma} \ln \left[\frac{W_0 a \cdot n_1 \cdot \gamma - ACR_0}{ACR_0(a \cdot \gamma \cdot RTT - 1)} \right]$$

Notons que la transition inverse, de l'état $\bar{F}_1^r F_{2_{SS}}$ transitoire vers l'état $\bar{F}_1^v F_{2_{SS}}$ transitoire, n'est possible que si la condition 2.6.6 n'est pas respectée. Dans ce cas, l'instant $t_{1_{SS}}^{v,t}$ de transition est égal à $t_{1_{SS}}^{r,t}$.

Enfin, il est possible que $\frac{ACR}{n_1}$ atteigne $C - E$ ce qui amène la file F_2 à se remplir. La transition s'effectue alors de l'état $\bar{F}_1^r F_{2_{SS}}$ transitoire vers l'état $\bar{F}_1^r \bar{F}_{2_{SS}}^r$, ou de l'état $\bar{F}_1^v F_{2_{SS}}$ transitoire vers l'état $\bar{F}_1^v \bar{F}_{2_{SS}}^r$. Ceci se produit à l'instant $t_{3_{SS}}^{rr}$ tel que :

$$\frac{ACR(t_{3_{SS}}^{rr})}{n_1} = C - E$$

soit, d'après l'équation 2.6.2 :

$$t_{3_{SS}}^{rr} = \frac{1}{\gamma} \ln \left[\frac{n_1 \cdot (C - E)}{ACR_0} \right] \tag{2.6.9}$$

La transition s'effectue vers l'état présentant le délai de transition le plus court après calculs de ces quatre temps de transitions.

2.6.5 Etat stable : $\frac{W_0}{RTT} \geq thp_{in}^r$

Nous sommes à présent dans un état $\bar{F}_1 F_2$ normal où le débit d'entrée thp_{in} est imposé par le réseau ($thp_{in} = thp_{in}^r$). Comme nous l'avons indiqué dans le théorème 2.6.2, les états à étudier sont $\bar{F}_1^r F_{2_{SS}}$ et $\bar{F}_1^r F_{2_{CA}}$ normaux. Dans ces états, la file d'attente de l'ingress buffer est nécessairement croissante. En effet, en intégrant le débit entrant dans la file F_1 moins le débit sortant, nous obtenons :

$$Q_1(t) = Q_{1_0} + \int_0^t \left(thp_{in}^r(u) - \frac{ACR(u)}{n_1} \right) du$$

soit, d'après les expressions 2.6.2 et 2.6.5 :

$$Q_1(t) = \begin{cases} Q_{1_0} + \frac{ACR_0}{a \cdot n_1 \cdot \gamma} \left(\exp(\gamma \cdot t) - 1 \right) & \text{si } W(t) \leq W_{th} \\ Q_{1_c} - W_c & \\ \quad + \sqrt{2 \frac{ACR_0}{a \cdot n_1 \cdot \gamma} \left(\exp(\gamma \cdot t) - \exp(\gamma \cdot t_c) \right) + W_c^2} & \text{si } W(t) > W_{th} \end{cases} \tag{2.6.10}$$

qui est une fonction croissante du temps. Rappelons que :

$$Q_{1_c} = \begin{cases} Q_{1_0} & \text{si } W_0 > W_{th} \\ Q_{1_{th}} = Q_1(t_{th}) & \text{si } W_0 \leq W_{th} \end{cases}$$

Hormis le cas d'un débordement de la file F_1, les transitions possibles depuis l'état $\bar{F}_1^r F_{2_{SS}}$ normal sont :

- vers l'état $\bar{F}_1^r F_{2_{SS}}$, à l'instant t_{th} dont l'expression est indiqué dans le paragraphe 2.6.2 ;
- vers l'état $\bar{F}_1^r F_2$ transitoire, si thp_{in}^r dépasse $\frac{W}{RTT}$, ce qui se produit à l'instant $t_{1_{SS}}^r$ calculé dans l'équation 2.6.8 ;

- vers l'état $\bar{F}_1^r \bar{F}_{2_{SS}}^r$, lorsque $\frac{ACR}{n_1}$ a atteint la valeur $C-E$. Ce cas intervient à l'instant $t_{3_{SS}}^{rr}$ calculé dans l'équation 2.6.9.

En mode "congestion avoidance", depuis l'état $\bar{F}_1^r F_{2_{CA}}$, nous pouvons toujours aller vers l'état $\bar{F}_1^r \bar{F}_{2_{CA}}^r$, à l'instant $t_{3_{CA}}^{rr}$ égal à $t_{3_{SS}}^{rr}$, lorsque $\frac{ACR}{n_1} = C-E$ et que $W > W_{th}$. Il s'agit de la seule transition envisageable depuis l'état $\bar{F}_1^r F_{2_{CA}}$, hormis le cas d'un débordement de buffer.

Remarque : A partir de l'état $\bar{F}_1^r F_{2_{CA}}$ normal, lorsque $\frac{ACR}{n_1}$ devient égal à $C-E$ et que nous sommes censés passer par conséquent dans l'état $\bar{F}_1^r \bar{F}_{2_{CA}}^r$, la transition s'effectue en fait directement vers l'état $\bar{F}_1^v \bar{F}_{2_{CA}}^r$. En effet, les courbes de $\frac{ACR}{n_1}$ et thp_{in} étant confondues dans l'état $\bar{F}_1^r F_{2_{CA}}$ normal, si $\frac{ACR}{n_1}$ devient égal à $C-E$, thp_{in} le devient également. L'ACR continue donc de croître alors que le débit d'entrée des paquets TCP dans le réseau, thp_{in}, reste imposé par le débit de sortie du goulot d'étranglement, et demeure donc voisin de $C-E$. C'est pourquoi nous pouvons considérer que nous passons directement de l'état $\bar{F}_1^r F_{2_{CA}}$ à l'état $\bar{F}_1^v \bar{F}_{2_{CA}}^r$.

2.7 Cas F_1 et F_2 non vides : $\bar{F}_1 \bar{F}_2$

Dans l'état $\bar{F}_1 \bar{F}_2$, les deux files sont saturées. Pour rester dans cet état, il faut et il suffit que les débits d'entrée dans ces deux files restent supérieurs aux taux de service. Nous avons alors les deux conditions suivantes :

1. $thp_{in}(t) > \frac{ACR(t)}{n_1}$
2. $\frac{ACR(t)}{n_1} + E > C$

2.7.1 Expression de l'Allowed Cell Rate

D'après les équations 2.2.1 et 2.2.2, le débit de retour des cellules RM à la source ABR est proportionnel au débit de sortie des paquets TCP hors du goulot d'étranglement. Nous avons donc :

$$\frac{dACR}{dt}(t) = \begin{cases} \frac{n_1 \cdot RIF \cdot PCR \cdot thp_{out}(t)}{N_{rm}} & \text{si l'ACR est croissant} \\ \frac{-n_1 \cdot RDF \cdot ACR(t) \cdot thp_{out}(t)}{N_{rm}} & \text{si l'ACR est décroissant} \end{cases} \tag{2.7.1}$$

Lorsque les deux files F_1 et F_2 sont non vides, l'équation 2.2.11, qui avait permis de déterminer aisément le débit thp_{out} dans les cas précédents, n'est malheureusement plus valable ici. Ceci rend plus complexe l'étude de l'état $\bar{F}_1 \bar{F}_2$, d'autant que le taux de service de la file F_1, ACR, est variable au cours du temps. En revanche, comme pour l'équation 2.2.7, il est toujours possible de considérer que les débits d'entrée dans le goulot d'étranglement sont, dans un système stable à boucle fermée, proportionnels au débits de sortie. D'où :

$$thp_{out}(t) = C \cdot \frac{ACR(t)}{ACR(t) + n_1 E} \tag{2.7.2}$$

A partir des équations 2.7.1 et 2.7.2, nous obtenons l'équation différentielle suivante :

$$\frac{dACR(t)}{dt} = \begin{cases} \frac{n_1 \cdot RIF \cdot PCR \cdot C \cdot ACR(t)}{N_{rm}(ACR + n_1 E)} & \text{si l'ACR croît} \\ \frac{n_1 \cdot RIF \cdot PCR \cdot C \cdot ACR(t)^2}{N_{rm}(ACR + n_1 E)} & \text{si l'ACR décroît} \end{cases} \tag{2.7.3}$$

Ceci conduit aux équations suivantes, d'inconnue $ACR(t)$:

- si $ACR(t)$ croît :

$$ACR(t) + n_1 E \ln(ACR(t)) = ACR_0 + n_1 E \ln(ACR_0) + \frac{n_1}{N_{rm}} C \cdot RIF \cdot PCR \cdot t \tag{2.7.4}$$

- si $ACR(t)$ décroît :

$$\ln(ACR(t)) - \frac{n_1 E}{ACR(t)} = \ln(ACR_0) - \frac{n_1 E}{ACR_0} - \frac{n_1}{N_{rm}} C \cdot RDF \cdot t \tag{2.7.5}$$

Ces deux équations, indiquant la valeur du débit ACR en fonction du temps, ne se résolvent que numériquement. Ceci nous amène à définir la variable "ACR" pour poursuivre la résolution du système. Ainsi, nous donnerons dans les paragraphes suivants les expressions de thp_{in}, thp_{out}, Q_1, Q_2 et W en fonction d'ACR. A partir des deux équations précédentes, il suffira alors de déterminer la valeur d'ACR à chaque instant pour obtenir les valeurs des autres variables [6].

Remarque : Afin de déterminer une expression explicite d'$ACR(t)$, nous aurions pu songer à faire l'approximation $thp_{out}(t) \approx C - E$ dans l'état $\bar{F}_1\bar{F}_2$. Cette approximation permet de s'affranchir de calculs trop complexes et n'aboutissant pas à des résultats explicites. Cette approximation est justifiée par le fait que, lorsque deux files sont non vides et que nous sommes en mode "congestion avoidance", la source TCP n'est pas très agressive et laisse le flux exogène occuper davantage le goulot d'étranglement. Cependant, cela revient à considérer que le flux exogène est prioritaire et que le taux de service accordé au flux TCP dans le goulot est égal à $C - E$. Dans ce cas, l'étude est ramenée à une étude sans flux exogène et nous nous reporterons simplement au document [AH98].

De plus, la particularité de notre étude est de considérer que le débit de la file F_1 peut varier. Or, s'il est vrai que TCP n'est pas très agressif dans le cas de deux files d'attente à débits constants (comme nous pouvons le vérifier dans le document [BES99]), il n'en est pas de même lorsque le débit ACR varie. Lorsque nous nous trouvons en mode "congestion

6. Si nous souhaitons obtenir une approximation d'ACR, il est possible de négliger la variation de $ln(ACR)$ par rapport à ACR ou à $1/ACR$

avoidance" et que les deux files sont remplies, il se peut effectivement que le débit thp_{in} soit peu agressif. Mais comme l'ACR varie et que la file F_1 est non vide, il est possible, dans le même temps, que le débit d'entrée dans la file F_2, égal à ACR, soit nettement supérieur à $n_1(C-E)$, auquel cas le flux d'entrée des cellules TCP dans le goulot d'étranglement est agressif vis-à-vis du flux exogène. En fait, comme $thp_{out} = C\frac{ACR}{ACR+n_1 E}$, nous ne pouvons considérer que $thp_{out} \approx C - E$ si et seulement si :

$$ACR \approx n_1(C-E)$$

ce qui n'est qu'occasionnellement le cas lorsque nous utilisons le "Relative Rate Marking" : même si l'ACR oscille autour de $n_1(C-E)$, l'amplitude de ces oscillations peut être suffisamment grande pour que nous ne puissions plus considérer qu'ACR est voisin de $C-E$. Le lecteur remarquera au passage, comme nous le découvrirons dans les expériences de la partie III, que l'amplitude de ces oscillations dépend des valeurs des paramètres Q_L, Q_H, RDF et RIF.

2.7.2 L'influence d'ACR sur les autres variables

Comme il est précisé dans le paragraphe précédent, nous déduisons les expressions de thp_{out}, W, thp_{in}, Q_1 et Q_2 à partir de la variable "ACR".

Nous commençons par citer une propriété du taux d'arrivée des paquets TCP :

THÉORÈME 2.7.1
Dans l'état $\bar{F}_1\bar{F}_2$, le débit de sortie thp_{out} ne peut être inférieur au minimum des taux de service des deux files, ni supérieur au maximum de ces taux de service :

$$\min\left\{C-E, \frac{ACR}{n_1}\right\} \leq thp_{out} \leq \max\left\{C-E, \frac{ACR}{n_1}\right\}$$

Preuve : D'après l'expression 2.7.2 de thp_{out} en fonction d'ACR, nous pouvons déduire les deux équivalences suivantes :

$$thp_{out} \geq C-E \Longleftrightarrow C-E \leq \frac{ACR}{n_1} \tag{2.7.6}$$

$$\text{et } thp_{out} \geq \frac{ACR}{n_1} \Longleftrightarrow C-E \geq \frac{ACR}{n_1} \tag{2.7.7}$$

d'où le théorème précédent.

■

Ce théorème engendre de nombreux corollaires en mode "congestion avoidance", comme nous le verrons dans le paragraphe traitant du débit thp_{in}.

D'après les équations 2.2.5 et 2.7.2, nous déterminons également l'expression de W en fonction d'ACR :

- lorsque l'ACR croît :

$$W(ACR) = \begin{cases} \frac{N_{rm}}{a \cdot n_1 RIF \cdot PCR}(ACR - ACR_0) + W_0 & \text{si } W \leq W_{th} \\ \sqrt{\frac{2 \cdot N_{rm}}{a \cdot n_1 RIF \cdot PCR}(ACR - ACR_c) + W_c^2} & \text{si } W > W_{th} \end{cases} \quad (2.7.8)$$

- lorsque l'ACR décroît :

$$W(ACR) = \begin{cases} -\frac{N_{rm}}{a \cdot n_1 RDF} \ln \frac{ACR}{ACR_0} + W_0 & \text{si } W <\leq W_{th} \\ \sqrt{-\frac{2 \cdot N_{rm}}{a \cdot n_1 RDF} \ln \frac{ACR}{ACR_c} + W_c^2} & \text{si } W > W_{th} \end{cases} \quad (2.7.9)$$

L'occupation de la file F_2 s'obtient toujours par intégration de la différence entre le débit entrant $\frac{ACR}{n_1} + E$ et le débit sortant C :

$$\frac{dQ_2}{dACR} = \frac{dQ_2}{dt} \cdot \frac{dt}{dACR} \quad (2.7.10)$$

$$= \left(\frac{ACR}{n_1} + E - C\right) \cdot \frac{dt}{dACR} \quad (2.7.11)$$

D'après les équations 2.7.11 et 2.7.3, nous obtenons, lorsque l'ACR croît :

$$Q_2(ACR) = Q_{2_0} + \frac{N_{rm}}{n_1 RIF \cdot PCR \cdot C}\left[\frac{ACR^2 - ACR_0^2}{2n_1}\right.$$
$$\left. -n_1 E(C-E)\ln\frac{ACR}{ACR_0} + (2E-C)\cdot(ACR - ACR_0)\right] \quad (2.7.12)$$

De même, lorsque l'ACR décroît :

$$Q_2(ACR) = Q_{2_0} - \frac{N_{rm}}{n_1 RDF \cdot C}\left[\frac{ACR - ACR_0}{n_1}\right.$$
$$\left. +n_1 E(C-E)\cdot\left(\frac{1}{ACR} - \frac{1}{ACR_0}\right) + (2E-C)\ln\frac{ACR}{ACR_0}\right] \quad (2.7.13)$$

L'évolution de Q_2 dépend simplement de la croissance ou de la décroissance d'ACR.

2.7.2.1 Le débit d'entrée thp_{in}

D'après les équations 2.2.10 et 2.7.2, nous trouvons :

$$thp_{in}^r(ACR) = \begin{cases} \left(1 + \frac{1}{a}\right) C \frac{ACR}{ACR + n_1 E} & \text{si } W \leq W_{th} \\ \left(1 + \frac{1}{aW(ACR)}\right) C \frac{ACR}{ACR + n_1 E} & \text{si } W > W_{th} \end{cases} \quad (2.7.14)$$

De même que dans les sections 2.5 et 2.6, nous distinguons une phase transitoire et une phase normale en mode "slow start" tandis que, d'après le théorème 2.5.1, seule la phase normale est envisagée en mode "congestion avoidance".

La phase transitoire : De même que dans les sections 2.5 et 2.6, nous avons en mode "slow start" :

$$thp_{in}(ACR) = \min\left\{thp^r_{in}(ACR), \frac{W(ACR)}{RTT}\right\}$$

avec, ici, $RTT = \tau + n_1\frac{Q_{1_0}+1}{ACR_0} + \frac{Q_{2_0}+1}{C}$.

Nous étudions dans ce paragraphe l'évolution de Q_1 dans le cas où $thp_{in} = \frac{W}{RTT}$, c'est-à-dire dans l'état $\bar{F}_1\bar{F}_{2_{SS}}$ transitoire.

L'occupation de la file F_1 s'obtient toujours par intégration de la différence entre débit entrant thp_{in} et le débit sortant $\frac{ACR}{n_1}$:

$$\frac{dQ_1}{dACR} = \frac{dQ_1}{dt}\cdot\frac{dt}{dACR} \tag{2.7.15}$$

$$= \left(\frac{W}{RTT} - \frac{ACR}{n_1}\right)\cdot\frac{dt}{dACR} \tag{2.7.16}$$

Etat $\bar{F}_1\bar{F}^r_{2_{SS}}$ transitoire : D'après les équations 2.7.16, 2.7.8 et 2.7.3, nous obtenons, lorsque l'ACR croît :

$$Q_1(ACR) = Q_{1_0} + \frac{N_{rm}}{n_1 RIF\cdot PCR\cdot C}\left[\frac{ACR^2 - ACR_0^2}{2n_1}\left(\frac{N_{rm}}{aRTT\cdot RIF\cdot PCR} - 1\right)\right.$$
$$+\left(\frac{W_0}{RTT} - E + \frac{N_{rm}}{a\cdot n_1 RTT\cdot RIF\cdot PCR}(n_1E - ACR_0)\right)(ACR - ACR_0)$$
$$\left.+\frac{n_1E}{RTT}\left(W_0 - \frac{ACR_0 N_{rm}}{a\cdot n_1 RIF\cdot PCR}\right)\ln\frac{ACR}{ACR_0}\right] \tag{2.7.17}$$

Etat $\bar{F}_1\bar{F}^v_{2_{SS}}$ transitoire : De même, lorsque l'ACR décroît, en utilisant à présent l'équation 2.7.9 :

$$Q_1(ACR) = Q_{1_0} + \frac{N_{rm}}{2a\cdot n_1^2 ACR_0\cdot ACR\cdot RDF^2\cdot C\cdot RTT}\times$$
$$\left[ACR_0\cdot N_{rm}\left(2a\cdot n_1 ACR\cdot RDF(E\cdot RTT - W_0) - 2n_1E + ACR\ln\frac{ACR}{ACR_0}\right)\ln\frac{ACR}{ACR_0}\right.$$
$$\left.+\left(2n_1E\cdot N_{rm} + 2a\left(ACR_0\cdot ACR\cdot RDF\cdot RTT - n_1^2E\cdot RDF\cdot W_0\right)\right)(ACR - ACR_0)\right] \tag{2.7.18}$$

La phase normale : Nous nous intéressons maintenant aux états $\bar{F}_1\bar{F}_2$ normaux, i.e à tous les états en mode "congestion avoidance", et aux états $\bar{F}_1\bar{F}_{2_{SS}}$, lorsque $thp_{in} = thp^r_{in}$.

L'expression de thp^r_{in} est donnée dans 2.7.14. Cette expression permet de déduire deux propriétés importantes :

COROLLAIRE 2.7.1
Dans l'état $\bar{F}_1\bar{F}_{2_{CA}}$, hormis le cas d'un débordement de buffer, les deux files F_1 et F_2 ne peuvent pas se remplir ni se vider simultanément.

Preuve : En mode "congestion avoidance", l'expression 2.7.14 indique que :

$$thp_{in}(ACR) = (1 + \frac{1}{aW(ACR)})C\frac{ACR}{ACR + n_1E}$$

En négligeant le terme en $\frac{1}{W}$, il vient :

$$\begin{aligned} thp_{in}(ACR) &\approx C\frac{ACR}{ACR + n_1E} \\ &= thp_{out}(ACR) \end{aligned}$$

En reprenant le théorème 2.7.1 valable pour thp_{out}, nous prouvons le corollaire précédent.

■

De la même manière, concernant le mode "slow start", nous pouvons énonocer le théorème suivant :

THÉORÈME 2.7.2
En mode "slow start", lorsque les deux files F_1 et F_2 sont non vides et que nous sommes dans une phase normale, le débit thp_{in} respecte l'encadrement suivant :

$$\min\left\{\frac{ACR}{n_1}, \left(1 + \frac{1}{a}\right)C - E\right\} \leq thp_{in} \leq \max\left\{\frac{ACR}{n_1}, \left(1 + \frac{1}{a}\right)C - E\right\}$$

Preuve : En mode "slow start", l'expression 2.7.14 indique que :

$$thp_{in}(ACR) = (1 + \frac{1}{a})C\frac{ACR}{ACR + n_1E}$$

D'où l'équivalence suivante :

$$thp_{in} \geq \frac{ACR}{n_1} \Leftrightarrow \left(1 + \frac{1}{a}\right)C - E \geq \frac{ACR}{n_1}$$

ce qui prouve le théorème précédent.

■

Nous en déduisons le corollaire :

COROLLAIRE 2.7.2
Dans l'état $\bar{F}_1\bar{F}_{2_{SS}}$ normal, hormis le cas d'un débordement de buffer, les deux files F_1 et F_2 ne peuvent pas se vider simultanément.

Preuve : Si Q_1 décroît, alors le débit d'entrée dans F_1 est inférieur au débit de sortie. Par conséquent, $thp_{in} \leq \frac{ACR}{n_1}$. Or, d'après le théorème 2.7.2 :

$$\begin{aligned} & thp_{in} \leq \frac{ACR}{n_1} \\ \Longrightarrow\ & thp_{in} \geq (1+\frac{1}{a})C - E \geq C - E \\ \Longrightarrow\ & \frac{ACR}{n_1} \geq C - E \end{aligned}$$

Comme $\frac{ACR}{n_1}$ est le débit entrant dans F_2 et $C-E$ le débit sortant, nous en déduisons que si Q_1 décroît, Q_2 croît.

De même, si Q_2 décroît, $\frac{ACR}{n_1} \leq C - E$. Or, d'après le théorème 2.7.2 :

$$\begin{aligned} & \frac{ACR}{n_1} \leq C - E \leq (1+\frac{1}{a})C - E \\ \Longrightarrow\ & \frac{ACR}{n_1} \leq thp_{in} \end{aligned}$$

Cela signifie que si Q_2 décroît, Q_1 croît. D'où le corollaire précédent.

■

Nous terminons ce paragraphe par les équations d'évolutions de Q_1 en phase normale. De même que pour la phase transitoire, nous intégrons le débit entrant moins le débit sortant :

$$\frac{dQ_1}{dACR} = \frac{dQ_1}{dt} \cdot \frac{dt}{dACR} \tag{2.7.19}$$

$$= \left(thp_{in} - \frac{ACR}{n_1}\right) \cdot \frac{dt}{dACR} \tag{2.7.20}$$

Nous déterminons donc les expressions de Q_1 en phase normale d'après les équations 2.7.20, 2.7.14 et 2.7.3.

Etat $\bar{F}_1\bar{F}^r_{2_{SS}}$ normal : L'ACR croît, nous sommes en mode "slow start", dans une phase normale :

$$Q_1(ACR) = Q_{1_0} + \frac{N_{rm}(ACR - ACR_0)}{n_1 RIF \cdot PCR \cdot C}\left[\left(C(1+\frac{1}{a}) - E\right) - \frac{ACR + ACR_0}{2n_1}\right] \tag{2.7.21}$$

Etat $\bar{F}_1\bar{F}^v_{2_{SS}}$ normal : L'ACR décroît, nous sommes en mode "slow start", dans une phase normale :

$$Q_1(ACR) = Q_{1_0} - \frac{N_{rm}}{n_1 RDF \cdot C}\left[\left(C(1+\frac{1}{a}) - E\right)\ln\frac{ACR}{ACR_0} - \frac{ACR - ACR_0}{n_1}\right] \quad (2.7.22)$$

Etat $\bar{F}_1\bar{F}^r_{2_{CA}}$: L'ACR croît, nous sommes en mode "congestion avoidance" :

$$Q_1(ACR) = Q_{1_c} + \frac{N_{rm}(ACR - ACR_c)}{n_1 RIF \cdot PCR \cdot C}\left[C - E - \frac{ACR + ACR_c}{2n_1}\right] \quad (2.7.23)$$

Etat $\bar{F}_1\bar{F}^v_{2_{CA}}$: L'ACR décroît, nous sommes en mode "congestion avoidance" :

$$Q_1(ACR) = Q_{1_c} - \frac{N_{rm}}{n_1 RDF \cdot C}\left[(C-E)\ln\frac{ACR}{ACR_c} - \frac{ACR - ACR_c}{n_1}\right] \quad (2.7.24)$$

2.7.3 Les transitions

Nous venons d'exprimer toutes les variables de notre système en fonction d'ACR. Comme nous avons pu le constater, les sous-états de l'état $\bar{F}_1\bar{F}_2$ sont différenciés uniquement par la croissance ou la décroissance de l'ACR, par l'expression de thp_{in}, et par le mode de croissance de la fenêtre. Il importe à présent de déterminer les instants de transitions vers d'autres états.

2.7.3.1 Transitions, lorsque l'ACR croît

Les dates de changement d'état sont déterminés par la valeur d'ACR lors des transitions et par la formule 2.7.4, qui conduit à :

$$t = \frac{ACR - ACR_0 + n_1 E \ln\frac{ACR}{ACR_0}}{\frac{n_1}{N_{rm}} C \cdot RIF \cdot PCR} \quad (2.7.25)$$

La valeur d'ACR permettant de calculer l'instant t de transition dépend de l'état vers lequel le système se dirige. Si l'ACR croît, les transitions vers d'autres états peuvent s'effectuer lorsque :

- $\frac{ACR}{n_1} = thp^r_{in}(ACR)$. Nous envisageons le cas d'un passage de l'état $\bar{F}^r_1\bar{F}^r_2$ normal à l'état $\bar{F}^v_1\bar{F}^r_2$ normal. D'après l'équation 2.7.14, cet événement survient :
  - en mode "slow start", lorsque $ACR = n_1\left[C(1+\frac{1}{a}) - E\right]$;
  - en mode "congestion avoidance", lorsque $ACR = n_1(C-E)$.

La transition inverse, de l'état $\bar{F}_1^v \bar{F}_2^r$ vers l'état $\bar{F}_1^r \bar{F}_2^r$, est impossible. En effet, thp_{in}, ACR et $C-E$ vérifient, en mode "slow start", le théorème 2.7.2, et, en mode "congestion avoidance", le théorème 2.7.1. De ce fait, dès lors qu'ACR est supérieur à thp_{in} en phase normale, il est également supérieur à $C-E$. Comme ACR croît tandis que $C-E$ est constant, ACR restera supérieur à $C-E$. Donc, d'après ces mêmes théorèmes, ACR demeurera supérieur à thp_{in} tant que durera l'état $\bar{F}_1 \bar{F}_2^r$.

- $\frac{ACR}{n_1} = \frac{W(ACR)}{RTT}$. Cette transition ne peut se produire qu'en mode "slow start" et au cours d'une phase transitoire. Elle marque le passage de l'état $\bar{F}_1^v \bar{F}_2^r$ transitoire à l'état $\bar{F}_1^r \bar{F}_2^r$ transitoire, ou inversement[7]. Nous avons alors, d'après 2.7.8 :

$$ACR = \frac{n_1 W_0 - \frac{N_{rm} \cdot ACR_0}{aRIF \cdot PCR}}{RTT - \frac{N_{rm}}{aRIF \cdot PCR}}$$

- $thp_{in}^r(ACR) = \frac{W(ACR)}{RTT}$. Cette transition ne peut également se produire qu'en mode "slow start". Elle marque le passage d'une phase normale à une phase transitoire, ou inversement. D'après les équations 2.7.14 et 2.7.8, ACR vaut alors :

$$\begin{aligned} ACR = \frac{1}{2N_{rm}} \big[& (ACR_0 - n_1 E) N_{rm} + n_1 RIF \cdot PCR (RTT \cdot C(1+a) - W_0) \\ & + \Big((N_{rm}(ACR_0 - n_1 E) + n_1 RIF \cdot PCR (RTT \cdot C(1+a) - W_0))^2 \\ & \qquad -4 n_1 E \cdot N_{rm} (n_1 RIF \cdot PCR \cdot W_0 - ACR_0 \cdot N_{rm}) \Big)^{1/2} \big] \end{aligned}$$

- Les limites de la file F_1 sont atteintes : $Q_1(ACR) = 0$ ou $Q_1(ACR) = B_1$. Le premier cas marque le passage de l'état $\bar{F}_1^v \bar{F}_2^r$ vers l'état $F_1 \bar{F}_2^r$, et le second le débordement de l'ingress buffer depuis l'état $\bar{F}_1^r \bar{F}_2^r$. D'après les expressions de Q_1 en fonctions d'ACR et de thp_{in}, ceci se produit :

  - en mode "slow start" et période normale (cf équation 2.7.21), lorsque :

$$\begin{aligned} ACR = n_1 \left(C(1 + \frac{1}{a}) - E \right) + \frac{n_1}{a} \Big\{ & (C + a(C-E))^2 + a \Big[-2C \frac{ACR_0}{n_1} \\ & + a \left(\frac{ACR_0^2}{n_1^2} + 2 \frac{ACR_0}{n_1} (E - C) + 2 \frac{C \cdot RIF \cdot PCR (Q_{1_0} - \{B_1, 0\})}{N_{rm}} \right) \Big] \Big\}^{1/2} \end{aligned}$$

7. Une nouvelle fois, nous pouvons comparer les croissances de $\frac{W}{RTT}$ et d'$\frac{ACR}{n_1}$ afin de savoir si nous pouvons passer de l'état $\bar{F}_1^v \bar{F}_2^r$ transitoire à l'état $\bar{F}_1^r \bar{F}_2^r$ transitoire, ou inversement. Par exemple, si $\frac{W}{RTT} < \frac{ACR}{n_1}$ et que $\frac{dW}{RTTdt} < \frac{dACR}{n_1 dt}$, nous n'avons aucune chance de transiter directement de l'état $\bar{F}_1^v \bar{F}_{2_{SS}}^r$ transitoire vers l'état $\bar{F}_1^r \bar{F}_{2_{SS}}^r$ transitoire. D'après l'expression 2.7.8 de W, nous avons la condition :

$$\begin{aligned} & \frac{dW}{RTTdt} \geq \frac{dACR}{n_1 dt} \\ & \Leftrightarrow N_{rm} \geq aRTT \cdot RIF \cdot PCR \end{aligned}$$

Cette condition n'est malheureusement pas généralisable, ni dans un sens, ni dans l'autre.

– en mode "congestion avoidance" (cf équation 2.7.23), lorsque :

$$ACR = n_1(C-E) + n_1\left[(C-E)^2 + 2\frac{ACR_c(E-C)}{n_1} + \frac{ACR_c^2}{n_1^2} + 2\frac{C\cdot RIF\cdot PCR(Q_{1_c}-\{B_1,0\})}{N_{rm}}\right]^{1/2}$$

– en mode "slow start" et période transitoire (cf équation 2.7.17), l'équation en ACR :

$$Q_1(ACR) = \begin{cases} 0 & \text{si } F_1 \text{ se vide} \\ B_1 & \text{si } F_1 \text{ se remplit} \end{cases}$$

ne présente pas de solution explicite. Aussi nous donnons, dans chaque système que nous étudions, directement la valeur numérique de la date de transition vers l'état $F_1\bar{F}^r_{2_{SS}}$ transitoire ou la date du débordement de la file F_1.

- Les limites ou le seuil supérieur du goulot d'étranglement sont atteints, c'est-à-dire que Q_2 vérifie l'une des trois égalités suivantes :

$$Q_2(ACR) = \begin{cases} 0 & \text{si } F_2 \text{ se vide} \\ Q_H & \text{si } F_2 \text{ se remplit} \\ B_2 & \text{si } F_2 \text{ se remplit} \end{cases}$$

Dans le premier cas, nous passons de l'état $\bar{F}^r_1\bar{F}^r_2$ vers l'état $\bar{F}^r_1F_2$, ou de l'état $\bar{F}^v_1\bar{F}^r_{2_{SS}}$ transitoire vers l'état $\bar{F}^v_1F_{2_{SS}}$ transitoire. Le passage de l'état $\bar{F}^v_1\bar{F}^r_2$ normal à l'état $\bar{F}^v_1F_2$ est impossible car les deux files ne peuvent pas se vider en même temps.

Dans le second cas, le seuil Q_H est dépassé ; l'ACR se met à décroître $2\tau_3+\tau_2$ plus tard et nous passons de l'état $\bar{F}^v_1\bar{F}^r_2$ à l'état $\bar{F}^v_1\bar{F}^v_2$, ou de l'état $\bar{F}^r_1\bar{F}^r_{2_{SS}}$ à l'état $\bar{F}^r_1\bar{F}^v_{2_{SS}}$. Le passage de l'état $\bar{F}^r_1\bar{F}^r_{2_{CA}}$ vers l'état $\bar{F}^r_1\bar{F}^v_{2_{CA}}$ est impossible car les deux files ne peuvent pas se remplir en même temps en mode "congestion avoidance".

Enfin, dans le dernier cas, le goulot d'étranglement déborde et il y a perte de paquet. Cette perte de paquet au niveau du goulot ne peut pas se produire dans l'état $\bar{F}^r_1\bar{F}^r_{2_{CA}}$ car les deux files ne peuvent pas se remplir en même temps.

Là encore, d'après l'expression 2.7.12 de Q_2, il n'y a pas de solution explicite en ACR pour ces équations. Nous donnons simplement une résolution numérique de celles-ci.

- $W(ACR) = W_{th}$. Nous passons en mode "congestion avoidance". D'après l'équation 2.7.8, ceci se produit lorsque :

$$ACR = ACR_0 + \frac{a\cdot n_1 RIF\cdot PCR(W_{th}-W_0)}{N_{rm}}$$

Nous venons d'indiquer toutes les transitions envisageables lorsque nous nous trouvons dans un sous-état de $\bar{F}_1\bar{F}_2^r$. En remplaçant dans la formule 2.7.25 ACR par sa valeur au moment de la transition, nous pouvons déterminer les instants de chaque transition. Comme nous l'avons vu, certaines transitions sont impossibles. Nous nous reporterons au diagramme d'états présenté en annexe, sur le schéma A.4, pour connaître l'ensemble des transitions possibles depuis ces sous-états.

2.7.3.2 Transitions, lorsque l'ACR décroît

Les dates de changement d'état sont déterminés par la valeur de l'ACR lors des transitions et par la formule 2.7.5, qui conduit à :

$$t = \frac{\ln \frac{ACR}{ACR_0} - n_1 E\left(\frac{1}{ACR} - \frac{1}{ACR_0}\right)}{\frac{n_1}{N_{rm}} C \cdot RDF} \tag{2.7.26}$$

Si l'ACR décroît, les transitions vers d'autres états peuvent s'effectuer lorsque :

- $\frac{ACR}{n_1} = thp_{in}^r(ACR)$. Cette transition marque le passage de l'état $\bar{F}_1^v\bar{F}_2^v$ normal à l'état $\bar{F}_1^r\bar{F}_2^v$ normal. La valeur d'ACR dans ce cas est calculée par les mêmes formules que lorsque l'ACR croît. Seule change la date de transition, calculée par la formule précédente.

  Notons que, de la même manière que lorsque l'ACR croît, la transition inverse, de l'état $\bar{F}_1^r\bar{F}_2^v$ normal vers l'état $\bar{F}_1^v\bar{F}_2^v$ normal, est impossible. En effet, thp_{in}, ACR et $C-E$ vérifient, en mode "slow start", le théorème 2.7.2, et, en mode "congestion avoidance", le théorème 2.7.1. De ce fait, dès lors qu'ACR est inférieur à thp_{in} en phase normale, il est également inférieur à $Cte \times C - E$. Comme ACR décroît tandis que $Cte \times C - E$ est constant, ACR restera inférieur à $Cte \times C - E$. Donc, d'après ces mêmes théorèmes, ACR demeurera inférieur à thp_{in} tant que durera l'état $\bar{F}_1\bar{F}_2^v$.

- $\frac{ACR}{n_1} = \frac{W(ACR)}{RTT}$ en mode "slow start" et phase transitoire. A cet instant, $\frac{ACR}{n_1}$ devient inférieur à $\frac{W}{RTT}$ (puisque ACR décroît tandis que W croît) et nous passons de l'état $\bar{F}_1^v\bar{F}_{2_{SS}}^v$ transitoire à l'état $\bar{F}_1^r\bar{F}_{2_{SS}}^v$ transitoire. Compte tenu de l'expression 2.7.9 de W lorsque l'ACR décroît, il est impossible d'obtenir une solution explicite d'ACR de cette équation aussi ne donnerons-nous qu'une évaluation numérique de la date de transition.

- $\frac{W(ACR)}{RTT} = thp_{in}^r(ACR)$ en mode "slow start". Comme thp_{in}^r décroît (puisque $\frac{dthp_{in}^r}{dt}$ est du même signe que $\frac{dACR}{dt}$) et que W croît, thp_{in}^r deviendra donc à cet instant inférieur à $\frac{W}{RTT}$ et nous passerons d'une phase transitoire à une phase normale. De même, compte tenu des expressions 2.7.14 et 2.7.9 de thp_{in}^r et W, nous ne donnons qu'une évaluation numérique de la date de transition.

- Dans une phase transitoire ou normale,

$$Q_1(ACR) = \begin{cases} 0 & \text{si } F_1 \text{ se vide} \\ B_1 & \text{si } F_1 \text{ se remplit} \end{cases}$$

Dans le premier cas, nous passons de l'état $\bar{F}_1^v\bar{F}_2^v$ à l'état $F_1\bar{F}_2^v$. Dans le second, nous sommes dans l'état $\bar{F}_1^r\bar{F}_2^v$, et l'ingress buffer déborde.

- Dans une phase transitoire ou normale,

$$Q_2(ACR) = \begin{cases} 0 & \text{si } F_2 \text{ se vide} \\ Q_L & \text{si } F_2 \text{ se vide} \\ B_2 & \text{si } F_2 \text{ se remplit} \end{cases}$$

Dans le premier cas, nous passons de l'état $\bar{F}_1^r\bar{F}_2^v$ vers l'état $\bar{F}_1^rF_2$, ou de l'état $\bar{F}_1^v\bar{F}_{2_{SS}}^v$ transitoire vers l'état $\bar{F}_1^vF_{2_{SS}}$ transitoire. Notons que la transition vers un état où l'ACR décroît et la file F_2 est vide n'est envisageable que si la condition 2.6.1 n'est pas respectée. De plus, le vidage de la file F_2 est impossible dans l'état $\bar{F}_1^v\bar{F}_2^v$ normal car les deux files ne peuvent pas se vider en même temps.

Dans le second cas, l'ACR recommence à croître dès l'arrivée de la cellule RM indiquant la fin de la congestion, et nous passons de l'état $\bar{F}_1^r\bar{F}_2^v$ vers l'état $\bar{F}_1^r\bar{F}_2^r$, ou de l'état $\bar{F}_1^v\bar{F}_{2_{SS}}^v$ transitoire vers l'état $\bar{F}_1^v\bar{F}_{2_{SS}}^r$ transitoire. La transition depuis l'état $\bar{F}_1^v\bar{F}_2^v$ normal vers l'état $\bar{F}_1^v\bar{F}_2^r$ normal est impossible car les deux files ne peuvent pas se vider en même temps en phase normale.

Enfin, dans le dernier cas, nous sommes dans l'état $\bar{F}_1^v\bar{F}_2^v$ ou dans l'état $\bar{F}_1^r\bar{F}_{2_{SS}}^v$, l'$ACR$ ne décroît pas assez vite, le buffer du goulot d'étranglement déborde et un paquet est perdu. Notons que le débordement du goulot est impossible dans l'état $\bar{F}_1^r\bar{F}_{2_{CA}}^v$ car les deux files ne peuvent pas se remplir en même temps dans le mode "congestion avoidance".

- $W(ACR) = W_{th}$. Nous passons en mode "congestion avoidance". D'après l'expression 2.7.9 de W, ceci se produit lorsque :

$$ACR = ACR_0 \exp\left[\frac{a \cdot n_1 RDF(W_0 - W_{th})}{N_{rm}}\right]$$

Nous nous reporterons au diagramme A.4 de l'annexe pour avoir un récapitulatif des diverses transitions envisageables depuis les sous-états de $\bar{F}_1\bar{F}_2$.

Toutes les équations de ces sous-états sont exprimées à partir de la variable ACR. Lors de l'implémentation de nos résultats, nous reprendrons la même démarche, qui consiste à déterminer l'ACR en fonction du temps, et les autres variables en fonction d'ACR.

Remarque : Notons qu'en raison de la conservation des paquets pendant un RTT, il est impossible que la file F_2 se vide entièrement sans que les paquets contenus dans cette file se retrouvent dans la file F_1. Aussi la décroissance de la file F_2 génère toujours une croissance de la file F_1, d'autant plus que le nombre de paquets dans le réseau augmente du seul fait que la taille de la fenêtre augmente toujours. De manière plus formelle, la compensation de la décroissance de la file F_2 s'exprime ainsi :

$$\frac{dQ_2}{dt} = thp_{in} - thp_{out} + E - \frac{dQ_1}{dt}$$

Cette équation rejoint les théorèmes 2.7.2 et 2.7.1 stipulant que les deux files F_1 et F_2 ne peuvent pas se vider simultanément en phase normale.

Nous avons décrit dans ce chapitre les différentes phases et transitions que peut rencontrer une connexion TCP sur ABR. Nous avons expliqué le comportement des sources, des destinations et de divers éléments du réseau. L'évolution des variables d'états a également été étudiée afin de comprendre les phénomènes et interactions qui entrent en jeu. Nous allons à présent poursuivre notre étude par le dimensionnement des mémoires tampons ainsi que des seuils de buffers, par la recherche du débit des connexions et du délai moyen de transmission d'un paquet, par l'observation du comportement de la connexion lors d'une perte de paquet, et enfin par l'analyse des modifications apportées à notre modèle par l'augmentation du nombre de connexions et de circuits virtuels.

Chapitre 3

Taille maximale des files d'attente

Il est intéressant pour les mécanismes de contrôle de congestion de connaître, en plus du débit de la connexion, la taille maximale que peut atteindre la longueur des files d'attente. Cela permet de dimensionner la taille des buffers de manière à éviter les pertes de cellules ou de paquets. Le calcul des longueurs maximales des files d'attente s'obtient à partir des équations différentielles présentées dans le chapitre 2. Nous montrons ici comment évaluer simplement ces grandeurs. Nous nous intéressons à la longueur maximale de la file F_1, puis à la longueur maximale de la file F_2. Bien entendu, cette longueur maximale est relative à l'état dans lequel nous nous trouvons.

3.1 Longueur maximale de la file F_1

La longueur maximale de la file d'attente de l'ingress buffer est atteinte lorsque la dérivée de $Q_1(t)$ en fonction du temps est nulle et que nous transitons d'un état où la file F_1 se remplit vers un état où la file F_1 se vide. Concrètement, ceci se produit lorsque le taux de service de l'ingress buffer devient égal, puis supérieur au débit d'entrée des paquets TCP dans l'ingress buffer. Précisons que nous avons dans ce cas un maximum local de la file d'attente F_1, en fonction de l'état dans lequel est notre système. Nous définissons ainsi l'instant $t_{1_{max}}$ auquel ce maximum est atteint :

DÉFINITION 3.1.1
Un maximum local de la longueur de la file d'attente F_1 est obtenu à l'instant $t_{1_{max}}$ tel que :

$$\frac{d^2Q_1(t_{1_{max}})}{dt^2} \leq 0 \ et \ \frac{dQ_1(t_{1_{max}})}{dt} = 0$$

soit tel que,

$$thp_{in}(t^+_{1_{max}}) \leq ACR(t^+_{1_{max}}),$$
$$thp_{in}(t^-_{1_{max}}) \geq ACR(t^-_{1_{max}}),$$
$$thp_{in}(t_{1_{max}}) = ACR(t_{1_{max}})$$

On ne tient pas compte du délai, égal à τ_1, entre le moment où thp_{in} devient égal à ACR, et l'instant où les paquets envoyés à ce débit parviennent à l'ingress buffer.

■

Les expressions de Q_1, thp_{in} et ACR sont celles exprimées dans le chapitre 2, dans les cas où F_1 se remplit. Remarquons que les instants $t_{1_{max}}$ correspondant aux maxima locaux de Q_1 ont déjà été calculés dans ce chapitre : ce sont en fait les temps de transition entre les phases où F_1 se remplit et les phases où F_1 se vide. Il reste donc à calculer la longueur maximale de F_1 à ces instants, à savoir $Q_{1_{max}} = Q_1(t_{1_{max}})$. Voici les cas de maxima locaux répertoriés au cours d'un cycle TCP :

- Lorsque seule la file F_1 est non vide, nous avons vu dans la section 2.6 que, si la condition 2.6.6 est vérifiée, la file F_1 ne peut pas se remplir puis se vider. D'après la définition précédente, il n'y a donc pas de maximum local pour Q_1 dans ce cas.

- Lorsque F_1 et F_2 sont toutes deux non vides, les maxima locaux se produisent aux instants $t_{1_{max}}$, calculés en fonction d'ACR dans la section 2.7, tels que :

  - ACR croît, $ACR_0 < n_1 \cdot thp_{in_0}$, nous sommes dans un état $\bar{F}_1^r \bar{F}_2^r$ normal, et l'ACR dépasse $n_1 \cdot thp_{in}^r$. Ceci se produit en mode "slow start" lorsque $ACR = n_1 [C(1 + 1/a) - E]$, et en mode "congestion avoidance" lorsque $ACR = n_1(C - E)$.
  - ACR croît, $ACR_0 < n_1 \frac{W_0}{RTT}$, nous sommes dans un état $\bar{F}_1^r \bar{F}_{2_{SS}}^r$ transitoire, et l'ACR dépasse $n_1 \frac{W}{RTT}$. Ceci ne peut se produire qu'en mode "slow start", lorsque $ACR = \frac{n_1 W_0 - \frac{N_{rm} ACR_0}{aRIF \cdot PCR}}{RTT - \frac{N_{rm}}{aRIF \cdot PCR}}$.
  - ACR décroît, $ACR_0 < n_1 \cdot thp_{in_0}$, thp_{in} décroît et rejoint $\frac{ACR}{n_1}$. Ceci se produit uniquement lorsque $ACR = 0$. C'est un cas très particulier.

  La taille maximale de la file F_1 dans les trois cas précédents est calculée à partir des formules démontrées dans la section 2.7, exprimant Q_1 en fonction d'ACR. Nous exprimons donc la valeur de $Q_{1_{max}}$ en fonction de la valeur d'ACR lorsque le maximum local est atteint.

Au vu des expressions des maxima locaux de Q_1, le maximum absolu atteignable par la file F_1 dépend des valeurs de départ Q_{1_0}, ACR_0 ou W_0. En effet, nous comprenons que plus les variables Q_1, ACR et W sont faibles au début de la croissance, plus elles vont avoir le temps de croître et d'atteindre un rythme de croissance élevé au moment de la saturation. Ainsi, plus le système commence sa phase de croissance depuis des valeurs de départ basses, plus il a d'inertie au moment où les buffers commencent à être saturés. Il est alors plus difficile et long d'interrompre la croissance de la file et nous atteignons un maximum pour la file F_1 plus important que si l'amplitude des oscillations de la taille de la file d'attente était faible, c'est-à-dire dans le cas de valeurs de départ de W_0, Q_{1_0} et ACR_0 plus élevées. Toutefois, il est bien évident, dans le cas de TCP, qu'il est impossible de spécifier un maximum absolu de la longueur de la file d'attente Q_1, étant donné que la fenêtre TCP va croître jusqu'au jour où une perte surviendra. Plus les buffers seront grands, plus la perte sera lointaine, mais elle finira toujours par arriver (sous réserve, bien sûr, qu'il subsiste des paquets à émettre ou que la taille maximale de la fenêtre de congestion ne soit pas limitée à une trop faible valeur). Notre but étant de protéger le réseau, nous souhaitons alors que les pertes de paquets se produisent dans F_1 plutôt que dans F_2. Afin que la saturation se produise d'abord dans l'ingress buffer, nous devons

donc de préférence éviter de surdimensionner F_1 par rapport à F_2. Ceci nous conduit finalement à ne pas spécifier de maximum pour F_1 et d'engager les opérateurs à ne pas utiliser de buffers d'accès plus grands que les buffers du réseau.

3.2 Longueur maximale de la file F_2

La longueur maximale de la file d'attente du goulot d'étranglement est obtenue de même à l'instant où la dérivée de $Q_2(t)$ s'annule, après avoir été positive. Cet événement correspond au moment où la file F_2 recommence à décroître, c'est-à-dire lorsque ACR devient inférieur au débit $C - E$. Nous définissons donc $t_{2_{max}}$, instant où un maximum local de Q_2 est atteint, par :

DÉFINITION 3.2.1
Un maximum local de la longueur de la file d'attente F_2 est obtenu à l'instant $t_{2_{max}}$ tel que :

$$\frac{d^2Q_2(t_{2_{max}})}{dt^2} \leq 0 \text{ et } \frac{dQ_2(t_{2_{max}})}{dt} = 0$$

soit tel que

$$ACR(t^-_{2_{max}}) \geq C - E$$
$$ACR(t^+_{2_{max}}) \leq C - E$$
$$ACR(t_{1_{max}}) = C - E$$

On ne tient pas compte du délai, égal à τ_2, séparant l'instant où ACR devient égal à $C - E$, du moment où les cellules émises à ce débit parviennent au goulot d'étranglement.

■

A la différence du cas précédent, les instants $t_{2_{max}}$ indiquant les maxima locaux n'ont pas été calculés jusqu'à présent. Ils correspondent aux moments où l'ACR est dans une phase décroissante et retombe à la valeur de saturation du goulot d'étranglement, à savoir $C - E$. A cet instant, la file d'attente du goulot, $Q_2(t)$, atteint son maximum local $Q_{2_{max}}$; à $t^+_{2_{max}}$, F_2 se met à décroître. Dans le chapitre 2, nous avons repertorié tous les cas où F_2 était non vide et où, simultanément, l'ACR était dans une phase décroissante, après que la file d'attente Q_2 eut dépassé le seuil Q_H et que la source ABR en eut été avertie. Les maxima locaux interviennent donc aux moments où ACR devient égal, puis inférieur à $C - E$:

- Lorsque seule F_2 est non vide, nous avons vu dans la section 2.5 que la file F_2 ne peut pas se vider, dans aucun des états de $F_1\bar{F}_2$. Il n'y a donc pas de maximum local pour Q_2 dans ce cas.

- Lorsque F_1 et F_2 sont toutes deux non vides, les maxima locaux de Q_2 sont obtenus lorsque l'ACR décroît et atteint $C - E$. Toutes les variables de l'état $\bar{F}_1\bar{F}_2$ étant exprimées en fonctions d'ACR, il est aisé d'obtenir $t_{2_{max}}$ et $Q_{2_{max}}$ à partir de la

valeur d'ACR dans ce cas, égale à $C - E$. Les maxima locaux de Q_2 se produisent donc :

- en mode "slow start", lorsque :

$$t_{2_{max}} = \frac{\ln \frac{C-E}{ACR_0} - n_1 E \left(\frac{1}{C-E} - \frac{1}{ACR_0}\right)}{\frac{n_1}{N_{rm}} C \cdot RDF}$$

et

$$Q_{2_{max}} = Q_{2_0} - \frac{N_{rm}}{n_1 C \cdot RDF} \left[\frac{C - E - ACR_0}{n_1} + (2E - C) \ln \frac{C-E}{ACR_0} + n_1 E (C - E) \left(\frac{1}{C-E} - \frac{1}{ACR_0}\right)\right] \quad (3.2.1)$$

- en mode "congestion avoidance", lorsque $ACR = thp_{in} = C - E$, ce qui se produit au même instant et pour la même valeur de $Q_{2_{max}}$ qu'en mode "slow start", puisqu'ACR et Q_2 ont la même expression en mode "slow start" et en mode "congestion avoidance".

Il est possible de dimensionner le buffer du goulot d'étranglement de manière à éviter tant que possible les pertes dans le réseau. En effet, en régime stationnaire, lorsque les deux files sont non vides, c'est l'ACR qui contrôle le trafic arrivant au goulot d'étranglement. Nous pouvons donc spécifier la taille maximale qu'il faut donner au buffer du goulot d'étranglement pour éviter une perte dans le réseau. En envisageant le pire cas à partir du débit maximal (PCR) et minimal (MCR) atteignables par l'ACR, cette taille vaut, d'après la formule 3.2.1 :

$$B_{2_{max}} \geq Q_H^* - \frac{N_{rm}}{n_1 C \cdot RDF} \left[\frac{C - E - PCR}{n_1} + (2E - C) \ln \frac{C-E}{PCR} + n_1 E (C - E) \left(\frac{1}{C-E} - \frac{1}{MCR}\right)\right] \quad (3.2.2)$$

Q_H^* représente la taille de la file d'attente lorsque la source ABR reçoit la première cellule RM avec le bit CI égal à 1. Dans le pire des cas, Q_H^* vaut :

$$Q_H^* = Q_H + (PCR/n_1 + E - C) \cdot (\tau_3 + \tau/2)$$

Ce dimensionnement de F_2 ne garantit l'absence de perte que si les deux files sont non vides. Si F_1 est vide, le contrôle de trafic est assuré par TCP et nous pouvons nous attendre, particulièrement en mode "slow start", à ce que la taille maximale du buffer soit occasionnellement dépassée. Néanmoins, ces pertes ne doivent pas nous préoccuper outre mesure, dans le sens où TCP a besoin de recalculer son seuil de "slow start" pour l'adapter au trafic du réseau. En revanche, en régime stationnaire, et c'est le cas qui nous intéresse, la source ABR doit contrôler le trafic.

3.2.1 Optimisation des seuils Q_H et Q_L

A partir de l'équation 3.2.2, il est possible d'optimiser la valeur de Q_H dans le pire cas en connaissant la taille du buffer du goulot d'étranglement.

Toutefois, il est louable d'éviter les pertes mais il est souhaitable également que la bande passante allouée au service ABR puisse être utilisée efficacement. Une façon de le permettre est d'exiger que la file F_2 ne se vide jamais complètement. Nous allons donc calculer la valeur minimale de la file F_2, de manière à ce que cette valeur ne s'annule jamais. Conformément au chapitre 2, le goulot d'étranglement se vide uniquement dans l'état $\bar{F}_1\bar{F}_2$. De même que pour le calcul des maxima, la valeur minimale de Q_2 est obtenue lorsque Q_2 passe d'une phase de décroissance à une phase de croissance. Ceci se produit lorsque le débit ACR a recommencé à croître et qu'il atteint la valeur $n_1(C-E)$. D'après la formule 2.7.12, nous avons donc :

$$Q_{2_{min}} = Q_L^* + \frac{N_{rm}}{n_1 RIF \cdot PCR \cdot C}\left[\frac{n_1^2(C-E)^2 - MCR^2}{2n_1}\right.$$
$$\left. -n_1E(C-E)\ln\frac{n_1(C-E)}{MCR} + (2E-C)\cdot(n_1(C-E)-MCR)\right] \quad (3.2.3)$$

Q_L^* représente la taille de la file d'attente lorsque la source ABR reçoit la première cellule RM avec le bit CI égal à 0. Dans le pire des cas, i.e si $ACR = MCR$:

$$Q_L^* = Q_L + (MCR/n_1 + E - C)\cdot(\tau_3 + \tau/2)$$

L'équation 3.2.3 indique que la valeur minimale de Q_2 peut être négative, si la valeur choisie pour Q_L est trop petite. Afin d'éviter une sous-utilisation de la bande passante, nous choisissons donc la valeur de Q_L qui annule l'expression de $Q_{2_{min}}$: la file F_2 effleure alors l'état vide, ce qui induit un comportement optimal et correspond à une file peu remplie et une bande passante entièrement utilisée. Par ailleurs, afin de limiter l'amplitude des oscillations, c'est-à-dire de minimiser la différence $Q_{2_{max}} - Q_{2_{min}}$, il faut choisir $Q_H = Q_L$. Ceci nous conduit au paramétrage des seuils Q_H et Q_L :

$$Q_L = Q_H = -(MCR/n_1 + E - C)\cdot(\tau_3+\tau/2) - \frac{N_{rm}}{n_1 RIF \cdot PCR \cdot C}\left[\frac{n_1^2(C-E)^2 - MCR^2}{2n_1}\right.$$
$$\left. +n_1E(C-E)\ln\frac{n_1(C-E)}{MCR} + (2E-C)\cdot(n_1(C-E)-MCR)\right] \quad (3.2.4)$$

Notons que la valeur de Q_H permet, grâce à l'équation 3.2.2, de déterminer la taille du buffer du goulot d'étranglement nécessaire pour éviter les pertes dans le réseau.

3.3 Nombre de paquets perdus

Afin de visualiser le nombre de paquets perdus par un nœud en fonction de la taille maximale de la file d'attente de ce nœud calculée plus haut, nous représentons sur la

figure 3.1 la différence entre une file d'attente "virtuelle" $(Q_j(t))$, dont l'évolution a été explicitée dans le chapitre 2, et une file d'attente "réelle", c'est-à-dire limitée par la taille du buffer.

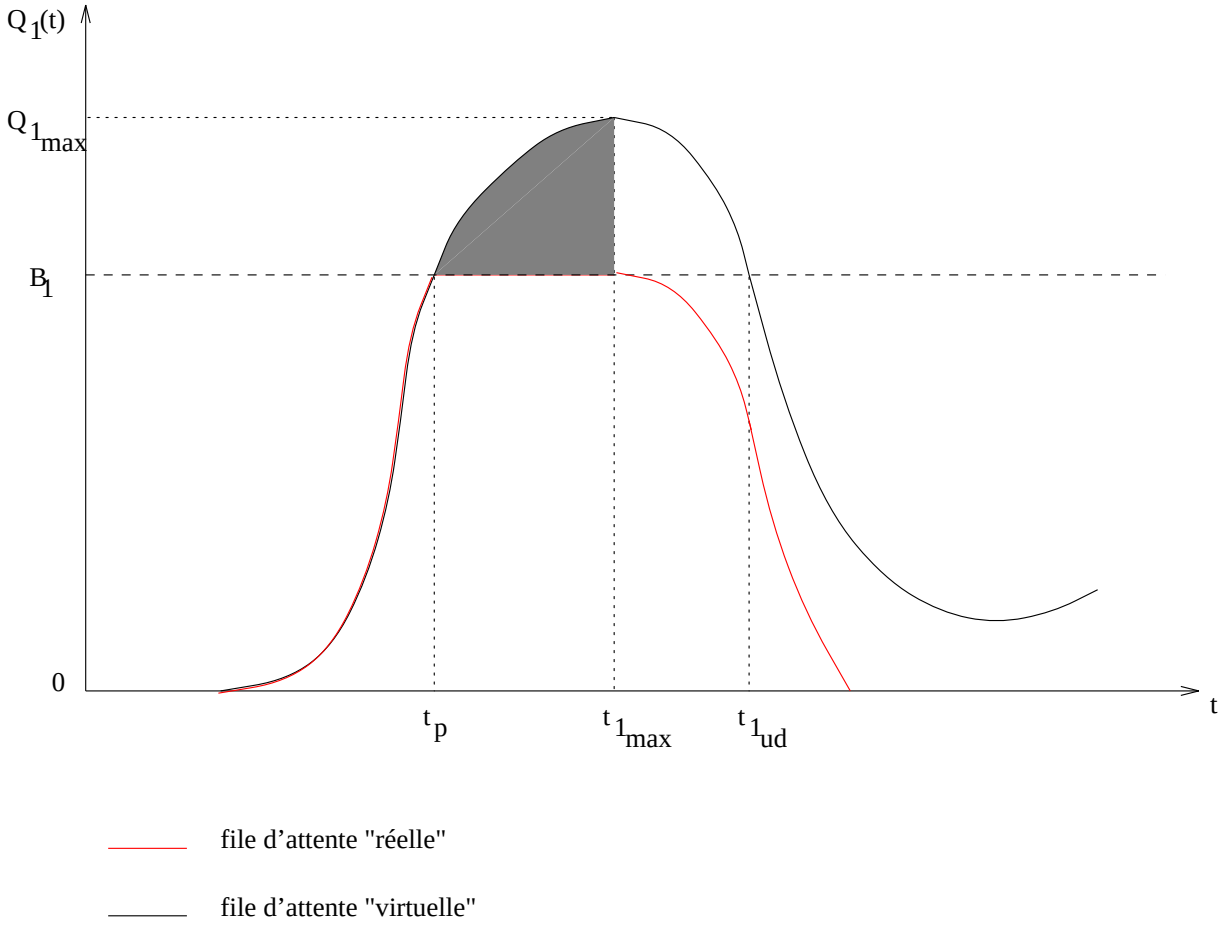


FIG. 3.1 - *File d'attente "réelle" et file d'attente "virtuelle"*

Si la longueur maximale d'une file d'attente dépasse la taille limite du buffer associé à cette file, c'est-à-dire si les contraintes énoncées précédemment concernant la taille maximale des files d'attente ne sont pas respectées, un certain nombre de paquets sera perdu. Ce nombre de paquets perdus correspondra au nombre de paquets qui sont arrivés au buffer mais qui n'ont pas pu être stockés par manque de place. Cela représente en moyenne la différence entre la longueur maximale qu'aurait atteint la file d'attente si le buffer avait été suffisamment grand, et le nombre de paquets réellement acceptés par ce buffer. Le nombre de paquets perdus vaut donc :

$$N_{p_j} = \frac{1}{t_{j_{max}} - t_p} \int_{t_p}^{t_{j_{max}}} (Q_j(t) - B_j)\, dt$$

où $j \in \{1, 2\}$ est associé à la file F_j.

Remarquons dans cette formule, comme nous pouvons le voir sur le schéma 3.1, que les paquets sont perdus entre l'instant où a lieu la première perte (t_p) et l'instant où la file

d'attente se met à décroître ($t_{j_{max}}$). En effet, la file "virtuelle" $Q_j(t)$ que nous obtiendrions si le buffer était suffisamment grand décroît au même moment que la file "réelle" : cela se produit à l'instant où le débit d'entrée devient inférieur au débit de sortie. Dès cet instant, la file "réelle" devient inférieure à la taille maximale du buffer, ce que ne montre pas la file "virtuelle" puisque sa valeur maximale est au-dessus de la taille limite du buffer. Le buffer ayant refusé des paquets, la valeur maximale de la file "réelle" ne dépasse pas la taille maximale du buffer et dès que la file se met à décroître, les nouveaux paquets arrivant à ce buffer peuvent être stockés. Par ailleurs, comme les débits d'entrée et de sortie ne dépendent pas du remplissage des files, nous pouvons également remarquer que la pente de la file d'attente "réelle" est identique à la pente de la file d'attente "virtuelle".

Chapitre 4

Calcul de Performances

En terme de performances, il est intéressant de calculer les valeurs moyennes du débit (THP) et du round trip time (RTT) des paquets émis par la source TCP. Ceci permet de quantifier le taux d'utilisation du lien et le délai moyen d'un paquet sur notre connexion TCP sur ABR.

Pour calculer ces valeurs, il s'agit d'exprimer la durée de chaque phase (T_i, $i \in [0,3]$), le nombre de paquets transmis durant chaque période (N_i) et le délai moyen d'un paquet TCP dans chaque phase (R_i), sachant que :

- Le débit moyen est égal au quotient du nombre de paquets N transmis avec succès au cours d'un cycle par la durée de ce cycle (c'est-à-dire la somme des durées de chaque phase) :

$$THP = \frac{W_{max} + \sum_{\text{phases i}} N_i}{RTO + \sum_{\text{phases i}} T_i} \tag{4.0.1}$$

  Pendant un RTO, W_{max} paquets sont transmis, ce qui explique les termes W_{max} et RTO respectivement du numérateur et dénominateur.

- Le round trip time moyen est égal au quotient du temps de séjour de tous les paquets TCP transmis durant le cycle par le nombre de paquets N transmis avec succès durant tout le cycle :

$$RTT = \frac{\sum_{\text{phases i}} N_i R_i}{\sum_{\text{phases i}} N_i} \tag{4.0.2}$$

- Le taux d'utilisation du lien pour la source TCP est le quotient du débit THP par le taux de service du goulot d'étranglement C :

$$LU = \frac{THP}{C}$$

Nous calculons dans ce chapitre les durées, le nombre de paquets transmis et le délai moyen d'un paquet en fonction de la phase dans laquelle nous nous trouvons. Les phases que peut rencontrer notre connexion TCP sur ABR ont été décrites dans le chapitre 2.

Pour le calcul explicite du débit THP et du round trip time RTT moyens en fin de connexion, il importe de spécifier, en fonction des paramètres de la connexion, quelles sont les phases que traverse notre connexion durant sa période d'activité, afin de connaître la durée de l'ensemble de la connexion, le nombre de paquets transmis et les délais de ces paquets. Une évolution de notre étude pourrait être alors de généraliser les phases que sera amenée à traverser une connexion TCP sur ABR, en fonction des paramètres spécifiés au départ. Cependant, il paraît difficile de déterminer une fonction exacte, capable d'indiquer précisément quels états notre réseau va traverser, car la multiplicité des paramètres de départ est trop importante. De plus, à cause de la complexité des interactions entre TCP et ABR, nous ne pouvons déterminer de comportement cyclique dans le cas d'une connexion TCP sur ABR, bien que nous soyons capables d'en trouver un dans le cas de connexions ABR ou TCP isolées. C'est la raison pour laquelle nous nous contentons simplement ici de calculer les performances du système état par état. Lors d'une expérience particulière, en connaissant les valeurs numériques de départ, nous pourrons ainsi calculer le débit et le round trip time dans chaque phase, puis faire la moyenne à la fin de la période d'activité de la connexion TCP sur ABR, c'est-à-dire une fois que nous aurons déterminé les états traversés par notre système.

4.1 Durée des périodes

Les durées des périodes (T_i) ont été calculées dans le chapitre 2 et correspondent aux dates de transitions entre les différentes phases, puisque nous raisonnons en temps relatif à chaque phase.

La durée d'un cycle TCP est égale à la somme des durées des phases traversées par le système durant ce cycle.

4.2 Nombre moyen de paquets transmis

Le nombre moyen de paquets TCP transmis avec succès durant la phase i est calculé en intégrant le taux d'arrivée des paquets TCP à leur destination, c'est-à-dire le débit de sortie du flux TCP hors du goulot d'étranglement :

$$N_i = \int_0^{T_i} thp_{out}(t)dt$$

A partir du débit thp_{out} et de la durée T_i de la phase i ($i \in [0,3]$), exprimés dans le chapitre 2, le calcul des N_i donne les résultats suivants, en fonction de la durée de la phase que traverse notre connexion :

$\underline{F_1 F_{2_{SS}}}$

Si $T_0 \leq t_{th}$:

$$N_{0_{SS}}(T_0) = \frac{a \cdot W_0}{\beta} \left[\exp\left(\frac{\beta \cdot T_0}{a \cdot T} \right) - 1 \right]$$

Si $T_0 > t_{th}$:

$$N_{0_{CA}}(T_0) = \frac{a \cdot W_0}{\beta} \left[\exp\left(\frac{\beta \cdot t_{th}}{aT} \right) - 1 \right] + \frac{(T_0 - t_{th})^2}{2 \cdot a \cdot T^2} + \frac{W_{th} \cdot (T_0 - t_{th})}{T}$$

$\underline{F_1 F_{2_{CA}}}$

$$N_{0_{CA}}(T_0) = \frac{T_0^2}{2 \cdot a \cdot T^2} + \frac{W_{th} \cdot T_0}{T}$$

$\underline{\bar{F}_1 F_2}$

$$N_1(T_1) = \frac{ACR_0}{\gamma \cdot n_1} \left[\exp\left(\gamma \cdot T_1 \right) - 1 \right]$$

$\underline{F_1 \bar{F}_{2_{SS}}}$

Si $T_2 \leq t_{th}$:

$$N_{2_{SS}}(T_2) = \left(C - \frac{E}{1 + \frac{1}{a}} \right) T_2$$

Si $T_2 > t_{th}$:

$$N_{2_{CA}}(T_2) = \left(C - \frac{E}{1 + \frac{1}{a}} \right) t_{th} + (C - E) \cdot (T_2 - t_{th})$$

$\underline{F_1 \bar{F}_{2_{CA}}}$

$$N_{2_{CA}}(T_2) = (C - E) \cdot T_2$$

$\underline{\bar{F}_1 \bar{F}_2^r}$

$$N_3^r(T_3) = \frac{N_{rm}}{n_1 RIF \cdot PCR} [ACR(T_3) - ACR_0]$$

$\underline{\bar{F}_1 \bar{F}_2^v}$

$$N_3^v(T_3) = -\frac{N_{rm}}{n_1 RDF} \ln \frac{ACR(T_3)}{ACR_0}$$

Comme nous pouvons le constater, certains états peuvent être regroupés et transmettent, à durée égale, le même nombre de paquets.

A partir de la durée d'une phase et du nombre de paquets transmis durant cette période, on pourra calculer le débit moyen de la connexion TCP durant la phase considérée en utilisant la formule 4.0.1.

4.3 Nombre moyen de paquets émis par la source

Nous pouvons également nous intéresser au nombre de paquets émis par la source, dans le but par exemple de le comparer au nombre de paquets effectivement parvenus jusqu'à la destination. Ce nombre de paquets envoyés par la source est simplement fonction du débit d'entrée des paquets dans le réseau, soit encore de la taille de la fenêtre de congestion. Ainsi :

$$N_e(t) = N_{e_0} + \int_0^t thp_{in}(u)du$$

Le nombre de paquet émis par la source TCP au cours d'une phase se calcule donc facilement en intégrant le débit entrant des paquets TCP pendant la durée de cette phase :

$$N_{e_i} = \int_0^{T_i} thp_{in}(u)du$$

Ces paquets émis comprennent bien sûr l'ensemble des paquets qui seront perdus par le débordement de l'ingress buffer ou du goulot d'étranglement.

4.4 Délai moyen d'un paquet

Le délai moyen d'un paquet TCP, vu par ce paquet, durant la phase i est la somme du temps de traversée à vide, plus le temps passé dans la file F_1, plus le temps passé dans la

file F_2. Nous avons alors :

$$R_i = T + \int_0^{T_i} \frac{Q_1(t) \cdot thp_{in}(t)}{ACR(t) \cdot T_i} dt + \int_0^{T_i} \frac{Q_2(t) \cdot ACR(t)}{n_1 \cdot C \cdot T_i} dt$$

A cause des expressions déjà complexes de $Q_1(t)$, $Q_2(t)$ et $ACR(t)$, le temps moyen que passe un paquet dans les files F_1 ou F_2 ne s'exprime pas toujours aisément à l'aide une formule close. Nous donnerons donc les délais moyens de transmission d'un paquet dans chaque phase sous la forme d'un résultat numérique, calculé à partir des paramètres de la connexion que nous souhaitons analyser.

Chapitre 5

Occurrence d'une perte

Lorsqu'une perte ou un débordement a lieu dans l'ingress buffer ou au niveau du goulot d'étranglement, il s'agit de déterminer comment vont être modifiées les évolutions des variables du système (ACR, W, Q_1 et Q_2) avant et après la détection de cette perte par la source TCP. Pour cela, nous étudions les mécanismes de détection de pertes proposés dans les versions de TCP les plus couramment utilisées, et observons les conséquences d'une perte sur les différents éléments du réseau.

Nous montrons dans ce chapitre l'enchaînement des états lorsqu'une perte survient. Il va de soi qu'une analyse aussi complète que celle du chapitre 2 pourrait être menée uniquement sur la phase transitoire déclenchée par une perte de paquet. Nous ne souhaitons cependant pas expliquer de nouveau les méthodes du chapitre 2, qui s'appliquent de la même manière dans ce cas particulier.

Notons que le fait d'étudier notre système en terme de paquets nous permet de modéliser de manière cohérente une connexion TCP sur ATM. En effet, même si ce sont des cellules qui sont effectivement perdues, qu'il y ait au bout du compte une ou huit cellules d'un même paquet qui soient perdues ne change rien au raisonnement car, de toutes façons, le paquet entier sera détruit. Le débit effectif ne sera donc pas altéré par le fait de raisonner en paquets. Néanmoins, en ce qui concerne le remplissage des files d'attente, nous supposerons que des mécanismes de gestion de mémoire appropriés sont implémentés au niveau des files afin que les cellules d'un paquet perdu soient éliminées de la file d'attente et ne viennent pas perturber notre calcul du remplissage des files.

5.1 Notations

Avant de commencer la description des mécanismes engendrés par l'occurrence d'une perte, nous définissons dans le tableau 5.1 les notations adoptées dans ce chapitre.

T_r	granularité du temporisateur
t_p	date de la première perte
t_s	constatation par la source du non-retour de l'acquittement du paquet perdu : la source cesse d'émettre
t_d	la source décrète que le paquet a été perdu et le retransmet
t_{s1}	instant où la file F_1 ne reçoit plus aucun paquet
t_{s2}	instant où la file F_2 ne reçoit plus aucun paquet
t_{sA}	instant où l'ACR n'évolue plus
W_i	taille de la fenêtre initiale au recommencement d'un nouveau cycle TCP

TAB. 5.1 - *Notations adoptées pour l'étude des pertes*

5.2 TCP Tahoe

Dans les deux sections suivantes, nous rappelons et modélisons les notions vues dans la partie I concernant la détection de perte avec le protocole TCP.

Sur la version originelle de TCP Tahoe, la détection d'une perte de paquet TCP s'effectue par l'intermédiaire d'un temporisateur de retransmission, nommé "Retransmission Time Out (RTO)". Ce temporisateur est utilisé et déclenché par TCP chaque fois qu'un paquet est émis. Si ce paquet n'est pas acquitté lorsque le temporisateur de retransmission est expiré, il sera retransmis.

La valeur du temporisateur est comprise entre 1 et 64 secondes. Elle est calculée actuellement à partir de l'estimation du round trip time par l'algorithme de Van Jacobson, explicité dans [JAC88]. Le RTO théorique est posé égal à la somme de la moyenne estimée du RTT et du double de sa variance, soit :

$$RTO_t = RTT + 2 \cdot Var[RTT]$$

La variance du RTT est égale à l'accroissement du RTT vu par deux acquittements consécutifs, soit $\frac{2}{thp_{out}}$. Notre modèle théorique nous permet donc de calculer la valeur du RTO_t, à partir du calcul du RTT indiqué dans le chapitre 4.

Par ailleurs, comme il serait coûteux en terme de CPU de maintenir un temporisateur actif pour chaque paquet en cours de transmission, la source ne conserve qu'un seul temporisateur. Ce temporisateur expire toutes les T_r secondes (en général, $T_r = 500$ ms). A l'expiration de de temporisateur, la source regarde les paquets dont la date d'émission augmentée du RTO_t est inférieure à la date courante, et qui auraient donc déjà dû être acquittés. La durée de temporisation effective peut donc être modélisée comme une variable uniformément répartie entre RTO_t et $T_r \cdot \lceil \frac{RTO_t}{T_r} \rceil$. Comme nous considérons essentiellement des valeurs moyennes, le temps nécessaire à la détection d'une perte de paquet depuis l'instant où ce paquet a été émis est donc égal à :

$$RTO = \frac{RTO_t + T_r \cdot \lceil \frac{RTO_t}{T_r} \rceil}{2}$$

Le round trip time pouvant varier en fonction de la charge du réseau, la valeur du temporisateur est modifiée de la même manière, et double à chaque retransmission.

5.3 TCP Reno : Fast Retransmit et Fast Recovery

A l'origine, le mécanisme du "fast retransmit" fut conçu dans le cadre de la version TCP Tahoe. Il est fondé sur le principe d'acquittement des paquets par la destination. Lorsque la destination TCP reçoit correctement "a" paquets, elle les acquitte en renvoyant à la source un numéro d'acquittement égal au numéro de séquence du prochain paquet attendu. Si elle reçoit un paquet hors séquence, elle envoie un acquittement de numéro identique au paquet précédent ; on parle alors "d'acquittement dupliqué". Afin d'éviter une attente trop longue pour retransmettre un paquet supposé perdu, on décrète la retransmission d'un paquet lorsque la source reçoit, de la part de la destination, quatre acquittements du même paquet, lui indiquant ainsi qu'elle attend toujours le paquet suivant. A cet instant, ce paquet est décrété perdu et la source le retransmet. C'est le principe du "fast retransmit". Au même instant, la fenêtre de congestion W est ramenée à la valeur $\frac{W}{2}+3$ paquets, et augmentée d'un paquet à chaque réception d'un nouvel acquittement dupliqué. A la première réception d'un acquittement non dupliqué, la fenêtre de congestion retombe à la valeur $\frac{W}{2}$. C'est le principe du "fast recovery", inclus à partir de la version TCP-Reno.

Dans notre modèle, nous avons étudié et implémenté 2 solutions :

- un modèle de détection de perte fondé sur les mécanismes de "fast retransmit" et "fast recovery" ;
- un modèle fondé sur un temporisateur de retransmission, calculé à partir de l'algorithme de Van Jacobson.

En fonction de ces deux propositions, nous étudions à présent les évènements majeurs qui se produisent lorsqu'une perte survient dans l'une ou l'autre des deux files d'attente de notre modèle.

5.4 Chronologie d'une perte

Au moment de la perte, le système évolue dans l'un des états décrits dans le chapitre 2. L'une des deux files au moins est non vide, proche de son seuil de saturation. La perte survient à l'instant t_p, lorsque la capacité d'un buffer est dépassée. Au moment où cette perte a lieu, nous n'observons aucun changement au niveau de la source ou de la destination, car rien ne leur indique qu'un paquet a été perdu. Il faut attendre l'instant t_s pour que la source constate qu'un paquet n'est pas encore parvenu à la destination, ou tout au moins que ce paquet n'a pas encore été acquitté. Cet instant représente le temps qu'aurait dû mettre l'acquittement du paquet perdu pour revenir à la source. Ce temps correspond au temps de passage dans les files, augmenté du temps de propagation sur les liens. Nous avons donc :

$$t_s - t_p \approx \begin{cases} n_1 \frac{Q_1+1}{ACR} + \tau_2 + \frac{Q_2+1}{C} + \tau_3 + \tau/2 & \text{si la perte a lieu dans la file } F_1 \\ \frac{Q_2+1}{C} + \tau_3 + \tau/2 & \text{si la perte a lieu dans la file } F_2 \end{cases}$$

A partir de l'instant t_s, la source TCP attend l'acquittement du paquet perdu. Tant que cet acquittement n'est pas arrivé, la source ne peut rien faire et reste donc inactive et silencieuse. La fenêtre W cesse de croître à l'instant t_s, et le débit thp_{in} devient égal à 0. La source n'envoie alors plus de paquet jusqu'à la détection de la perte (à l'instant t_d).

Nombre de paquets correctement transmis : Le nombre de paquets émis par la source TCP depuis le commencement de cette phase (à $t = 0$) et jusqu'au moment où elle cesse d'émettre, c'est-à-dire à l'instant $t = t_s$, est égal à $N_{e_i}(t_s)$ ($i \in [1,3]$), où $N_{e_i}(t_s)$ est calculé à partir des formules explicitées dans le chapitre 4, en fonction de l'état i dans lequel se trouve notre système.

Le nombre de paquets correctement envoyés par la source et reçus par la destination TCP est donc égal au nombre total de paquets envoyés par la source, $N_{e_i}(t_s)$, diminué du nombre de paquets perdus par la file F_1, égal à N_{p_1}, et du nombre de paquets perdus par la file F_2, égal à N_{p_2}. Les quantités N_{p_1} et N_{p_2} ont toutes deux été définies dans le chapitre 3. Depuis l'instant de la première perte, le nombre de paquets correctement transmis par la source, c'est-à-dire le nombre de paquets dont la source a reçu les acquittements, vaut :

$$N_i = N_{e_i} - (N_{p_1} + N_{p_2})$$

A partir de N_i et de la durée du cycle (jusqu'à l'instant t_d), nous sommes capables de calculer le débit efficace ("goodput") moyen de la connexion TCP durant ce cycle, en divisant le nombre de paquets correctement transmis par la durée de la période, c'est-à-dire la durée du cycle, sur laquelle ces paquets ont été transmis.

Remarque : Le nombre de paquets correctement transmis jusqu'à l'instant t_s a déjà été calculé dans le chapitre 4. En effet, nous avions calculé alors le nombre de paquets sortant du goulot d'étranglement, c'est-à-dire en fait le nombre de paquets arrivant effectivement à destination, puisque, d'après notre modèle, les paquets ne peuvent être perdus qu'à l'ingress buffer ou au goulot d'étranglement. Nous avons montré dans ce paragraphe comment retrouver le nombre de paquets acquittés à partir du nombre de paquets émis et du nombre de paquets perdus.

Cette méthode peut être utile lorsque seul un paquet TCP est perdu par le goulot d'étranglement. Ceci se produit si la source ABR est transparente pour le trafic TCP, c'est-à-dire si l'ABR ne régule pas le trafic TCP. Comme le flux exogène est constant et que la fenêtre TCP n'augmente que de 1 pour chaque fenêtre émise (en mode "congestion avoidance"), le goulot d'étranglement ne peut alors perdre qu'un seul paquet par fenêtre. Il est dans ce cas plus simple de calculer le débit efficace à partir du nombre de paquets émis N_{e_i}.

5.4.1 La période d'inactivité de la source

A l'instant t_s, notre connexion transite vers une phase particulière, caractérisée par une fenêtre de congestion W constante, égale à $W_0 = W(t_s)$, et un débit d'entrée des paquets TCP dans le réseau $thp_{in} = 0$. Comme nous l'avons précisé au début de ce chapitre, l'étude de cette phase est similaire aux études menées au chapitre 2.

5.4.1.1 Détection de la perte

La phase considérée se termine lorsque la perte de paquet est détectée, c'est-à-dire à l'instant :

$$t_d = \begin{cases} \frac{3}{thp_{out}(t_p)} & \text{si la perte est détectée} \\ & \text{par le "fast retransmit"} \\ RTO - RTT(t_p) & \text{si la perte est détectée} \\ & \text{par le temporisateur de retransmission} \end{cases}$$

La durée $\frac{3}{thp_{out}}$ représente le temps nécessaire aux trois acquittements dupliqués qui suivent l'acquittement du paquet manquant (arrivé à $t = 0$) pour parvenir à la source, lorsque nous utilisons le mécanisme "fast retransmit".

A l'instant t_d, la source TCP redevient active et nous recommençons un nouveau cycle TCP en prenant pour valeurs initiales les valeurs de ACR, Q_1 et Q_2 à la fin de la phase d'inactivité de la source. En ce qui concerne la valeur initiale de W au début du nouveau cycle, elle est égale à :

$$W_i = \begin{cases} W/2 & \text{si nous utilisons le "fast recovery"} \\ 1 & \text{si la perte est détectée par le temporisateur de retransmission} \end{cases}$$

Nous ne considèrons pas les 3 paquets supplémentaires rajoutés à la fenêtre dans le cas du "fast recovery". Nous redémarrons simplement un nouveau cycle TCP en mode "congestion avoidance", avec $W = W_{th}$, sans tenir compte de la phase intermédiaire d'attente de réception d'un acquittement non dupliqué. Autrement dit, nous ne tenons pas compte de la période de retransmission et de recouvrement d'erreur. Comme nous l'avons remarqué lors d'études précédentes, cette phase est suffisamment courte pour pouvoir être négligée. Précisons également que le "fast recovery" est considéré ici comme le commencement d'un nouveau cycle TCP.

5.4.1.2 Exploration des comportements des différentes variables

Nous avons vu que la fenêtre de congestion W restait inchangée durant toute la période d'inactivité de la source TCP, jusqu'au moment où la perte du paquet est détectée et où la connexion redémarre un nouveau cycle. Toutefois, les autres variables peuvent évoluer. En effet, avant que la perte ne soit détectée, plusieurs phénomènes se produisent.

Vidage de l'ingress buffer : Dès que l'ingress buffer ne reçoit plus de paquet, la file F_1 commence à se vider ou reste vide. Ceci se produit à l'instant t_{s1}, lorsque le dernier paquet envoyé par la source TCP avant que son débit thp_{in} ne devienne égal à 0, c'est-à-dire avant la phase d'inactivité de la source, parvient à l'ingress buffer. Nous avons donc :

$$t_{s1} = \tau_1$$

Avant t_{s1}, Q_1 évolue de la même manière que dans la phase précédente, c'est-à-dire que Q_1 garde la même expression que dans l'état précédent. Cette expression a été calculée au chapitre 2. Après t_{s1}, si l'ingress buffer n'est pas encore vide, Q_1 décroît au rythme suivant :

$$\frac{dQ_1(t)}{dt} = -\frac{ACR(t)}{n_1}$$

Comme nous le voyons par la suite, l'expression de l'ACR servant à calculer Q_1 est une expression classique, explicitée dans le chapitre 2, dépendant de l'état dans lequel se trouve notre connexion. A partir de t_{s1}, la file F_1 de l'ingress buffer décroît jusqu'à être vide, ou bien jusqu'à l'instant où un nouveau cycle TCP recommence (égal à t_d), auquel cas nous pouvons nous reporter à l'un des états décrits dans le chapitre 2.

Vidage du goulot d'étranglement : La file F_2 du goulot d'étranglement ne reçoit plus de paquet à partir du moment où tous les paquets envoyés par la source avant que cette dernière ne s'arrête d'émettre sont parvenus jusqu'au goulot d'étranglement. A cet instant, la file F_1 est également nécessairement vide puisque ces paquets étaient les derniers qu'elle aurait pu recevoir. L'instant auquel ces paquets arrivent à l'entrée du goulot d'étranglement est égal au temps de propagation sur les liens, plus le temps de service de la file F_1, à savoir :

$$t_{s2} = \tau_1 + \tau_2 + n_1 \frac{Q_1(\tau_1) + 1}{ACR(\tau_1)}$$

Avant t_{s2}, la file Q_2 évolue normalement, comme décrit dans le chapitre 2, en fonction de l'état dans lequel se trouve le système, c'est-à-dire notamment en fonction du remplissage de la file F_1. Après t_{s2}, dans l'hypothèse où le buffer du goulot d'étranglement est non vide, la file F_2 se met à décroître, au rythme suivant :

$$\frac{dQ_2(t)}{dt} = -(C - E)$$

La file F_2 du buffer du goulot d'étranglement va donc décroître, jusqu'à être vide, ou bien jusqu'à la détection de la perte (à l'instant t_d), auquel cas nous recommencerons un cycle TCP "normal", à partir d'un état défini par les conditions de départ et explicité dans le chapitre 2.

Fin d'évolution de l'ACR : L'ACR ne peut pas cesser d'évoluer tant que les files F_1 et F_2 ne sont pas vides, puisque nécessairement dans ce cas des cellules sont toujours en transit, rapportant de l'information à la source ABR. Cependant, lorsque les dernières cellules émises par la source TCP ont transité par l'unité de contrôle ABR, la source ABR n'a plus de cellule de données à émettre, et ne peut donc plus envoyer de cellule RM, car les cellules RM doivent être intercalées entre des cellules de données.

Ceci signifie que, lorsque les dernières cellules RM intercalées entre les dernières cellules de données émises par la source reviennent à l'unité de contrôle de la source ABR, cette même unité de contrôle ne reçoit plus d'information jusqu'au recommencement d'un nouveau cycle TCP. Cela se produit à l'instant :

$$t_{sA} = RTT - \tau_1$$

Ce temps correspond au temps de retour des dernières cellules RM à l'unité de contrôle, depuis l'envoi des derniers paquets par la source TCP. Auparavant, l'ACR évolue normalement, en suivant un des comportements décrits dans le chapitre 2. Après t_{sA}, l'ACR cesse d'évoluer et demeure constant jusqu'au recommencement d'un nouveau cycle TCP, à l'instant t_d.

Remarque : Il existe certaines conditions particulières où la source ABR peut envoyer des cellules RM sans avoir envoyé N_{rm} cellules de données. La première possibilité est d'avoir déjà envoyé M_{rm} cellules de données et qu'un temps égal à T_{rm} depuis l'envoi de la dernière cellule RM se soit écoulé. Une autre possibilité pour la source consiste à émettre des cellules RM "out-of-rate". Ces cellules ont cependant un débit très limité, égal à TCR=10 cellules/s. Nous pouvons nous reporter au document [ATM96] pour plus de détails sur ces cas particuliers que nous ne traiterons pas ici. Dans notre étude, nous regardons simplement le cas général où les cellules RM ne peuvent être envoyées qu'après N_{rm} cellules de données. S'il n'y a pas de paquet TCP, il n'y pas de cellules de données, donc pas de cellules RM ni de retour d'information à la source ABR. Le débit ACR n'évolue donc plus dès que la source TCP cesse d'émettre des paquets et que les files d'attente se sont vidées.

D'autre part, il se peut également que la connexion ABR soit rompue, à cause de l'inactivité de cette connexion. Ceci se produit si la période d'inactivité de la connexion est supérieure au paramètre ADTF, soit si :

$$ADTF \leq t_d - t_{sA}$$

Néanmoins, comme indiqué dans les hypothèses du chapitre 2, nous négligeons ce phénomène et choisissons à cet effet la valeur du paramètre ADTF supérieure à RTO.

5.5 Taille de fenêtre maximale

Dans ce paragraphe, nous calculons de manière approchée la taille maximale W_{max} que peut atteindre la fenêtre de congestion de la source TCP avant la perte d'un paquet.

Avant la perte d'un paquet TCP, les deux files F_1 et F_2 sont pleines et le nombre de paquets originaires de la source contrôlée et stockés dans les files est égal à :

$$B_1 + B_2 \frac{1}{1 + \frac{n_1 E}{ACR}}$$

W_{max} représentant le nombre de paquets non acquittés en circulation dans le réseau, nous obtenons W_{max} à partir de l'équation suivante :

$$W_{max} - \tau_1 \cdot thp_{in}(t) - \tau_2 \frac{ACR(t)}{n_1} - (\tau_3 + \tau/2) \cdot thp_{out}(t) - 2 = B_1 + B_2 \frac{1}{1 + \frac{n_1 E}{ACR}}$$

Nous posons plusieurs hypothèses pour simplifier le calcul de W_{max} et obtenir une valeur approchée de la taille maximale atteignable. Lorsque W atteint W_{max}, nous supposons que :

- F_1 et F_2 sont non vides
- la source TCP est en mode "congestion avoidance"
- ACR est égal à PCR
- $thp_{in} = thp_{out} = C \frac{PCR}{PCR + n_1 E}$

Ces hypothèses correspondent à l'état $\bar{F}_1^v \bar{F}_{2_{CA}}$. Il s'agit ici de donner une valeur maximale de W_{max}, plutôt que de donner une valeur exacte à chaque cycle, les cycles TCP n'étant pas périodiques dans le cas d'une utilisation avec le mécanisme ABR. Ceci donne donc :

$$W_{max} = 2 + (\tau_1 + \tau_3 + \tau/2) \cdot C \frac{PCR}{PCR + n_1 E} + \tau_2 \frac{PCR}{n_1} + B_1 + B_2 \frac{1}{1 + \frac{n_1 E}{PCR}} \qquad (5.5.1)$$

et

$$W_{th_{max}} = W_{max}/2 \qquad (5.5.2)$$

En reportant l'expression 3.2.2 de $Q_{2_{max}}$ dans B_2, on obtient les tailles de fenêtre maximales atteignables par W_{max} et W_{th} en fonction de B_1. Ces valeurs servent à borner nos courbes d'évolution de $W(t)$, ainsi qu'à délimiter les cycles TCP, le mode "slow start" et le mode "congestion avoidance".

Chapitre 6

Connexions TCP et ABR multiples

L'étude d'une connexion TCP sur ABR unique, avec flux exogène, est réaliste dans le sens où le flux exogène caractérise l'arrivée d'autres connexions au goulot d'étranglement et le partage du lien avec ces autres connexions. Nous nous concentrons dans ce cas simplement sur le comportement de la connexion TCP sur ABR, le reste du trafic étant modélisé par le flux exogène.

Le problème qui nous intéresse ici est la juxtaposition de plusieurs connexions TCP ou de plusieurs CV ABR sur un même lien, lorsque ces connexions et ces CV sont originaires de la même source et parviennent à la même destination. Ce cas représente la possibilité d'avoir plusieurs applications d'un même utilisateur et plusieurs utilisateurs de la même entreprise situés au même endroit et envoyant leurs données au même destinataire, par exemple à un autre site de la même entreprise. Leurs données suivent le même chemin et rencontrent par conséquent le même délai de transmission aller-retour. Nous souhaitons étudier le comportement de chaque connexion TCP ou de chaque CV ABR ouvert par ces utilisateurs ; l'intérêt majeur est que l'opérateur maîtrise de manière identique chacune de ces connexions puisqu'elles proviennent du même réseau local.

6.1 Etude de plusieurs connexions TCP synchrones

Nous considérons n connexions TCP synchrones partageant un même lien. Nous modélisons ainsi le cas d'un utilisateur ouvrant plusieurs connexions TCP sur sa station de travail. Le terme "synchrone" signifie que ces n connexions ont des cycles TCP synchronisés, i.e démarrant et finissant aux mêmes instants. Ce phénomène n'est pas irréaliste car, plus particulièrement dans le cas d'ATM, les pertes dans les buffers sont souvent multiples, c'est-à-dire que toutes les connexions TCP risquent de perdre au moins un paquet au même instant. Si cela se produit, toutes les connexions recommencent un nouveau cycle TCP en même temps ; c'est pour cela que nous parlons de connexions TCP "synchrones". Nous représentons sur la figure 6.1 le schéma de ces n connexions TCP. Les expériences réalisées par simulation dans la partie III permettront de valider la pertinence de cette analyse pour des connexions non synchronisées.

L'objectif de la présente étude est de déterminer en quoi le partage de la capacité du lien entre n connexions va affecter le comportement de chacune des connexions, en particulier l'évolution de la fenêtre W et du débit d'entrée des paquets TCP dans le réseau, à savoir

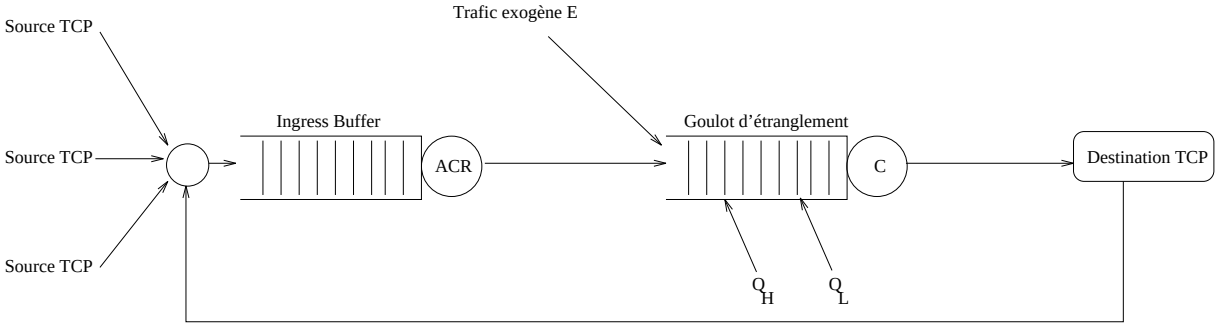


FIG. 6.1 - *Modélisation de n connexions TCP synchrones avec flux exogène*

thp_{in}. Précisons que dans le cas de multiples connexions TCP synchronisées, les valeurs des variables W, thp_{in} et thp_{out} pour chacune des connexions sont identiques. Nous montrons cependant comment les évolutions de ces variables diffèrent de l'étude d'une connexion TCP unique.

6.1.1 Files d'attente vides

Lorsque ni l'ingress buffer, ni le goulot d'étranglement ne sont saturés, l'évolution de chacune des n fenêtres est identique au cas d'une seule connexion TCP : la fenêtre TCP croît normalement puisque les paquets ne sont pas retardés dans les files. Le débit ACR croît quant à lui n fois plus rapidement puisqu'il y a n fois plus de paquets envoyés, donc n fois plus de cellules de données et par conséquent n fois plus de cellules RM qui reviennent à la source durant la même période. Par le même raisonnement que celui de la section 2.4, nous avons alors :

$$\frac{dACR(t)}{dt} = n_1 \frac{n \cdot W(t-\tau)}{T \cdot N_{rm}} RIF \cdot PCR$$

Le reste des variables reste inchangé par rapport à l'étude d'une seule connexion. Les transitions s'effectuent lorsque la capacité de la ligne est atteinte, c'est-à-dire lorsque $\min\{\frac{ACR}{n_1 \cdot n}, \frac{C-E}{n}\}$ est atteint par thp_{in}. Dans le cas de n connexions TCP synchronisées, la saturation est donc atteinte beaucoup plus rapidement, ce qui signifie qu'au moment de la transition, la fenêtre W est plus petite que dans le cas d'une seule connexion. Cette remarque a un impact important en mode congestion avoidance. En effet, d'après la formule 2.2.5, nous avons :

$$\frac{dW(t)}{dt} = \frac{thp_{out}}{aW} \text{ en mode “congestion avoidance”}$$

Nous déduisons de cette équation que plus la fenêtre W est réduite, plus la vitesse de croissance de W sera importante en mode “congestion avoidance”.

6.1.2 Etat saturé

Lorsque l'un au moins des deux buffers est non vide, c'est-à-dire lorsque nous sommes dans un état saturé, la principale valeur modifiée est le débit thp_{out} d'arrivée des paquets de chaque connexion TCP à destination.

En effet, le débit de sortie "global", que nous notons THP_{OUT}, des paquets originaires de l'ensemble des connexions TCP considérées reste égal au débit thp_{out}, calculé dans la section 2.4 pour l'étude d'une connexion TCP sur ABR unique. Nous en déduisons que le débit thp_{out} par connexion vaut :

$$thp_{out_i} = \frac{THP_{OUT}}{n} \quad \text{où } i \text{ est l'identifiant de la connexion}$$

Ceci modifie l'expression de thp_{in} et de W pour chaque connexion TCP. En revanche, le débit de retour des cellules RM est le même que dans le cas d'une seule connexion. En effet, le lien est occupé par des paquets issus de diverses connexions, mais qui transitent par le même CV ABR. Le débit de retour des cellules RM est donc toujours égal à $n_1 THP_{OUT}$ et l'expression d'ACR ne change pas.

En ce qui concerne le remplissage des deux files d'attente, nous pouvons déduire les expressions de Q_1 et Q_2 à partir des expressions de thp_{in} et d'ACR, elles-mêmes déduites de thp_{out}. Ainsi, lorsque n connexions TCP affluent sur l'ingress buffer, nous avons :

$$\begin{aligned}
\frac{dQ_1(t)}{dt} &= THP_{IN} - \frac{ACR}{n_1} \\
&= THP_{OUT}\left(1 + \frac{1}{a}\frac{dW}{dack}\right) - \frac{ACR}{n_1} \\
&= THP_{OUT}\left(1 + \frac{1}{aW}\right) - \frac{ACR}{n_1} \text{ en mode "congestion avoidance"}
\end{aligned} \tag{6.1.1}$$

Comme les fenêtres de congestion de chaque connexion TCP sont moins larges que dans le cas d'une seule connexion, le remplissage de F_1 est plus rapide en mode "congestion avoidance". En revanche, en ce qui concerne le remplissage de F_2, si F_1 est non vide et que l'ABR joue donc son rôle de protecteur du réseau, nous avons :

$$\frac{dQ_2(t)}{dt} = \frac{ACR}{n_1} - (C - E)$$

Comme ACR suit la même progression que dans le cas d'une seule connexion, l'envoi des paquets TCP de issus des n connexions est régulé par l'unité de contrôle ABR de manière à ne pas saturer le goulot d'étranglement. L'occupation du buffer du goulot d'étranglement est donc approximativement constante, qu'il y ait une ou plusieurs connexions sur le même lien.

6.2 Etude de plusieurs circuits virtuels ABR

Nous considérons ici la possibilité d'avoir non seulement plusieurs applications TCP sur la même station de travail, mais également la possibilité que plusieurs utilisateurs du même site, disposant tous d'une carte ATM avec une unité de contrôle ABR, communiquent avec la même entité distante. Ceci peut se produire par exemple lorsque plusieurs utilisateurs du CNET de Sophia Antipolis ouvrent une application "Web Browser" et une application FTP vers le CNET d'Issy-les-Moulineaux. Nous disposons alors de m circuits virtuels ABR représentant les multiples utilisateurs qui ont chacun ouvert n applications TCP. Ce type de configuration est modélisé par le schéma 6.2.

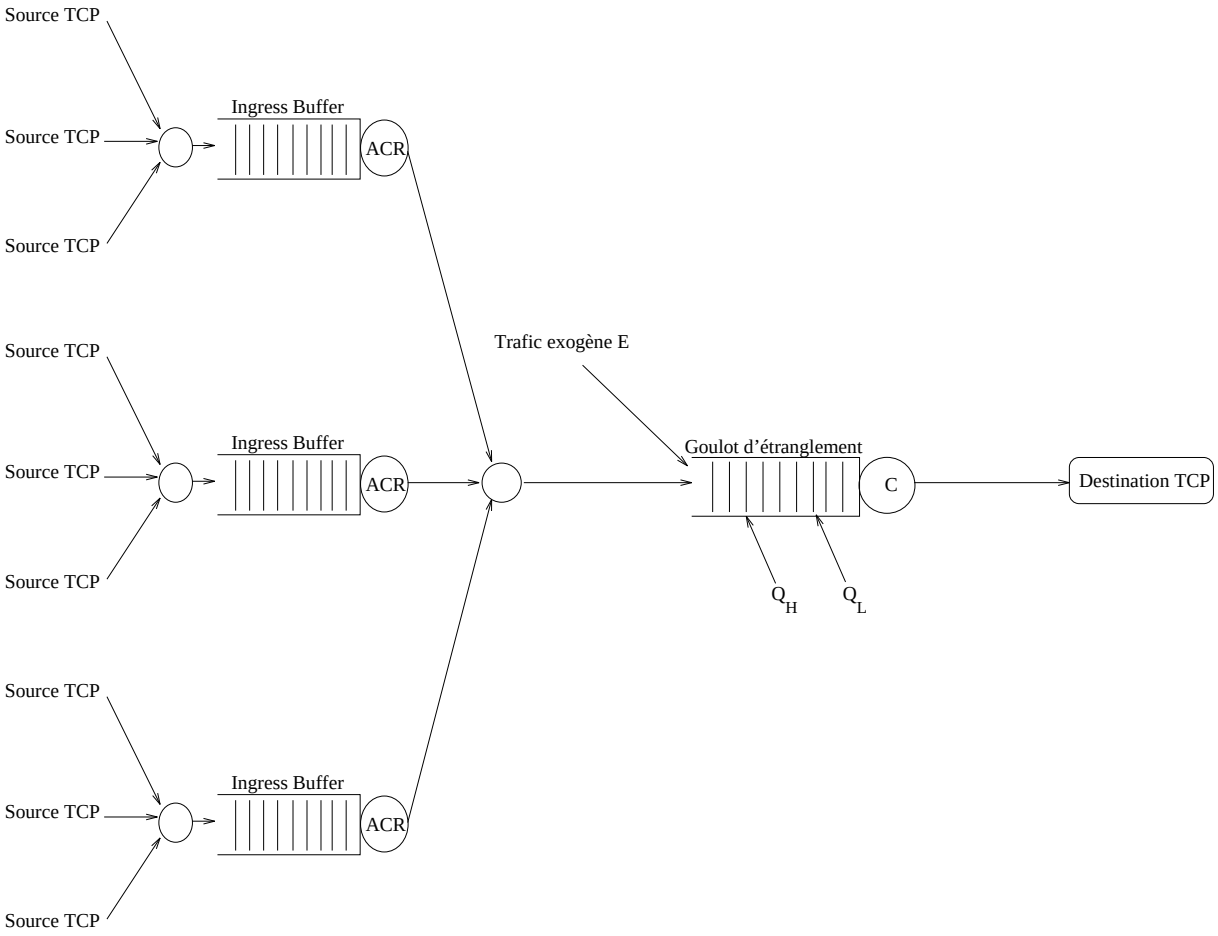


FIG. 6.2 - *Modélisation de m circuits ABR et n connexions TCP synchrones avec flux exogène*

Nous étudions de la même manière en quoi les évolutions des diverses variables différent de l'étude d'une seule connexion TCP sur un seul CV ABR d'une part, et de l'étude de plusieurs connexions TCP sur un seul CV ABR d'autre part.

6.2.1 Les deux files sont vides

Lorsque les ingress buffers et le goulot d'étranglement sont vides, toutes les fenêtres TCP croissent normalement, le retour des acquittements n'étant pas ralenti. Les expressions de

W, thp_{in} et thp_{out} pour chacune des n connexions TCP sont donc les mêmes que dans le cas d'une seule connexion TCP. De même, le débit de retour des cellules RM n'est pas influencé par le nombre de circuits ABR. En revanche, comme dans le cas de plusieurs connexions TCP sur un seul CV, l'ACR croît plus vite si le nombre de connexions TCP gérées par la même source ABR augmente. S'il y a n connexions TCP synchrones, l'ACR augmente n fois plus rapidement. Nous pouvons alors nous reporter à l'expression de $\frac{dACR}{dt}$ indiquée dans la section précédente.

La transition s'effectue lorsque la capacité de la ligne est atteinte, c'est-à-dire lorsque le débit d'entrée des paquets dans le réseau atteint $\min\{\frac{ACR}{n_1 \cdot n}, \frac{C-E}{nm}\}$ pour chaque connexion. Cela signifie toujours qu'au moment de la transition, les fenêtres TCP seront plus petites que dans le cas d'une seule connexion.

6.2.2 Saturation des ingress buffers

Comme les sources TCP sont synchronisées, les buffers se remplissent au même instant. Lorsque les ingress buffers sont non vides, le retour des acquittements des paquets de chaque connexion TCP s'effectue n fois moins rapidement, puisque le buffer est partagé par n connexion TCP. Par conséquent, le nouveau débit thp_{out} de chaque connexion vaut :

$$thp_{out_i} = \frac{THP_{OUT}}{n}$$

Nous retrouvons la valeur indiquée dans la section précédente. Cette valeur de thp_{out} permet de calculer le débit thp_{in} et la taille W de la fenêtre pour chaque connexion. Le nombre de connexions ABR n'intervient pas dans le calcul de thp_{in}, thp_{out} et W, puisque la file F_2 est vide et que l'ingress buffer est placé avant l'entrée dans le réseau. En revanche, la file F_2 est saturée plus rapidement, puisque nous avons m CV ABR qui affluent sur le goulot d'étranglement. La saturation de F_2 se produit lorsque :

$$\frac{ACR}{n_1} = \frac{C-E}{m}$$

6.2.3 Saturation du goulot d'étranglement

Lorsque le goulot d'étranglement est non vide, le retour des cellules RM induisant l'évolution d'ACR est indépendant des n connexions TCP partagées par le même CV ABR. Pour chacun des m CV ABR, ce retour s'effectue au débit $n_1 \frac{THP_{OUT}}{m \cdot N_{rm}}$, soit m fois plus lentement que dans le cas d'une seule connexion TCP sur un seul CV ABR. L'ACR croît donc ou décroît m fois plus lentement que lors de l'étude de la section 2.4. Ce phénomène compense l'augmentation des sources ABR et illustre le partage équitable de la bande passante.

En ce qui concerne le taux d'arrivée des paquets de chaque connexion TCP à destination, il est divisé non seulement par le nombre de CV ABR partageant le goulot d'étranglement, mais également par le nombre de connexions TCP partageant le même CV ABR. Nous avons alors, pour chaque connexion TCP :

$$thp_{out_i} = \frac{THP_{OUT}}{n \cdot m}$$

Nous pouvons reprendre ici la même discussion que celle menée au paragraphe 6.1. Dès que les buffers sont saturés, le débit de retour des acquittements de chaque connexion est effectivement plus lent que dans le cas d'une seule connexion. Cependant, comme les transitions vers des états saturés se sont produites plus tôt, les tailles de fenêtres sont plus petites. De ce fait, en mode "congestion avoidance", la somme des débits thp_{in} de chaque connexion est plus grande que pour une seule connexion (cf équation 6.1.1). Le premier buffer qui doit absorber les paquets issus des sources TCP se remplit donc plus rapidement. Mais, si ce buffer est le buffer d'une unité de contrôle ABR dont le débit ACR suit une évolution plus progressive qui est fonction du nombre de connexions actives se partageant le lien, cette unité de contrôle régule alors le flux des paquets TCP afin de ne pas saturer le réseau et de garantir, comme cela a été observé dans [OA97], un remplissage du goulot d'étranglement plus ou moins constant. Les ingress buffers jouent donc bien un rôle de tampons et absorbent le surplus de trafic TCP en protégeant le réseau, c'est-à-dire en évitant les pertes sur les supports partagés.

La deuxième partie de cette thèse aborde le problème de l'interaction entre les mécanismes de contrôle de flux de TCP et ceux présents dans le service ABR de l'ATM. Cette préoccupation est justifiée d'une part par l'importance de TCP dans le trafic actuel de l'Internet et d'autre part par l'usage croissant de l'ATM dans le cœur des réseaux. L'objectif de cette partie est donc d'offrir des éléments de réponse aux interrogations soulevées dans la première partie et de vérifier si le service ABR est adapté au transport de TCP. Un modèle analytique est développé pour soutenir cette hypothèse. Ce modèle considère la segmentation de paquets TCP et leur transport dans des cellules ATM dont le débit est régulé par les protocoles de l'ABR. Il s'appuie sur certains modèles développés dans la littérature pour l'analyse de TCP avec flux exogène d'une part, et celle de l'ABR d'autre part. Notre contribution consiste à rapprocher ces modèles pour représenter les interactions de deux mécanismes d'asservissement et tenir compte du fait que, dans ce cas, le débit d'entrée de TCP n'évolue pas de concert avec le débit d'entrée ABR. L'analyse de la dynamique de TCP sur ABR et les expressions des variables d'état sont calculées en résolvant un système d'équations différentielles. Les paramètres étudiés sont les performances moyennes d'une connexion TCP sur ABR exprimées en termes de délai de transit, de débit, de taux de pertes et de besoins mémoires. Le traitement des pertes de cellules fait l'objet d'un chapitre à part ; l'analyse de cette situation permet de mettre en évidence l'impact d'une perte sur les différentes phases du protocole TCP dans les versions Reno et Tahoe. Enfin, le modèle est étendu au cas de multiples connexions TCP multiplexées sur un ou plusieurs liens ABR.

Nous avons indiqué dans ce chapitre les grandes lignes des modifications apportées par le partage d'un lien entre plusieurs connexions TCP et plusieurs CV ATM-ABR. Les expressions des différentes variables du systèmes peuvent facilement être retrouvées à partir de ces indications et de l'étude menée au chapitre 2. Nous vérifions par ailleurs dans la partie suivante les remarques qualitatives que nous avons exprimées dans ce chapitre.

Troisième partie

Comportement et optimisation de TCP sur ABR

Cette partie présente la problématique du réglage des paramètres de connexion ABR pour un trafic TCP. A partir du modèle analytique, nous mesurons et nous optimisons les performances et la robustesse des paramètres ABR en fonction du contexte réseau. Nous montrons également les comportements et les interactions observés dans le cas de connexions TCP sur ABR.

Conformément au paragraphe 3.2 de la partie I, les objectifs que nous cherchons à atteindre pour déterminer un paramétrage ABR adéquat s'articulent autour des points suivants :

- Les sources s'adaptent rapidement à la disponibilité de la bande passante.
- Une source nouvelle parvient rapidement à un débit fort si la bande passante est disponible.
- Chaque source dispose d'une part équitable de la bande passante.
- Le taux de perte dans le réseau est faible quel que soit le comportement des sources.
- Les temps de traversée ne sont pas trop grands.

Nous montrons dans cette partie les performances que peut espérer un utilisateur ouvrant une connexion TCP sur un circuit virtuel ABR et nous proposons des conclusions générales sur le comportement de ces deux mécanismes.

L'approche adoptée ici est expérimentale, c'est-à-dire fondée sur des résultats numériques obtenus grâce au logiciel "PACTA" (Programme d'Analyse des performances d'une Connexion TCP sur ABR). Ce logiciel a été programmé à partir de l'étude analytique menée dans la partie II et des formules obtenues lors de cette analyse. "PACTA" permet d'étudier des configurations de réseaux particulières et de calculer numériquement les performances de ces réseaux : tracé des courbes d'occupation des buffers, de l'ACR et de W en fonction du temps, calcul du débit et du RTT moyens de la connexion. Le programme indique également les états traversés par la connexion.

Grâce à ce logiciel intégré à l'outil "Cytise" du CNET, l'opérateur peut chercher à optimiser par une approche empirique les paramètres RIF et RDF en fonction des objectifs précédents, et vérifier la robustesse des paramètres choisis au travers des multiples configurations étudiées. Il s'agit en effet de ne pas chercher à tout prix les meilleures performances possibles, mais plutôt de déterminer des paramètres qui permettent d'obtenir de bonnes performances dans plusieurs contextes réseaux différents.

Chapitre 1

Première approche pour le paramétrage de TCP sur ABR

Nous considérons dans un premier temps une connexion TCP sur ABR telle que celle représentée en figure 2.1 de la partie II. Nous cherchons à maximiser l'utilisation de la bande passante disponible, à déterminer le lieu des pertes (nœud d'accès ou réseau) et la taille des files d'attente. Les objectifs que nous nous fixons pour établir des règles de choix des paramètres ABR et déterminer une configuration ABR permettant d'assurer un service global performant et robuste sont ceux exprimés en introduction de cette partie.

1.1 Paramétrage théorique

L'analyse effectuée dans la partie II permet de fixer la valeur optimale de certains paramètres et simplifier d'autant la partie expérimentale. Nous rappelons dans cette section les paramètres déterminés par l'analyse.

1.1.1 Taille des seuils et des buffers

Dans le chapitre 3 de la partie II, nous avons montré comment paramétrer les seuils Q_H et Q_L grâce à la formule 3.2.4. Une fois ces seuils choisis, nous pouvons dimensionner efficacement la taille du buffer du goulot d'étranglement, de manière à éviter les pertes de cellules dans le réseau. La taille du buffer est donnée par la formule 3.2.2.

Remarquons toutefois que même si nous avons déterminé un paramétrage optimal des seuils Q_L et Q_H, ceux-ci dépendent souvent des caractéristiques du commutateur et s'avèrent difficilement modifiables par l'opérateur.

1.1.2 ACR decrease time factor

Le rôle de ce paramètre est d'associer une durée de validité à la valeur calculée de ACR. Si la source n'a pas émis de cellules RM depuis une durée supérieure ou égale à $ADTF$, le débit ACR est réinitialisé à la valeur ICR. Ce phénomène se produit lors d'une perte de paquet TCP, si le paramètre $ADTF$ est inférieur au RTO.

L'inconvénient de cette réinitialisation du débit ACR est qu'elle amène la source à recommencer la progressive convergence d'ACR vers une valeur correspondant à la bande passante disponible. Nous perdons donc toute l'information acquise au cycle TCP précédent. Afin d'éviter ce problème, nous choisissons une valeur de $ADTF$ égale à la valeur maximale $10,23$ s. Cela permet de garantir que la réinitialisation du débit ACR correspond bien à une période d'inactivité de la source, et non à une perte de paquet.

1.1.3 Peak Cell Rate

Pour garantir que toute la bande passante disponible puisse être utilisée, il faut fixer le PCR au débit maximal acheminable sur le lien, i.e au taux de service du goulot d'étranglement. Ainsi, si les autres sources de trafic, prioritaires ou non, n'émettent plus, toute la bande passante pourra être utilisée par notre connexion.

1.1.4 Minimum Cell Rate

Afin d'éviter des pertes intempestives pour le flux ABR, il faut s'assurer qu'en cas de forte charge le débit ACR puisse être diminué suffisamment de manière à écouler en même temps les flux prioritaires et non prioritaires. Mais nous souhaitons également que le débit minimal soit non nul, de manière à garantir un retour d'informations, autant pour la source TCP que pour la source ABR. Dans nos expériences, nous choisissons donc pour chaque source ABR un MCR supérieur ou égal au TCR et tel que la somme des MCR augmentée du flux exogène soit strictement inférieure à la capacité du lien.

1.2 Paramétrage par défaut

Certains paramètres ne sont jamais modifiés, car le changement de leur valeur influe peu sur les performances globales. Aussi nous fixons ces paramètres à leurs valeurs par défaut indiquées dans le document [ATM96], ou bien à leurs valeurs optimales, déterminées par expérience ou grâce aux études menées dans la littérature, notamment par Raj Jain dans [FJ98].

1.2.1 Nombre de cellules de données entre les cellules RM

Dans toutes nos expériences, nous avons positionné N_{rm} à 32. Il s'agit de la valeur par défaut spécifiée dans [ATM96]. Pour un trafic normal de données, cela signifie qu'au moins 1 cellule envoyée sur 32 est une cellule RM (Forward RM cell) soit 3 % du trafic. La valeur de N_{rm} conditionne la réactivité de la source aux congestions du réseau traversé. Pour une connexion de capacité 155 Mbps, le temps séparant l'envoi de deux cellules RM est de $86.4\mu s$ alors qu'il est de $8.60ms$ pour une connexion de 1.55 Mbps. A de très hauts débits, une valeur basse de N_{rm} se traduit par une très forte réactivité de la source mais aussi par une importante surcharge du trafic liée au contrôle de flux. Dans les configurations étudiées, nous considérons la surcharge associée à $N_{rm} = 32$ comme acceptable.

La source a également la possibilité d'émettre une nouvelle cellule RM forward lorsque

seulement M_{rm} cellules de données ont pu être transmises et qu'il s'est écoulé un temps T_{rm} depuis l'envoi de la dernière cellule RM. La valeur de M_{rm} est positionnée à 2, tandis que la valeur de T_{rm} est par défaut choisie égale à 100 ms. Notons que certaines valeurs de T_{rm} et M_{rm} peuvent également entraîner une surcharge du trafic.

1.2.2 Initial Cell Rate

Au commencement d'une connexion, sans connaissance préalable sur la bande passante disponible, l'expérience préconise la prudence. Aussi nous choisissons de fixer le débit initial à $2 \times MCR$. Cette valeur conditionne la valeur de départ de l'ACR. Plus cette valeur est élevée, plus l'ACR peut mettre du temps à redescendre en-dessous de la limite de saturation du lien. En revanche, une valeur trop faible peut entraîner une trop longue sous-utilisation du lien.

Lorsque la connexion est établie, l'ICR est recalculé dès le retour de la première cellule RM qui indique le délai d'aller retour FRTT (Fixed Round Trip Time) d'une cellule. ICR vaut à cet instant :

$$ICR = \min\left\{ICR, \frac{TBE}{FRTT}\right\}$$

où TBE est le nombre de cellules que la source ABR peut transmettre avant le retour de la première cellule RM. D'après le document [FJ98], nous fixons le Transient Buffer Exposure à 8192.

1.2.3 Cutoff Decrease Factor

Le rôle des paramètres CDF et CRM est de réduire le débit ACR lorsque les cellules RM ne reviennent pas à la source, par exemple lorsque celles-ci ont été perdues ou retardées à cause d'une congestion. Les valeurs par défaut de CRM et CDF sont respectivement égales à 256 et 1/16. Lorsque C_{rm} cellules de données ont été envoyées depuis la réception de la dernière cellule RM backward, l'ACR est réduit d'au moins $CDF \times ACR$, tant que cette réduction n'entraîne pas une valeur d'ACR inférieure à MCR.

1.2.4 Paramètres de TCP

Concernant la source TCP, quelques modifications peuvent être apportées pour une utilisation plus efficace sur les réseaux ATM. En premier lieu, la valeur par défaut du MTU (Maximum Transmission Unit) est égale à 536 octets, mais on préconise souvent pour les réseaux à hauts débits une taille de MTU égale à 1460 octets afin de minimiser la proportion de l'en-tête dans les paquets. De même, la taille de fenêtre maximale d'une source TCP standard ne permet pas toujours de parvenir à la saturation des liens, ce qui génère une sous-utilisation de ces liens. Aussi, nous préconisons une taille de fenêtre maximale élargie à 1 Mo (RFC 1323). Les autres paramètres de la source TCP restent inchangés, notamment le nombre de paquets acquittés par la réception d'un acquittement est égal à 2.

Chapitre 2

Optimisation numérique et robustesse du paramétrage ABR pour une connexion TCP

Le chapitre précédent est destiné à fournir un réglage adéquat de certains paramètres de TCP et d'ABR. Un sous-ensemble de ces paramètres bénéficie d'un réglage théorique obtenu grâce à l'analyse réalisée dans la seconde partie, tandis que les autres paramètres ont été choisis égaux à leurs valeurs par défaut spécifiées dans diverses recommandations. La détermination de ces paramètres permet de fixer les bases des expériences que nous allons réaliser ici pour le réglage des paramètres RIF et RDF.

Dans le présent chapitre, nous montrons tout d'abord par des exemples numériques les comportements observés pour une connexion TCP sur ABR, et expliquons l'influence des paramètres Q_L, Q_H, RIF et RDF sur les performances de la connexion. Nous adoptons ensuite une méthode plus rigoureuse qui permet de converger efficacement vers un paramétrage meilleur, c'est-à-dire qui garantit de bonnes performances du système dans de multiples configurations, y compris dans le cas de plusieurs connexions.

Les paramètres Q_L et Q_H ayant été déterminés par l'analyse, nous nous intéressons principalement aux facteurs RIF et RDF. Le positionnement de ces paramètres revêt une importance déterminante sur les performances des connexions TCP sur ABR, comme le confirment les études [FL97] et [STA98]. De plus, ce sont les seuls paramètres influents qui restent facilement modifiables par l'opérateur.

Les études de cas présentées au début de ce chapitre mettent clairement en évidence les phases traversées par une connexion TCP sur ABR (cf figure 2.1 de la partie II) et fournissent quelques résultats pertinents en termes de performance et de dimensionnement. Afin de simplifier l'exécution du programme "PACTA", nous nous plaçons dans une situation de pire cas où le débit ACR n'est pas limité par la valeur PCR. Nous nous intéressons donc simplement à la vitesse de croissance et de décroissance d'ACR. De même, nous réalisons des expériences utilisant soit un mécanisme de détection de perte par temporisateur, soit un mécanisme de détection par acquittements dupliqués. Nous pouvons ainsi comparer les avantages et les inconvénients des deux mécanismes. L'apport de la simulation, détaillé au paragraphe 2.3, est précisément de combiner toutes les caractéristiques d'une connexion réelle et de vérifier que les comportements observés par l'analyse se retrouvent dans la réalité.

2.1 Première expérience

Les études menées dans la partie III s'étalent sur des capacités de liens variant de 2Mbit/s à 20Mbit/s, des tailles de buffer variant de 1 000 à 10 000 cellules et des round trip times variant de 5 ms à 100 ms. Pour d'autres échelles de valeurs, une nouvelle étude sera nécessaire afin de déterminer dans ce cas le paramétrage adéquat. A partir du programme "PACTA", nous présentons ici un exemple numérique d'analyse de connexion TCP sur ABR. Outre les paramètres définis dans le chapitre précédent, nous choisissons la configuration de réseau suivante :

- C = 4 Mbit/s, E = 3 Mbit/s
- RIF = 0,00390625, RDF = 0,00625
- T = 0,023 s
- $B_1 = 100$ paq ≈ 1000 cells ≈ 51 ko et $B_2 = 100$ paq ≈ 1000 cells ≈ 51 ko
- $Q_H = 90$ paq ≈ 800 cells ≈ 46 ko et $Q_L = 20$ paq ≈ 200 cells ≈ 10 ko

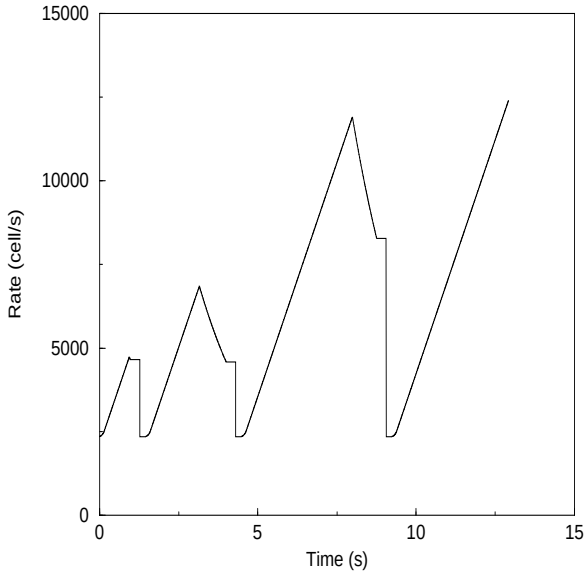
Nous souhaitons découvrir si les paramètres RIF, RDF, Q_L et Q_H sont optimaux pour ce type de configuration.

Nous présentons sur la figure 2.1 les évolutions de $ACR(t)$, $W(t)$, $Q_1(t)$ et $Q_2(t)$ en fonction du temps, et décrivons les états traversés par la connexion. A cause de la complexité des interactions entre TCP et ABR, nous ne pouvons pas déterminer de cycle périodique durant le temps d'activité de la connexion, bien qu'il en existe dans le cas de connexions TCP seul ou ABR seul. C'est pourquoi nous exprimons les formules d'évaluation de performances phase par phase et calculons numériquement le débit moyen à la fin de la période d'activité de la connexion.

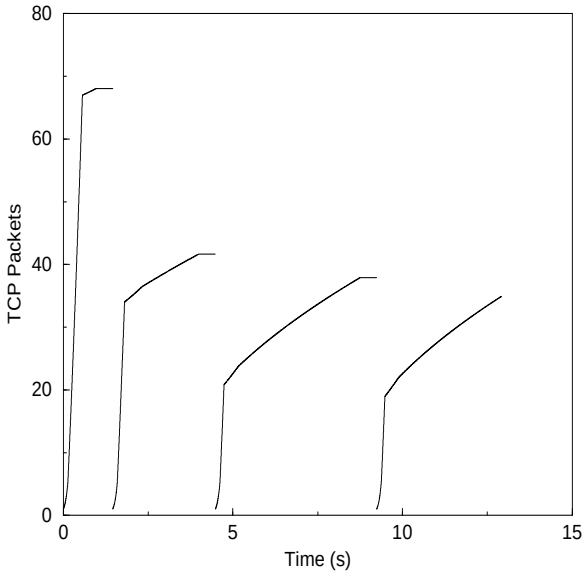
Le débit TCP moyen vaut 930 kbit/s, ce qui signifie que la bande passante allouée à la connexion TCP sur ABR, égale à 1 Mbit/s, est utilisée convenablement, en débit des pertes et des phases de démarrage des cycles TCP.

Comme nous le constatons sur ces figures, les événements majeurs que nous pouvons déterminer grâce à notre analyse sont les suivants :

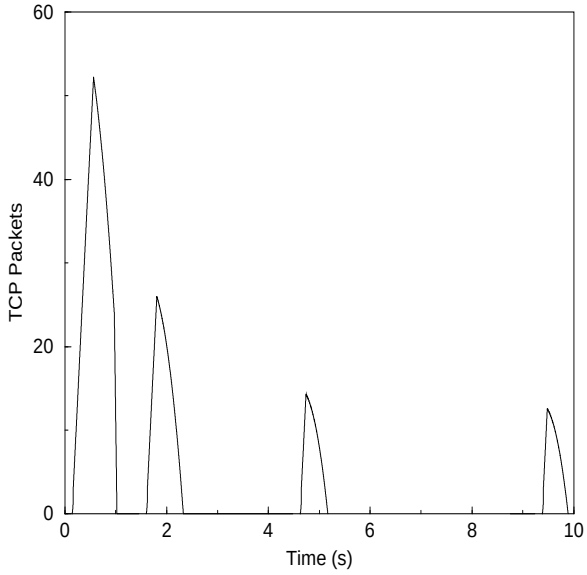
1. Au départ, les deux files d'attente sont vides, nous sommes en mode de croissance "slow start" pour TCP et le débit ACR de la source ABR croît.
2. Le débit thp_{in} croît et devient plus grand que la capacité du lien $C - E$: la file F_2 se remplit. F_1 reste vide car thp_{in} est encore plus petit que ACR.
3. Mais thp_{in} croît plus vite qu'ACR en mode "slow start" et a bientôt rejoint le taux de service de la file F_1 : F_1 se remplit.
4. En passant en mode "congestion avoidance", thp_{in} croît moins vite, la courbe de l'ACR repasse au-dessus de la courbe de thp_{in} : la file F_1 se vide.
5. Pendant ce temps, comme thp_{in} et ACR ne cessent de croître, la file F_2 grandit également et dépasse le seuil Q_H. Le goulot d'étranglement en avertit la source ABR ; l'ACR commence à décroître mais trop tard : la file F_2 est pleine et une perte survient. Un nouveau cycle TCP redémarre.



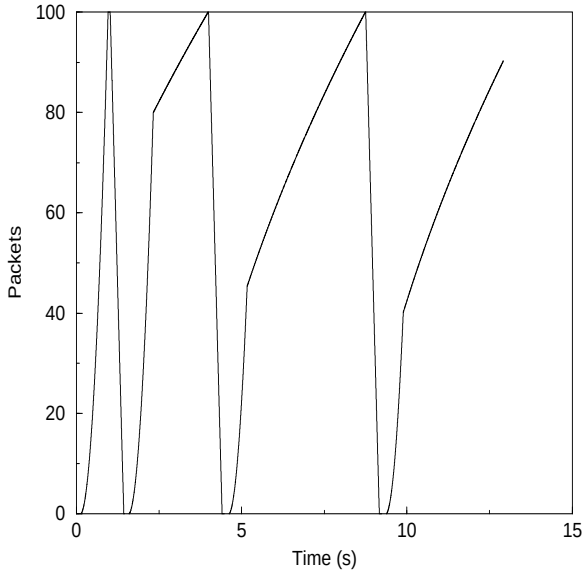
Allowed Cell Rate ACR(t)



Congestion Window W(t)



Ingress Buffer $Q_1(t)$



Goulot d'étranglement $Q_2(t)$

Fig. 2.1 - *Première expérience*

D'après notre analyse, nous sommes donc capables de décrire le comportement de la connexion et de montrer les évolutions du système au fil du temps. Nous connaissons les phénomènes qui entrent en jeu et comprenons les problèmes rencontrés en observant les états traversés durant la période d'activité de la connexion (l'enchaînement des états peut être visualisé simplement en traçant le parcours suivi sur les diagrammes de l'annexe A). Dans ce cas particulier, ceci nous amène à formuler les conclusions suivantes :

- L'ACR croît rapidement.
- Le flux TCP est peu perturbé par l'unité de contrôle ABR : tous les paquets TCP sont envoyés vers le réseau et ne restent pas bloqués au niveau de l'ingress buffer.
- Le buffer du goulot d'étranglement, dans le réseau, déborde en premier : le réseau et la source ABR ne réagissent pas suffisamment rapidement pour arrêter la congestion et empêcher le débordement.

L'opérateur souhaitera vraisemblablement protéger son réseau et déplacer la congestion au niveau de la source ABR, donc de l'ingress buffer, plutôt que de laisser déborder le goulot d'étranglement. Nous pouvons donc en déduire quelques règles pour optimiser notre réseau, et donner des recommandations concernant le choix des paramètres ABR :

- Le RIF est trop grand comparé au RDF : l'ACR croît trop rapidement et décroît trop lentement. C'est pourquoi le buffer de la file F_2 déborde. Nous devons choisir un RIF plus petit ou un RDF plus grand.
- La source ABR est avertie trop tard de la congestion dans le réseau : nous devrions donc choisir un seuil Q_H plus petit, afin que le débit ACR ait le temps de décroître suffisamment avant que la limite de F_2 ne soit atteinte.

2.2 Méthode adoptée pour l'optimisation de RIF et RDF

Suite à ce premier exemple, nous présentons maintenant les expériences réalisées afin d'améliorer RIF et RDF ; dans un premier temps, nous décrivons le plan des configurations de réseau étudiées et, dans un second temps, la méthode adoptée pour l'interprétation des résultats et l'optimisation des paramètres.

Il va de soi qu'il est impossible de déterminer de manière empirique un paramétrage qui demeure optimal dans n'importe quelle configuration. Toutefois, nous avons effectué un grand nombre d'expériences de manière à nous assurer que le paramétrage que nous proposons, s'il n'est pas forcément le meilleur dans tous les cas, est suffisamment robuste pour donner des performances très acceptables dans de multiples configurations de réseaux. A cet effet, nous avons fait varier les paramètres réseaux suivants :

- le round trip time,
- la taille des buffers,
- la capacité des liens,

- le trafic ou la charge du réseau, c'est-à-dire le flux exogène.

Pour chaque contexte réseau, nous faisons varier RIF et RDF afin de déterminer le couple RIF/RDF le plus robuste. L'ensemble des configurations étudiées n'est pas exhaustif mais l'interprétation des résultats obtenus a conduit à un paramétrage ABR que nous avons reconnu comme valable dans toutes les configurations que nous avons analysées.

2.2.1 Plan d'expérimentation

Nous avons réalisé plusieurs séries d'expériences en fonction de la capacité du lien. La durée de chacune est de 30 secondes. Pour chaque expérience, nous faisons varier cinq paramètres (le RTT, la taille des buffers, le flux exogène, RIF et RDF). Chacun de ces paramètres varie entre trois valeurs. Nous avons donc 3^5 expériences par série. Nous présentons dans les tableaux 2.1, 2.2 et 2.3 les différents paramètres de ces expériences.

L'objectif étant de guider l'opérateur dans les choix des paramètres RIF et RDF, nous fixons, comme point de départ de nos expériences, les facteurs RIF et RDF à des valeurs inférieures ou égales à leur valeur par défaut 1/16. Cette première étape dans le paramétrage de RIF et RDF a été préconisée dans [FJ98]. Le couple RIF et RDF choisi au terme des expériences doit résister aux changements de réseau sur l'échelle de valeurs considérée.

Paramètres (réseau, TCP et ABR)	**Modalité 1**	**Modalité 2**	**Modalité 3**
RIF	$1,95 \cdot 10^{-3}$	$3,91 \cdot 10^{-3}$	$7,81 \cdot 10^{-3}$
RDF	$6,25 \cdot 10^{-2}$	$3,13 \cdot 10^{-2}$	$1,56 \cdot 10^{-2}$
Source Ingress Buffer Size	1024 cells	5120 cells	10240 cells
Bottleneck Buffer Size	1024 cells	5120 cells	10240 cells
Q_H, Q_L	400 cells	2000 cells	4000 cells
Trafic exogène	1 Mbit/s	2,5 Mbit/s	4 Mbit/s
RTT	5 ms	20 ms	50 ms
MTU	1460 bytes		
PCR	5 Mbit/s		
MCR	0,5 Mbit/s		
ICR	1 Mbit/s		
N_{rm}	32 cells		
CRM	256 cells		
ADTF	10 s		
CDF	$6,25 \cdot 10^{-2}$		
M_{rm}	2 cells		
T_{rm}	0,10 s		

TAB. 2.1 - *Série d'expériences numéro 1 : $C = 5Mbit/s$*

Paramètres (réseau, TCP et ABR)	**Modalité 1**	**Modalité 2**	**Modalité 3**
RIF	$1,95 \cdot 10^{-3}$	$3,91 \cdot 10^{-3}$	$7,81 \cdot 10^{-3}$
RDF	$6,25 \cdot 10^{-2}$	$3,13 \cdot 10^{-2}$	$1,56 \cdot 10^{-2}$
Source Ingress Buffer Size	1024 cells	5120 cells	10240 cells
Bottleneck Buffer Size	1024 cells	5120 cells	10240 cells
Q_H, Q_L	400 cells	2000 cells	4000 cells
Trafic exogène	2 Mbit/s	5 Mbit/s	8 Mbit/s
RTT	5 ms	20 ms	50 ms
MTU	1460 bytes		
PCR	10 Mbit/s		
MCR	1 Mbit/s		
ICR	2 Mbit/s		
N_{rm}	32 cells		
CRM	256 cells		
ADTF	10 s		
CDF	$6,25 \cdot 10^{-2}$		
M_{rm}	2 cells		
T_{rm}	0,10 s		

TAB. 2.2 - *Série d'expériences numéro 2 : $C = 10Mbit/s$*

Paramètres (réseau, TCP et ABR)	**Modalité 1**	**Modalité 2**	**Modalité 3**
RIF	$1,95 \cdot 10^{-3}$	$3,91 \cdot 10^{-3}$	$7,81 \cdot 10^{-3}$
RDF	$6,25 \cdot 10^{-2}$	$3,13 \cdot 10^{-2}$	$1,56 \cdot 10^{-2}$
Source Ingress Buffer Size	1024 cells	5120 cells	10240 cells
Bottleneck Buffer Size	1024 cells	5120 cells	10240 cells
Q_H, Q_L	400 cells	2000 cells	4000 cells
Trafic exogène	4 Mbit/s	10 Mbit/s	16 Mbit/s
RTT	5 ms	20 ms	50 ms
MTU	1460 bytes		
PCR	20 Mbit/s		
MCR	2 Mbit/s		
ICR	4 Mbit/s		
N_{rm}	32 cells		
CRM	256 cells		
ADTF	10 s		
CDF	$6,25 \cdot 10^{-2}$		
M_{rm}	2 cells		
T_{rm}	0,10 s		

TAB. 2.3 - *Série d'expériences numéro 3 : $C = 20Mbit/s$*

Chacune de ces expériences a été réalisée grâce à l'analyseur de performances PACTA et validée par le simulateur STCP. Nous indiquerons l'intérêt du simulateur pour cette étude au paragraphe 2.3.

2.2.2 Interprétation des résultats

Pour chaque expérience, un rapport présentant les évolutions des variables ACR, W, Q_1 et Q_2 a été établi. Ce rapport indique aussi les performances obtenues dans chaque configuration, notamment :

- le nombre de paquets perdus au niveau du goulot d'étranglement et la probabilité de perte associée ;
- le nombre de paquets arrivés à destination, induisant le taux d'utilisation du lien ;
- le RTT moyen vu par un paquet.

La comparaison des couples RIF/RDF et l'obtention du couple le plus performant et le plus robuste aux changements de configuration ont été réalisées en comparant précisément les moyennes des performances obtenues dans chaque configuration. Toutefois, un couple RIF/RDF peut générer de bonnes performances moyennes tout en révélant des performances médiocres dans une configuration particulière. Nous devons donc comparer également les écarts types de ces performances. Pour associer alors les performances moyennes et leurs écarts types, nous utilisons la modélisation Signal sur Bruit (S/N) définie par le docteur Genichi Taguchi dans [WM93] et [TAG86]. Taguchi propose trois modèles de représentation du ratio signal sur bruit, selon que le signal doit être maximisé, minimisé ou ciblé. Ces trois modèles sont récapitulés dans le tableau 2.4.

Objectif	**SN Ratio**
Signal maximum	$SN_L = -10 \log \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{y_i^2} \right)$
Signal ciblé	$SN_T = 10 \log \frac{\bar{y}^2}{s^2}$
Signal minimum	$SN_S = -10 \log \left(\frac{1}{n} \sum_{i=1}^{n} y_i^2 \right)$

TAB. 2.4 - *Modélisation Signal/Bruit de Genichi Taguchi*

En tenant compte à la fois de la moyenne et de l'écart type, cette méthode nous permet de comparer efficacement la robustesse de chacun des couples RIF et RDF. Nous utilisons donc SN_L pour étudier la performance du nombre de paquets IP arrivés à destination et SN_S pour les performances du nombre de paquets IP perdus au niveau du goulot d'étranglement et du RTT.

2.2.3 Optimisation des paramètres

RIF et RDF conditionnent véritablement les écarts de performance constatés. Dans nos expériences, nous avons enregistré des écarts relatifs de performance allant de 9 % à 85 % pour les valeurs de RIF et RDF étudiées. L'importance de ces écarts confirme bien l'importance de ces facteurs sur le niveau global de performance. Dans les tableaux 2.5, 2.6 et 2.7, nous analysons les couples RIF/RDF afin de déterminer celui ou ceux qui permettent d'obtenir les performances les meilleures et les plus robustes.

C	RIF	RDF	RTT moyen	Taux d'utilisation			Pertes
				Moyenne	Ecart Type	SN_L	
5 Mbit/s	**7,81E-03**	**6,25E-02**	**0,601**	**0,846**	**0,111**	**-1,71**	**0**
	3,91E-03	6,25E-02	0,621	0,840	0,119	-1,85	0
	1,95E-03	6,25E-02	0,615	0,826	0,122	-2,02	0
	7,81E-03	**3,13E-02**	**0,604**	**0,854**	**0,0975**	**-1,56**	**0**
	3,91E-03	**3,13E-02**	**0,600**	**0,846**	**0,111**	**-1,72**	**0**
	1,95E-03	3,13E-02	0,596	0,839	0,109	-1,78	0
	7,81E-03	1,56E-02	0,593	0,855	0,0965	-1,546	$2{,}57.10^{-3}$
	3,91E-03	**1,56E-02**	**0,601**	**0,848**	**0,106**	**-1,67**	**0**
	1,95E-03	1,56E-02	0,612	0,841	0,108	-1,75	0

TAB. 2.5 - *Exploitation de la série d'expériences 1; C = 5 Mbit/s*

Remarquons dans les tableaux 2.5, 2.6 et 2.7 que le taux d'utilisation global du lien, dont la moyenne est voisine de 80 %, est un taux honorable sachant que le flux exogène varie de 20 à 80 % de la capacité du lien et qu'aucune procédure de gestion mémoire de type "EPD" n'est implémentée. D'autre part, le RTT reste toujours proportionnel à la capacité, ce qui démontre que l'écoulement des paquets se déroule correctement et assez rapidement malgré les deux files d'attente. Enfin, nous observons que la probabilité de perte au goulot d'étranglement augmente avec la capacité ce qui suggère un redimensionnement des buffers, visiblement trop petits pour absorber l'augmentation des débits d'entrée. Néanmoins, compte tenu de la variation accordée aux paramétres E, B_1, B_2 et RTT pour chaque capacité de lien, nous pouvons considérer que les pertes dans le réseau sont peu fréquentes et par conséquent que ce réseau est bien protégé par l'unité de contrôle ABR.

C	RIF	RDF	RTT moyen	Taux d'utilisation			Pertes
				Moyenne	Ecart Type	SN_L	
10 Mbit/s	**7,81E-03**	**6,25E-02**	**0,295**	**0,834**	**0,127**	**-2,00**	**0**
	3,91E-03	6,25E-02	0,298	0,822	0,143	-2,28	0
	1,95E-03	6,25E-02	0,305	0,808	0,150	-2,51	0
	7,81E-03	3,13E-02	0,299	0,830	0,132	-2.07	$1{,}64.10^{-4}$
	3,91E-03	**3,13E-02**	**0,297**	**0,830**	**0,127**	**-2,03**	**0**
	1,95E-03	3,13E-02	0,303	0,817	0,134	-2,22	0
	7,81E-03	1,56E-02	0,306	0,842	0,122	-1,87	$3{,}05.10^{-3}$
	3,91E-03	**1,56E-02**	**0,290**	**0,832**	**0,127**	**-2,00**	$\mathbf{9{,}40}.10^{-5}$
	1,95E-03	**1,56E-02**	**0,296**	**0,825**	**0,132**	**-2,11**	**0**

TAB. 2.6 - *Exploitation de la série d'expériences 2; C = 10 Mbit/s*

Pour chaque série d'expériences, nous regardons le RTT moyen et le signal sur bruit SN_L afin de déterminer le couple RIF/RDF qui procure les meilleures performances moyennes. Un des objectifs de l'unité de contrôle ABR étant également de limiter les pertes dans le réseau, nous nous assurons par ailleurs que le couple déterminé ne génère pas un taux de pertes trop important dans le goulot, en comparaison des autres couples. Compte tenu de ces critères, nous choisissons dans chacun des tableaux les couples RIF/RDF qui demeurent au-dessus de la moyenne sur tous les couples du RTT, du taux d'utilisation du

lien et du taux de pertes. Les paramètres convenables sont notés en gras dans les tableaux. Précisons que les objectifs que nous nous fixons ici peuvent être différents en fonction du réseau que nous voulons optimiser : si nous nous plaçons dans un cas où la charge du réseau est forte mais où nous souhaitons limiter les pertes sur ce réseau, nous rechercherons alors les couples RIF/RDF tels que le taux de pertes ne dépasse pas une valeur donnée. Si au contraire nous souhaitons déterminer les couples RIF/RDF garantissant un RTT faible, nous fixerons notre objectif sur le RTT moyen.

C	RIF	RDF	RTT moyen	Taux d'utilisation			Pertes
				Moyenne	Ecart Type	SN_L	
20 Mbit/s	**7,81E-03**	**6,25E-02**	**0,164**	**0,805**	**0,159**	**-2,66**	**7,71.**10^{-4}
	3,91E-03	6,25E-02	0,166	0,798	0,160	-2,72	$6{,}95.10^{-5}$
	1,95E-03	6,25E-02	0,172	0,792	0,153	-2,67	0
	7,81E-03	3,13E-02	0,163	0,811	0,147	-2.45	$1{,}42.10^{-3}$
	3,91E-03	3,13E-02	0,164	0,798	0,159	-2,73	$4{,}26.10^{-4}$
	1,95E-03	**3,13E-02**	**0,168**	**0,797**	**0,153**	**-2,64**	**0**
	7,81E-03	1,56E-02	0,169	0,808	0,147	-2,48	$6{,}09.10^{-3}$
	3,91E-03	1,56E-02	0,162	0,804	0,152	-2,58	$1{,}30.10^{-3}$
	1,95E-03	**1,56E-02**	**0,168**	**0,803**	**0,154**	**-2,63**	**0**

TAB. 2.7 - *Exploitation de la série d'expériences 3; C = 20 Mbit/s*

Les couples RIF/RDF identifiés dans ces expériences garantissent un bon niveau de performance et de robustesse sur l'ensemble des configurations de réseau étudiées. Le type de réseau que nous voulons optimiser ainsi que le critère de performance que nous souhaitons privilégier (RTT, taux d'utilisation du lien ou taux de pertes) nous conduira à sélectionner le meilleur paramétrage ABR parmi ces valeurs.

2.3 L'apport de la simulation

La configuration étudiée analytiquement est volontairement simple. L'objectif de cette étude est de déterminer le réglage des paramètres ABR qui permet de mieux utiliser la bande passante disponible. Les résultats obtenus font donc apparaître le niveau de performance et de robustesse des connexions et permettent de déduire un bon paramétrage. Ces expériences et ce réglage ont été confirmés par la simulation. Nous expliquons ici l'intérêt des simulations que nous avons menées et montrons leur adéquation avec l'analyse.

2.3.1 Intérêts et complémentarité

L'analyse permet, comme nous l'avons vu, d'expliquer les comportements observés, de dévoiler certains phénomènes atypiques, et de démontrer nos intuitions concernant les interactions de TCP sur ABR. Si, avec les vitesses de traitement actuelles des ordinateurs, la rapidité des expériences analytiques ne semblent plus déterminante comparée aux simulations, il reste cependant notoire que l'intérêt ou l'efficacité des simulations réalisées ne peuvent être obtenus qu'avec l'appui de l'analyse. En effet, il est nettement plus facile d'interpréter des résultats de simulations lorsque nous connaissons à l'avance le type de

résultats auquel nous devons nous attendre. Cette connaissance est obtenue par l'analyse. De ce fait, une étude analytique préalable permet d'isoler les comportements, de déterminer les interactions dans tel ou tel cas, avec telle ou telle version, et de prévoir l'influence de tel ou tel paramètre sur la connexion.

C'est la démarche que nous avons adoptée dans notre étude : nous nous sommes placés dans des situations de pire cas, et avons déterminé analytiquement les comportements à prévoir, théoriquement, puis numériquement grâce à notre analyseur PACTA. L'intérêt de la simulation est de confirmer par la suite ces comportements lors d'une connexion réelle où tous les paramètres sont rassemblés pour une même expérience. Il s'agit alors, dans un premier temps, de vérifier que ces comportements sont bien observés lorsque tous les paramètres et mécanismes sont combinés. Puis, dans un second temps, nous pouvons déterminer l'influence des paramètres mineurs qui n'influent que ponctuellement sur le comportement de la connexion et que, par souci de simplification, nous n'avons pas considérés dans notre analyse.

2.3.2 Le simulateur STCP

Ces étapes et ces simulations ont été réalisées à l'aide du simulateur STCP version 3.2.6. Mis au point par Sam Manthorpe au cours de son doctorat à l'EPFL (Ecole Polytechnique de Lausanne), ce simulateur de réseaux TCP sur ATM implémente le code original du protocole TCP, version Berkeley 4.4 BSD (décrit dans [STE96]), et reprend les exemples donnés dans le document [ATM96] pour l'implémentation de la capacité ABR. Il a été conçu à l'origine pour comparer les performances de TCP sur UBR et de TCP sur ABR dans le cadre d'un projet de recherche pour les PTT suisses (cf [MLB97]). Il représente environ 30000 lignes de code C réparties dans une centaine de fichiers. STCP est un logiciel disponible gratuitement et peut être téléchargé à l'URL http://lrcwww.epfl.ch/ manthorp/stcp . Il est fourni avec un manuel d'utilisation de 64 pages [MAN97b].

STCP est un simulateur géré par événements ("event driven simulator"). Il permet la simulation d'un modèle de réseau composé de différentes entités qui peuvent être des files d'attente ("queues"), des couples de stations émettrice/réceptrice ("workstations"), des unités de contrôle ABR ("ABR control units"), des sources exogènes ("background source"), des commutateurs ABR ("switch ABR") et des "leaky buckets". De nombreuses options sont envisageables au niveau des nœuds (EPD, PPD...) et des générateurs de trafic (constants, poissonniens...). Nous ne détaillerons pas toutes ces options ici et invitons le lecteur à se reporter au manuel d'utilisation [MAN97b] pour connaître toutes les possibilités du simulateur. STCP est appelé par une ligne de commande shell en spécifiant divers paramètres dont le nom du fichier de configuration. Ce dernier est un fichier texte qui décrit le modèle de réseau que l'on souhaite étudier.

Nous avons choisi ce simulateur d'une part pour son adéquation avec notre étude et d'autre part pour sa simplicité d'utilisation et d'implémentation comparée à celles d'autres simulateurs tels que YATS (cf [SAN98]). En effet, pour mener à bien le plan de simulation prévu, il a été nécessaire d'implémenter plusieurs nouveaux modules de simulation, notamment :

- une implémentation du "Relative Rate Marking" car seul l'"Explicit Rate Marking" avait été implémenté par Manthorpe ;

- une automatisation du simulateur, destinée à faciliter le lancement et le traitement de plusieurs simulations simultanées.

Ces modifications ont été facilitées par la grande clarté d'implémentation de STCP. Avant de créer de nouveaux modules, nous avons de plus vérifié que les sources existantes étaient correctes et que les simulations donnaient bien les résultats attendus. Le code de STCP permettait une validation aisée des entités implémentées.

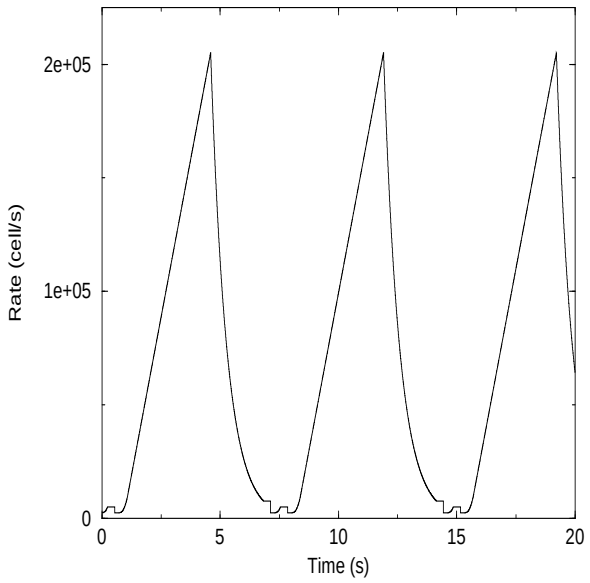
2.4 Comparaison de l'analyse avec la simulation

Après avoir vérifié le comportement du simulateur, nous avons validé notre étude analytique en comparant les résultats numériques obtenus par l'analyse avec ceux obtenus par simulation. Cette validation permet d'affirmer que l'optimisation des paramètres réalisée grâce à l'analyse est applicable dans la réalité. Nous présentons dans cette section une expérience significative montrant l'adéquation entre analyse et simulation et également les différences et compléments de l'une et de l'autre méthode. Pour visualiser un éventail plus large de ces analogies et distinctions, deux recueils rassemblent toutes les expériences réalisées grâce à l'analyse d'une part, et grâce au simulateur d'autre part.

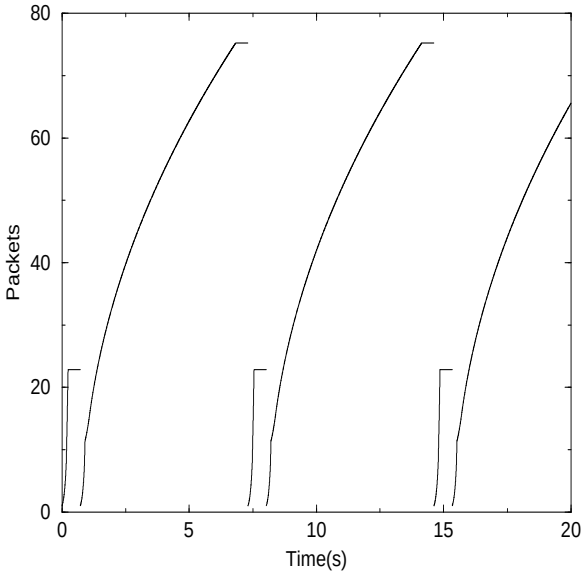
Le tableau 2.8 rassemble les paramètres de l'exemple que nous présentons. Les courbes d'évolutions obtenues grâce à l'analyseur de performances PACTA sont représentés sur la figure 2.2. Les courbes d'évolution obtenues grâce au simulateur STCP sont représentées sur la figure 2.3. La comparaison des résultats est donnée au paragraphe 2.4.1.

Paramètres (réseau, TCP et ABR)	**Valeur**
RIF	$4,0 \cdot 10^{-3}$
RDF	$5,0 \cdot 10^{-3}$
Q_H, Q_L	500 cells
Source Ingress Buffer Size	104 cells
Bottleneck Buffer Size	1024 cells
MTU	536 bytes
Capacité du lien	4 Mbit/s
PCR	20 Mbit/s
MCR	0,5 Mbit/s
ICR	1 Mbit/s
RTT à vide	0,020 s
N_{rm}	32 cells
CRM	524288 cells
ADTF	0,3 s
CDF	$6,25 \cdot 10^{-2}$
M_{rm}	2 cells
T_{rm}	0,10 s

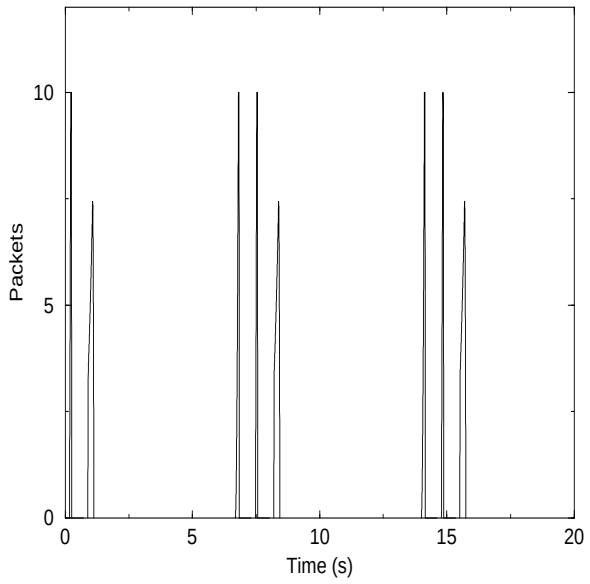
TAB. 2.8 - *Comparaison entre analyse et simulation*



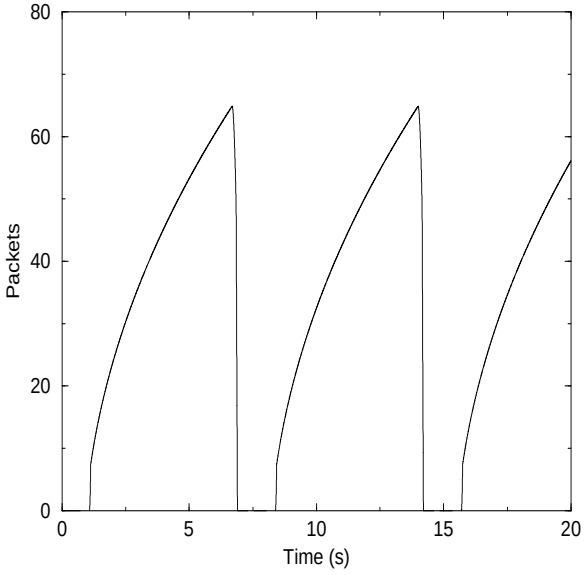
Allowed Cell Rate $ACR(\mathrm{t})$



Congestion Window $W(\mathrm{t})$

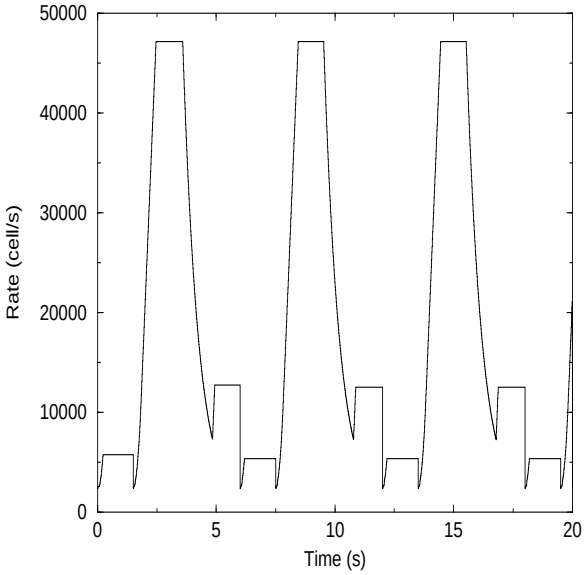


Ingress Buffer $Q_1(t)$

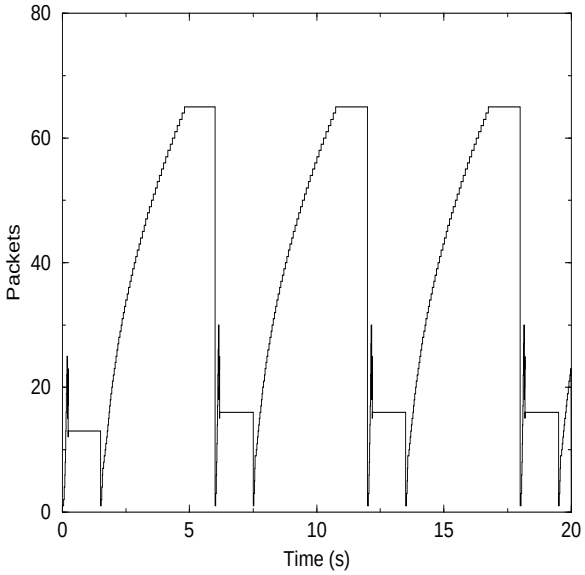


Goulot d'étranglement $Q_2(t)$

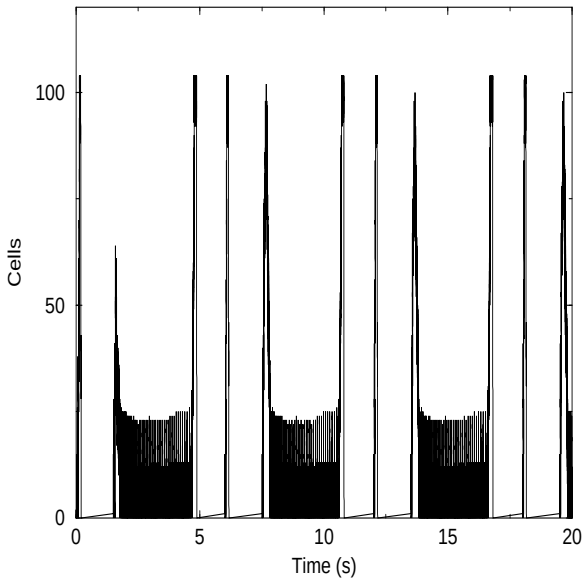
FIG. 2.2 - *Résultats obtenus par l'analyse (Analyseur PACTA)*



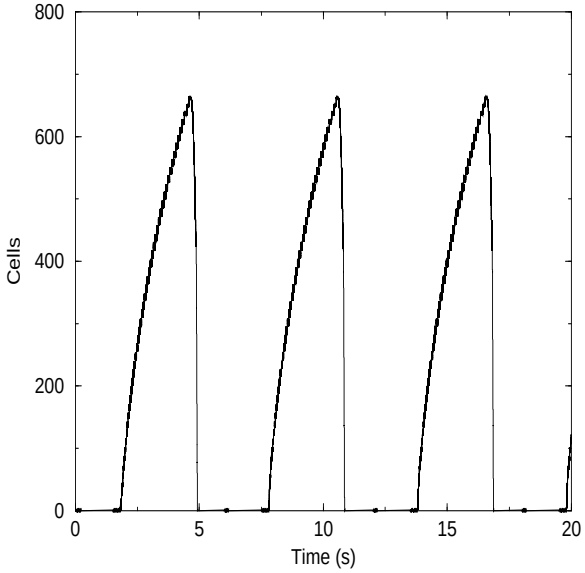
Allowed Cell Rate ACR(t)



Congestion Window W(t)



Ingress Buffer $Q_1(t)$



Goulot d'étranglement $Q_2(t)$

Fig. 2.3 - *Résultats obtenus par simulation (Simulateur STCP)*

2.4.1 Comparaison des résultats

Nous voyons bien les similitudes entre les courbes d'évolution d'ACR, W, Q_1 et Q_2. Les phases traversées sont identiques. Nous avons trois cycles, pendant lesquels nous observons une perte en "slow start" et une perte en "congestion avoidance". Les pertes se produisent toujours au niveau de l'ingress buffer. Sur l'évolution de Q_1, nous constatons trois pics groupés qui sont dûs aux deux pertes en "slow start" et en "congestion avoidance" et à un début de saturation heureusement freiné à temps grâce au passage en mode "congestion avoidance". Le débit ACR croît plus vite que le débit d'entrée TCP, si bien que les paquets sont majoritairement stockés dans le goulot d'étranglement. Il est clair que le facteur RIF est trop grand comparé au facteur RDF et que l'"élan" donné à l'ACR et surtout à Q_2 est long à freiner. Si bien que lorsque le débit ACR est enfin redescendu en-dessous de la capacité du goulot, la file d'attente F_2 décroît d'un seul coup et cette décroissance se répercute immédiatement dans l'ingress buffer qui déborde quasi instantanément. Nous comprenons ici l'intérêt d'être plus progressif dans la croissance et la décroissance d'ACR. Le passage par tous ces états se distingue aussi bien par l'analyse que par la simulation.

Les différences observées sur ces courbes proviennent majoritairement de deux phénomènes majeurs :

- Le débit ACR est limité par PCR dans le simulateur STCP tandis que dans l'analyseur il peut grandir indéfiniment. C'est une situation de pire cas que nous avons souhaité étudier comme annoncé au début de ce chapitre. C'est pourquoi nous observons une saturation du débit ACR pendant la simulation lorsque ce débit a atteint la valeur PCR; cette saturation n'est pas observée pendant l'expérience analytique. C'est aussi pour cela que le débordement de F_1 se produit plus tôt dans la simulation : partant de plus bas, l'ACR devient plus rapidement inférieur à la capacité du goulot, entraînant ainsi la file F_2 à décroître et la file F_1 à croître.

- STCP implémente le protocole TCP-Reno, tandis que la version de PACTA utilisée n'implémente pas les mécanismes "Fast Retransmit" et "Fast Recovery". De ce fait, la détection des pertes s'effectue uniquement par l'intermédiaire d'un temporisateur dans l'exemple analytique. Dans la simulation, nous observons le néfaste problème des "Successive Fast Retransmits" mentionné au paragraphe 1.2.1.3 de la partie I. Nous voyons bien ici le temps perdu par la source TCP avant de s'apercevoir que plusieurs paquets ont été perdus : plusieurs "Fast Retransmits" ont lieu pour finalement aboutir à une phase d'inactivité avant l'expiration du timer. Non seulement l'expiration du RTO n'a pas été évitée mais en plus il a été précédé de "Fast Retransmits" inutiles, et enfin le RTO a été augmenté, selon le principe de l'"Exponential Backoff". Cet exemple illustre bien l'inconvénient d'utiliser TCP-Reno sur ATM.

Ces deux différences sont voulues et peuvent facilement être corrigées dans le logiciel PACTA en limitant la valeur maximale d'ACR et en implémentant simultanément les mécanismes "fast retransmit" et "fast recovery", en complément du temporisateur de retransmission. L'intérêt encore une fois n'est pas de programmer un analyseur aussi complet qu'un simulateur mais d'expliquer et de dégrossir, grâce à l'analyse, les problèmes observés dans la réalité.

Hormis ces deux différences, nous percevons également les approximations de l'analyse fluide, au regard par exemple du remplissage de l'ingress buffer. Nous voyons que lorsque

F_2 est non vide, il y a en permanence 10 à 20 cellules, soit 1 à 2 paquets TCP contenus dans l'ingress buffer. Ce phénomène normal est dû à l'envoi de paquets TCP de manière discrète, c'est-à-dire par saccades, et non selon un flux continu. De ce fait, il se peut que, même si le débit TCP moyen est inférieur au taux de service de l'ingress buffer, certains paquets arrivent groupés et doivent attendre d'être servis un par un. Ce phénomène ne peut pas être observé par l'analyse qui ne considère que les débits moyens.

Néanmoins, cela ne constitue pas un phénomène prépondérant de l'interaction de TCP sur ABR. L'analyse fluide est donc une bonne approximation puisqu'elle met bien en avant les phases majeures d'une connexion TCP sur ABR.

2.5 Vérification de la robustesse du paramétrage pour plusieurs connexions

Les exemples numériques que nous avons étudiés précédemment s'appliquent à une seule connexion TCP sur un seul circuit virtuel ABR. Les autres connexions sont modélisées par le flux exogène. Nous souhaitons à présent valider notre étude et le paramétrage que nous avons déterminé en vérifiant que les valeurs choisies pour RIF et RDF conviennent bien dans des configurations plus réalistes, simulant de multiples connexions TCP sur plusieurs CVs ABR.

L'objectif du contrôle ABR est toujours de protéger le réseau en retardant l'envoi des cellules excédentaires. Il s'agit de garantir l'indépendance des connexions de manière à ce qu'un surcroît de trafic provenant des connexions étudiées ne vienne pas pénaliser les autres flux transitant par le réseau, c'est-à-dire par le goulot d'étranglement. En utilisant le paramétrage ABR préconisé dans le cas d'une seule connexion TCP sur ABR, nous vérifions que cet objectif est bien atteint pour plusieurs connexions TCP et plusieurs circuits virtuels ABR envoyant simultanément leurs trafics sur le réseau.

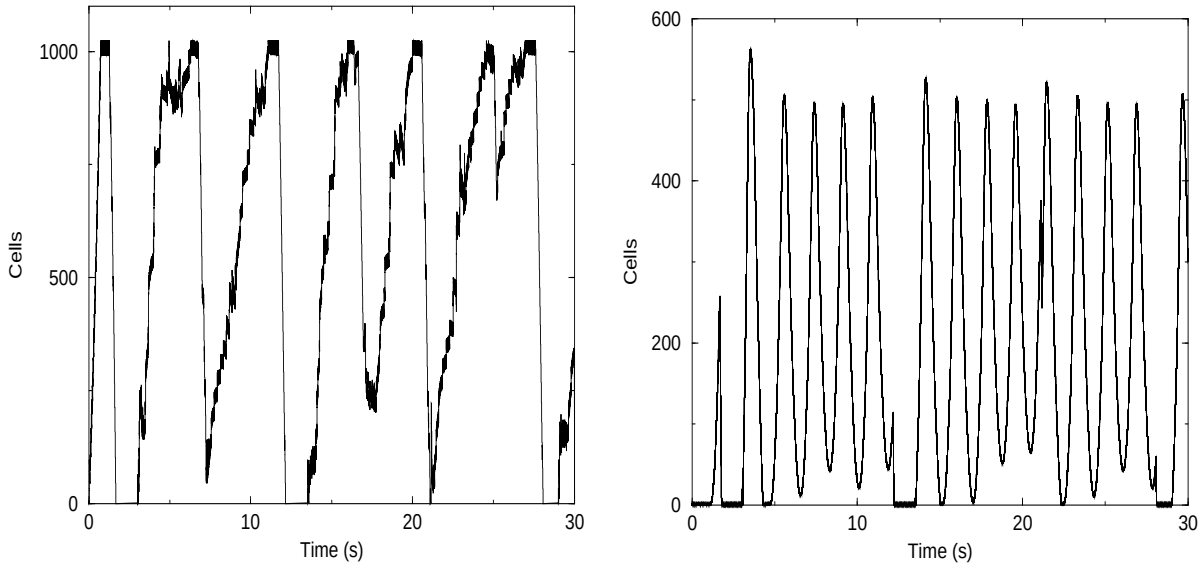
A cet effet, nous présentons ici deux exemples d'analyses d'un réseau avec plusieurs connexions et circuits virtuels. Le premier exemple est constituée de trois connexions TCP sur un seul CV ABR, selon le schéma présenté en figure 6.1 de la partie II. Les connexions TCP sont identiques. Les paramètres du système sont décrits dans le tableau 2.9. Selon la méthode proposée au paragraphe 2.2.3, nous avons choisi les paramètres RIF et RDF de manière à minimiser le taux de pertes dans le réseau ; en effet, le nombre de connexions augmentant, nous pouvons nous attendre à ce que la charge du réseau augmente et, dans ce cas, nous devons limiter les risques de congestion dans ce réseau et donc choisir un paramétrage de RIF et RDF moins agressif. Nous vérifions ici que la méthode proposée dans le cas d'une seule connexion pour dimensionner RIF et RDF conduit à un paramétrage convenable également dans le cas de plusieurs connexions.

Paramètres (réseau, TCP et ABR)	**Valeur**
RIF	$1,95 \cdot 10^{-3}$
RDF	$3,13 \cdot 10^{-2}$
Q_H, Q_L	400 cells
page suivante	

Source Ingress Buffer Size	1024 cells
Bottleneck Buffer Size	1024 cells
MTU	1460 bytes
Capacité du lien	5 Mbit/s
Trafic exogène	4 Mbit/s
PCR	5 Mbit/s
MCR	TCR
ICR	0,5 Mbit/s
RTT à vide	0,020 s
N_{rm}	32 cells
CRM	256 cells
ADTF	10 s
CDF	$6,25 \cdot 10^{-2}$
M_{rm}	2 cells
T_{rm}	0,10 s

TAB. 2.9 - *Configuration des multiples connexions TCP et circuits virtuels ABR*

Avec cette première configuration, nous obtenons les courbes d'occupation des buffers présentées en figure 2.4. Comme nous le voyons, l'unité de contrôle ABR est efficace puisque qu'aucune saturation du goulot d'étranglement n'est observée dans le réseau. Les pertes de cellules et de paquets sont bien repoussées à la périphérie du réseau, dans l'ingress buffer. Par ailleurs, le débit acheminé est bon, puisque le goulot n'est que rarement vide. L'ABR joue donc parfaitement son rôle de régulateur du trafic TCP et de protection du réseau, et ce en l'absence de tout autre mécanisme de gestion mémoire ou de lissage.

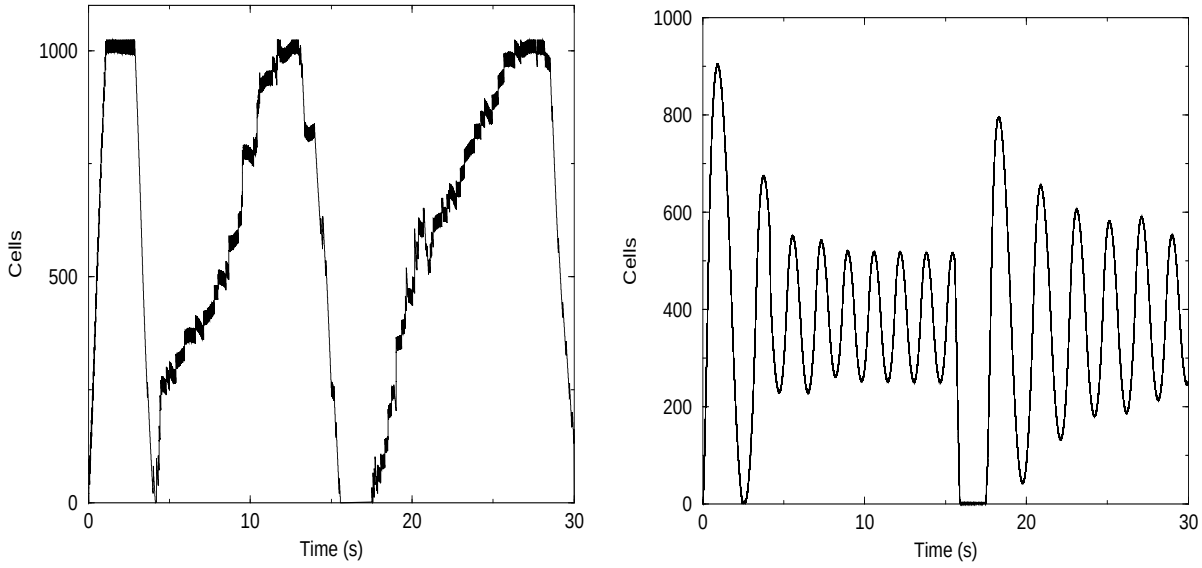


Ingress Buffer $Q_1(t)$ Goulot d'étranglement $Q_2(t)$

FIG. 2.4 - *Validation du paramétrage pour plusieurs connexions TCP sur un CV ABR*

Il en est de même dans le cas de neuf connexions TCP sur trois CVs ABR, dont la modélisation est représentée en figure 6.2 de la partie II. Les connexions TCP et circuits virtuels ABR sont également identiques et reprennent les paramètres spécifiés dans le tableau 2.9. Les résultats présentés en figure 2.5 montrent cette fois encore l'adéquation des

paramètres RIF, RDF, Q_L et Q_H dans le cas de plusieurs connexions. Le positionnement de RIF à $1,95 \cdot 10^{-3}$, de RDF à $3,13 \cdot 10^{-2}$, et de Q_L et Q_H à la valeur déterminée par l'équation 3.2.4 de la partie II correspondent bien aux objectifs fixés au départ (minimisation des pertes dans le réseau). Cette étude conforte donc la validité du choix de ces paramètres pour des configurations de réseaux différentes.



Ingress Buffer $Q_1(t)$ Goulot d'étranglement $Q_2(t)$

FIG. 2.5 - *Validation du paramétrage pour plusieurs connexions TCP sur plusieurs CVs ABR*

Apport de l'étude et conclusion

Au cours de cette thèse, nous avons recherché les mécanismes, les procédures et les services qui semblent le mieux appropriés au contrôle de trafic sur ATM. Compte tenu des caractéristiques du transfert de données dans les réseaux d'entreprises, l'efficacité de certaines solutions a été démontrée et l'opérateur dispose de plusieurs recommandations lui permettant de déployer un contrôle de flux adapté au mode de transfert ATM.

Par ailleurs, afin d'assurer le bon fonctionnement du protocole de transport de données le plus utilisé actuellement, à savoir TCP, lorsque celui-ci interagit avec les mécanismes de contrôle de trafic de l'ATM, nous avons montré l'apport de la capacité de transfert ABR. A l'aide d'un modèle analytique fluide, nous avons pu étudier précisément les interactions entre TCP et ABR et comprendre le comportement des connexions TCP sur ABR. La précision de ce modèle provient de l'analyse conjointe des mécanismes utilisés par TCP et ABR ; tandis que les précédentes études considéraient simplement le comportement de TCP ou d'ABR isolés, nous avons tenu compte ici du fait que le débit d'entrée des paquets issus de la source TCP n'évolue pas de concert avec le débit d'entrée généré par l'unité de contrôle ABR. Cette modélisation permet d'expliquer les phénomènes prédominants observés sur une connexion TCP sur ABR.

L'analyseur "PACTA", développé à partir de notre étude, fournit un moyen privilégié de prévoir et de comparer l'influence de plusieurs paramètres et mécanismes sur les performances d'une connexion TCP sur ABR. Conformément au contexte de réseau, nous pouvons, grâce à notre analyse, dimensionner correctement les paramètres TCP et ABR de manière à obtenir une configuration robuste et performante. De plus, les recherches et développements menés sur le simulateur de connexions TCP sur ATM ("STCP") apportent un supplément appréciable à notre analyse numérique en y ajoutant le dimensionnement des paramètres secondaires de l'ABR. Les modules mis au point sur ce logiciel permettent de simuler très simplement un grand nombre de configurations de réseaux et d'optimiser les performances des réseaux de demain. Les résultats obtenus par simulation sont à la fois proches et complémentaires de ceux obtenus par l'analyse.

Nous avons aussi étendu notre étude à plusieurs connexions TCP et plusieurs circuits virtuels ABR synchronisés. Nous avons montré comment les évolutions des différentes variables d'état peuvent être influencées par le nombre de connexions. Ceci nous autorise à prévoir le comportement de TCP sur ABR pour un système plus réaliste et à observer l'impact sur le dimensionnement des liens Internet où le nombre de connexions TCP est grand. Comme nous l'avons souligné, le paramétrage d'ABR proposé dans le cas d'une seule connexion nous est apparu également performant pour des connexions multiples.

L'ensemble des réalisations de notre étude, publiées dans [COS99c], [COS98], [COS99b], [COS99a] et [COS99d], montre que l'utilisation d'ABR pour le transport de flux TCP,

et plus généralement pour le transfert de données, présente des avantages indéniables et déterminants.

Un des premiers avantages du contrôle ABR est d'utiliser et de partager correctement la bande passante, c'est-à-dire d'assurer un débit honorable au protocole TCP, tout en repoussant les pertes à la périphérie du réseau. Ce point est très important car il permet de garantir que la connexion TCP saturée pourra réduire sa fenêtre et donc son débit sans provoquer la saturation du buffer backbone partagé par les autres utilisateurs. Ceux-ci sont donc protégés d'un utilisateur gourmand, ce qui ouvre la voie à une garantie de service tout en offrant un réel multiplexage statistique. Cette perspective n'est pas envisageable pour les services UBR, GFR ou VBR. En particulier, l'attrait d'ABR par rapport à la capacité UBR augmente avec le nombre de connexions supportées par le service. Plus ce nombre de connexions est important, plus l'efficacité d'ABR augmente. C'est pour cette raison que nous envisageons un service ABR partagé par différents clients ayant chacun une ou plusieurs connexions TCP, plutôt qu'un service ABR par client. L'opérateur gère alors les connexions TCP de ses utilisateurs au moyen d'une unité ABR et contrôle ainsi le trafic envoyé par les connexions TCP qui s'établissent dynamiquement sur les circuits virtuels ABR. Le service fourni par l'opérateur à l'entrée du réseau joue donc un rôle de factorisation du trafic TCP et de protection des clients les uns par rapport aux autres.

Notre étude souligne de plus que la complexité souvent reprochée à la capacité de transfert ABR est souvent un faux problème. En terme d'implémentation, les commutateurs ABR binaires, dont nous démontrons l'efficacité dans ce document, ne nécessitent pas de gestion de contexte par connexion ou par trame et sont dans tous les cas plus simples à mettre en œuvre que les procédures de gestion mémoire nécessaires au fonctionnement de TCP sur UBR. Par ailleurs, si l'avènement de cette capacité a été freiné par la difficulté de trouver un paramétrage adéquat, les résultats et conseils pratiques fournis dans ce document démontrent que rien aujourd'hui ne fait obstacle à l'utilisation d'ABR et surtout qu'il est nécessaire de l'implémenter pour pouvoir offrir au client un service adapté au trafic de données. D'une part, le paramétrage proposé dans ce document assure un fonctionnement robuste de l'ABR sur l'échelle de valeurs considérée. D'autre part, ce paramétrage est surtout utile pour des commutateurs utilisant le "Relative Rate Marking" c'est-à-dire le marquage par défaut. Pour des commutateurs plus évolués implémentant l'"Explicit Rate Marking" par exemple, il faut s'attendre à des performances encore meilleures.

Pour conclure, nous avons montré tout au long de notre travail qu'il est possible de tirer un large profit des interactions entre TCP et ABR en suivant les recommandations indiquées dans ce document. Nous sommes convaincus, au terme de notre étude et de nos lectures, que la capacité de transfert ABR est actuellement le meilleur moyen, voire le seul, de transporter correctement les flux TCP sur un réseau ATM tout en réalisant un multiplexage statistique efficace et performant. A l'heure où les opérateurs s'interrogent sur les maigres attraits des capacités UBR et VBR pour le transfert de données sur ATM et sur la manière de déployer un contrôle de trafic utile et efficace dans les réseaux RNIS-LB, il serait fort intéressant d'expérimenter sur des équipements réels l'apport du service ABR et des mécanismes ou procédures préconisés tout au long de cet ouvrage. A ce sujet, certains concurrents de France Télécom, tels que Telecom Italia, proposent déjà une offre ABR : il s'agit donc d'observer les caractéristiques de cette offre et les performances obtenues par leurs clients. D'autre part, compte tenu de la liberté laissée aux constructeurs de commutateurs, il importe également de tester leurs équipements pour connaître ceux qui conviendraient le mieux à une offre ABR de France Télécom. Enfin, nous avons montré

que le mécanisme de contrôle de trafic de l'ABR promet des performances avantageuses, même s'il n'est pas couplé à d'autres mécanismes de contrôle. Pourtant les performances attendues de certains autres mécanismes sont également attractives. Est-il alors utile pour l'opérateur de cumuler les mécanismes de contrôle ou au contraire le gain en performances est-il faible en proportion de l'investissement nécessaire ? Nous pensons notamment qu'il serait intéressant de quantifier le gain en performances apporté au transfert de données sur ABR par la segmentation virtuelle, les procédures de gestion mémoire ou les contrôleurs espaceurs.

Bibliographie

[ABN+95] E. ALTMAN, J. BOLOT, P. NAIN, D. ELOUADGHIRI, M. ERRAMDANI, P. BROWN, D. COLLANGE. Analysis of the TCP/IP flow control mechanism in high speed wide area networks. 34th IEEE Conference on Decision and Control, Décembre 1995.

[AH98] O. AIT-HELLAL. *Contrôle de flux dans les réseaux à hauts débits*. PhD thesis, UNSA, Novembre 1998.

[AHA96] O. AIT-HELLAL, E. ALTMAN. Problems in TCP Vegas and TCP Reno. De Nouvelles Architectures pour les Communications (DNAC), Décembre 1996.

[AHA97] O. AIT-HELLAL, E. ALTMAN. Analysis of TCP Vegas and TCP Reno. Montréal, Juin 1997. IEEE International Conference on Communications (ICC'97).

[AHA99] O. AIT-HELLAL, E. ALTMAN. Performance Evaluation of the Rate-Based Flow Control Mechanism for ABR Service : Generalization. New York, Mars 1999. IEEE Infocom'99.

[AHAEE99] O. AIT-HELLAL, E. ALTMAN, D. ELOUADGHIRI, M. ERRAMDANI. Performance Evaluation of the Rate-Based Flow Control Mechanism for ABR Service. *Telecommunication Systems*, 1999.

[ATM96] ATM Forum Technical Committee. *Traffic Management Specification*, Avril 1996. Draft Version 4.0.

[ATM97] ATM Forum Technical Committee. *LANE v2.0 LUNI Interface*, Juillet 1997.

[ATM98] ATM Forum Technical Committee. *Multi-Protocol Over ATM Version 1.0 MIB*, Juillet 1998.

[ATM99] ATM Forum Technical Committee. *Traffic Management Specification*, Avril 1999. Final Version 4.1.

[BA99] C. BARAKAT, E. ALTMAN. Analysis of TCP with several bottleneck nodes. Rio de Janeiro, Décembre 1999. IEEE Globecom.

[BBFa] A. BERGER, F. BONOMI, K. FENDICK. *Proposed TM Baseline Text on ABR Conformance*. ATM Forum Traffic Management Group. 95-0212R1.

[BBFb] A. BERGER, F. BONOMI, K. FENDICK. *The ABR Conformance Definition : more motivation and optimality.* ATM Forum Traffic Management Group. 95-0481.

[BC63] R. BELLMAN, K. COOKE. *Differential-Difference Equations.* Academic Press, 1963.

[BC96] P. BROWN, D. COLLANGE. Connexion TCP avec flux exogène fluide constant. Technical report, France Télécom-CNET, 1996. NT/PAA/ATR/ORE/4297.

[BC99] A.-M. BUSTOS, D. COLLANGE. Intérêt de WRED et WFQ pour une qualité de service différenciée. DE/DSE/ISE/99.10, Avril 1999.

[BCA+96] P. BROWN, D. COLLANGE, E. ALTMAN, J. BOLOT, P. NAIN, D. ELOUADGHIRI, M. ERRAMDANI. Performance of TCP/IP over the french research network : measurements and analysis. Technical report, France Télécom-CNET, Juin 1996. RP/PAA/ATR/ORE/4678.

[BES99] E. BESSON. Etude des performances d'une connexion TCP à deux goulots d'étranglement. Technical report, France Télécom-CNET, 1999.

[BP95] L.S. BRAKMO, L.L. PETERSON. TCP Vegas : End to End Congestion Avoidance on a Global Internet. *IEEE Journal on selected Areas in communications*, 13:1465–1480, 1995.

[BRS90] P. BOYER, Y. ROUAUD, M. SERVEL. Méthode et système de lissage et de contrôle de débit de communications temporelles asynchrones. Technical report, Brevet National INPI, Janvier 1990. 90/00770.

[BS96] L. BENMOHAMED, D. SU. Analysis of the rate-based traffic management proposal for ATM Networks. National Institute of Standards and Technology, 1996.

[BSG92] P. BOYER, M. SERVEL, F. GUILLEMIN. The Spacer-Controller : an efficient UPC/NPC for ATM Networks. Yokohama, 1992. ISS'92.

[BT91] P. BOYER, D. TRANCHIER. Specification of the FRP/DT. Londres, Avril 1991. ATM Network Planning and Evolution.

[BW95] A. BERGER, W. WHITT. A comparison of the sliding window and the leaky bucket. *Queueing Systems*, 12, 1995.

[BZ96] J.C.R. BENETT, H. ZHANG. Hierarchical Packet Fair Queueing Algorithms. Stanford, Août 1996. ACM SIGCOMM'96.

[CB96] H. CHEN, J. BRANDT. Performance Evaluation of EFCI switch per VC queueing. Technical report, ATM Forum Traffic Management Group, Février 1996.

[CC96] L. CERDA, O. CASALS. Improvements and Performance Study of the Conformance Definition for the ABR Service in ATM Networks. Polytechnic University of Catalonia, 1996.

[COL94] D. COLLANGE. Débit Utile pour TCP. Technical report, France Télécom-CNET, 1994. DE/ATR/01.94.

[COL96] D. COLLANGE. Performances d'une connexion ABR. Technical report, France Télécom-CNET, Août 1996.

[COL98] D. COLLANGE. Comportement et performances de TCP : cas de plusieurs connexions synchronisées. Technical report, France Télécom-CNET, Octobre 1998. NT/DSE/ISE/5694.

[COS96] J.-L. COSTEUX. Performances de TCP en présence de flux exogène et de pertes aléatoires. Technical report, France Télécom-CNET, Juin 1996. RP/PAA/ATR/ORE/4770.

[COS98] J.-L. COSTEUX. Fluid Analysis of a TCP Connection in presence of Exogenous Flow and Random Loss. Tampa, Décembre 1998. 37th IEEE Conference on Decision and Control.

[COS99a] J.-L. COSTEUX. Analysis of a TCP connection over ABR in presence of exogenous flow. Colmar, Juin 1999. International Conference on ATM.

[COS99b] J.-L. COSTEUX. Fluid analysis of a TCP connection over ABR. Edinburgh, Juin 1999. International Teletraffic Conference.

[COS99c] J.-L. COSTEUX. Fluid analysis of a TCP connection over ABR in presence of exogenous flow. *Telecommunication Systems*, 1999. Soumis à publication.

[COS99d] J.-L. COSTEUX. Fluid analysis of TCP connections over ABR VCs. Istanbul, Août 1999. Publié dans la revue "System Performance Evaluation : Methodologies and Applications", CSC Press LLC.

[CT98] Y. CHEN, J.S. TURNER. Design of a Weighted Fair Queueing Cell Scheduler for ATM Networks. Sydney, Novembre 1998. IEEE Globecom.

[DAV98] B. DAVIE. *Switching in IP networks*. Morgan Kaufmann Publishers, Mai 1998.

[DOS93] B.T. DOSHI. Deterministic Rule Based Traffic Descriptors for Broadband ISDN : Worst Case Behavior and Connection Admission Control. Houston, 1993. IEEE Globecom'93.

[FCB96] C. FENZY, D. COLLANGE, P. BROWN. Etude comparative des mécanismes de contrôle de trafic pour le service ABR. Technical report, France Télécom-CNET, Décembre 1996. DE/ATR/112.96.

[FCB98] C. FENZY, D. COLLANGE, A.-M. BUSTOS. Description de TCP : mécanismes et versions. Technical report, France Télécom-CNET, Septembre 1998. NT/DSE/ISE/5815.

[FF96] K. FALL, S. FLOYD. Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. $ftp://ftp.ee.lbl.gov/papers/sacks_v1.ps.Z$, Mars 1996.

[FJ93] S. FLOYD, V. JACOBSON. Random Early Detection gateways for congestion avoidance. In *IEEE/ACM Transactions of Networking*, volume 1, pages 397–413, Août 1993.

[FJ98] S. FAHMY, R. JAIN. Roles and Guidelines for ABR Parameter Values. http://www.cis.ohio-state.edu/ Jain, 1998. The Ohio State University, USA.

[FL97] C. FANG, A. LIN. TCP Performance in ATM Networks : ABR parameter tuning and ABR/UBR comparisons. In *IEEE SIGCON'97*, Singapour, Avril 1997.

[FLO94] S. FLOYD. TCP and Explicit Congestion Notification. *Computer Communication Review*, 24(5):8–23, Octobre 1994.

[FLO95] S. FLOYD. TCP and successive fast retransmits. ftp://ftp.ee.lbl.gov/floyd/papers/fastretrans.ps, Mai 1995.

[Fra98] France Télécom - CNET. *Evolution du réseau dorsal: introduction des contrôleurs-espaceurs MET AXD 312*, Août 1998. DE/DAC/NTR/39.

[GH96] G. GUERIN, J. HEINANEN. UBR + Service Category Definition. Technical report, ATM Forum Contribution, Décembre 1996. 96-1598.

[GOU99] C. GOULEAU. Modèle analytique fluide de l'algorithme RED avec une seule connexion TCP. DE/DSE/ISE/xx.99, Février 1999.

[GUI99] F. GUILLEMIN. *Modélisation Mathématique pour le Contrôle de Trafic dans les Réseaux Temporels Asynchrones.* PhD thesis, Université Pierre et Marie Curie Paris VI, Janvier 1999. Habilitation à diriger des recherches.

[HOE96] J.-C. HOE. Improving the Start-up Behavior of a Congestion Control Scheme for TCP. ACM Sigcomm'96, Septembre 1996.

[ITU96] ITU-T Recommendation I.371. *Traffic Control and Congestion Control in B-ISDN*, Août 1996.

[JAC88] V. JACOBSON. Congestion Avoidance and Control. In *Proceedings of the SIGCOMM'88 Symposium.* ACM Sigcomm, Août 1988.

[JAC90] V. JACOBSON. Modified tcp congestion avoidance algorithm. end2end mailing list, Avril 1990.

[JBB92] V. JACOBSON, R. BRADEN, D. BORMAN. TCP Extensions for High Performance. Technical report, RFC 1323, Mai 1992. Extension dans le RFC 1644, Juillet 1994.

[JK96] P. JOHANSSON, J. M. KARLSSON. Characteristics of an Explicit Rate ABR Algorithm. Lund Institute of Technology, Septembre 1996.

[JKG+96] R. JAIN, S. KALYANARAMAN, R. GOYAL, S. FAHMY, R. VISWANATHAN. ERICA Switch Algorithm : a Complete Description. Technical report, ATM Forum Traffic Management, Août 1996. 96-1172.

[KBC94] H.T. KUNG, T. BLACKWELL, A. CHAPMAN. Credit Update Protocol for Flow Controlled ATM Networks : Statistical Multiplexing and Adaptative Credit Allocation. University College London, Septembre 1994. Conference on Communications Architectures, Protocols and Applications.

[KG96] D. KOFMAN, M. GAGNAIRE. *Réseaux Hauts Débits*. InterEditions, 1996.

[KHBG91] H. KRONER, G. HEBUTERNE, P. BOYER, A. GRAVEY. Priority Management in ATM Switching Nodes. *IEEE Journal in Selected Areas in Communications*, 9, 1991.

[KJF+96] S. KALYANARAMAN, R. JAIN, S. FAHMY, R. GOYAL, S.-C. KIM. Buffer Requirements for TCP/IP over ABR. Août 1996.

[KL98] W.-J. KIM, B. G. LEE. The FB-RED algorithm for TCP over ATM. volume 1, pages 551–555, Sydney, Novembre 1998. Globecom'98.

[KP87] P. KARN, C. PARTRIDGE. Improving Round-Trip Time Estimates in Reliable Transport Protocols. *Computer Communication Review*, 17(5):2–7, 1987.

[LEG96] A. LEGOUT. Etude d'une connexion TCP à travers un système de 2 files d'attente. Technical report, France Télécom-CNET, Août 1996. RP/PAA/ATR/ORE/4824.

[LEM95] M. LEMERCIER. *Architecture et Evaluation d'un Mécanisme à Rejets Sélectifs Multiples*. PhD thesis, Université Paris VI, Janvier 1995.

[LF98] F. LE FAUCHEUR. IETF Multiprotocol Label Switching (MPLS) Architecture. Colmar, Juin 1998. ICATM'98.

[LM94] T. V. LAKSHMAN, U. MADHOW. Window-based congestion control for network with bandwidth delay product and random loss : a study of TCP/IP Performance. Grenoble, Juin 1994. High Performance Networking'94.

[LNO94] T. V. LAKSHMAN, A. NEIDHART, T. J. OTT. The Drop from Front Strategy in TCP and in TCP over ATM. 1994. Bell Communications Research.

[LR96] D. LAPSLEY, M. RUMSEWICZ. Improved Buffer Efficiency via the No Increase Flag in EFCI Flow Control. University of Melbourne, 1996.

[MAN97a] S. MANTHORPE. *Implications of the Transport Layer for Network Dimensioning*. PhD thesis, EPFL, 1997.

[MAN97b] S. MANTHORPE. STCP 3.2 User Manual. http://lrcwww.epfl.ch/ manthorp/stcp, Septembre 1997.

[MBLCM96] M. A. MARSAN, A. BIANCO, R. LO CIGNO, M. MUNAFO. TCP over ABR in ATM Networks with Variable Topology and Background Traffic. Politecnico di Torino, 1996.

[MG94] K. MOLDEKLEV, P. GUNNINBERG. Deadlock situations in TCP over ATM. Vancouver, Août 1994. 4th International IFIP Workshop on Protocols for High Speed Networks.

[MLB97] S. MANTHORPE, J.-Y. LE BOUDEC. A comparison of ABR and UBR to support TCP traffic. EPFL. Draft Version, 1997.

[MR98] L. MASSOULIE, J. ROBERTS. Fairness and Quality of Service for Elastic Traffic. Submitted to SIGCOMM'98, 1998.

[MRS99] L. MASSOULIE, J.W. ROBERTS, A. SIMONIAN. Analyse des mécanismes WFQ et WRED pour la garantie de qualité de service. DE/DAC/OAT/28.99, Mars 1999.

[NAB+97] P. NAIN, E. ALTMAN, J. BOLOT, D. ELOUADGHIRI, M. ERRAMDANI, P. BROWN, D. COLLANGE. Performance Modeling of TCP/IP in a Wide-Area Network. Technical report, INRIA, Mars 1997. RP 3142.

[OA97] T. J. OTT, N. AGGARWAL. TCP over ATM : ABR or UBR. Seattle, Juin 1997. ACM Sigmetrics.

[OMM98] H. OHSAKI, M. MURATA, H. MIYAHARA. Parameter tuning of Rate-Based congestion control algorithm for ABR service class in ATM networks. *International Journal of Communication Systems*, 11:103–128, 1998.

[OMS+97] E. OUBAGHA, L. MASSOULIE, A. SIMONIAN, P. BROWN, C. FENZY, D. COLLANGE. Etude comparative de deux mécanismes de contrôle de trafic pour le service ABR. Technical report, France Télécom-CNET, Février 1997. NT/DSE/ORE/5022.

[PFTV88] W. H. PRESS, B. P. FLANNERY, S. A. TEUKOLSKY, W. T. VETTERLING. *Numerical Recipes in C.* Cambridge University Press, 1988.

[RAB97] C. RABADAN. *Study of 2-store algorithms for ABR conformance.* Réunion d'experts COM13, Paris, Septembre 1997. ITU-T Q8/13 D7.

[RAB98] C. RABADAN. La capacité de transfert ATM "Available Bit Rate". Technical report, France Télécom-CNET, Février 1998. NT/CNET/5475.

[RAT90] E. RATHGEB. Policing mechanisms for ATM networks, modeling and performance comparisons. Morristown, 1990. 7th ITC Specialist Seminar.

[REK97] Y. REKHTER. Tag switching architecture overview. volume 85. IEEE Proceedings, Décembre 1997.

[RF94] A. ROMANOW, S. FLOYD. Dynamics of TCP Traffic over ATM Networks. In *Proc. ACM Sigcomm Conference*, pages 79–88, 1994.

[RIT96] M. RITTER. Analysis of Feedback-Oriented Congestion Control Mechanisms for ABR Services. Infocom, 1996.

[RIT97a] M. RITTER. Congestion Detection Methods and their Impact on the Performance of the ABR Flow Control Mechanism. Washington, Juin 1997. International Teletraffic Conference.

[RIT97b] M. RITTER. The Effect of Bottleneck Service Rate Variations on the Performance of the ABR Flow Control. Kobe, Avril 1997. Infocom.

[RIT98] S. RITZENTHALER. IP or ATM versus IP over ATM : the role of the ATM Forum and the IETF in setting standards. Colmar, Juin 1998. ICATM'98.

[RK98] C. RABADAN, F. KLAY. Un nouvel algorithme de contrôle de conformité pour la capacité de transfert "Available Bit Rate". Technical report, France Télécom - CNET, Février 1998. NT/CNET/5476.

[ROB91] J.W. ROBERTS. Performance Evaluation and Design of Multiservice Networks. *COST 224 Final Report*, 1991.

[SAN98] A. SANTOS. YATS - Yet Another Tiny Simulator (ATM Simulation). http://www.ifn.et.tu-dresden.de/TK/yats/yats.html, Janvier 1998.

[SK98] S. SHAKKOTTAI, A. KUMAR. TCP over end-to-end ABR : a study of TCP performance with end-to-end rate control and stochastic available capacity. Sydney, Novembre 1998. Globecom.

[SLCG93] M. SIDI, W.Z. LIU, I. CIDON, I. GOPAL. Congestion control through input rate regulation. *IEEE Transactions on Communications*, 41(3):471–477, Mars 1993.

[STA98] W. STALLINGS. *High Speed Networks (TCP and ATM design principles).* Prentice Hall, 1998.

[STE96] W. STEVENS. *TCP/IP Illustrated.* Addison-Wesley Publishing Company, 1996.

[STE97] W. STEVENS. *TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms*, Janvier 1997. RFC 2001.

[SZC90] S. SHENKER, L. ZHANG, D. CLARK. Some Observations on the Dynamics of a Congestion Control Algorithm. *ACM Computer Communication Review*, 20(4), Octobre 1990.

[TAG86] G. TAGUCHI. *Introduction to Quality Engineering (Designing quality into products and processes).* Asian Productivity Organization, 1986. ISBN-92-833-1083-7.

[TOU] F. TOUTAIN. Fair Queueing : Etat de l'art et proposition d'une discipline à pertes équitables. Télécom Bretagne.

[TUR86] J. TURNER. New directions in communications. *IEEE Communications Magazine*, 24:8–15, Octobre 1986.

[WM93] R. E. WALPOLE, R. H. MYERS. *Probability and Statistics for Engineers and Scientiscs.* Prentice Hall, 3ème edition, 1993.

[YH94] N. YIN, M. G. HLUCHYJ. On Closed Loop Rate Control for ATM Cell Relay Networks. 1994. Wellfleet Communications, Motorola Codex.

Annexe A

Diagrammes d'états

Nous indiquons ici, pour une connexion TCP sur ABR, les diverses transitions envisageables depuis les états F_1F_2, $\bar{F}_1F_2$, $F_1\bar{F}_2$ et $\bar{F}_1\bar{F}_2$. Ces transitions sont représentées par des diagrammes d'états. Nous distinguons les diagrammes en fonction du remplissage des deux files F_1 et F_2. Nous ne représentons pas dans ces schémas les transitions qui suivent un débordement de buffer et une perte de paquet. Les numéros indiqués pour chaque état servent à identifier les états traversés par la connexion TCP sur ABR, lors du lancement de notre programme PACTA.

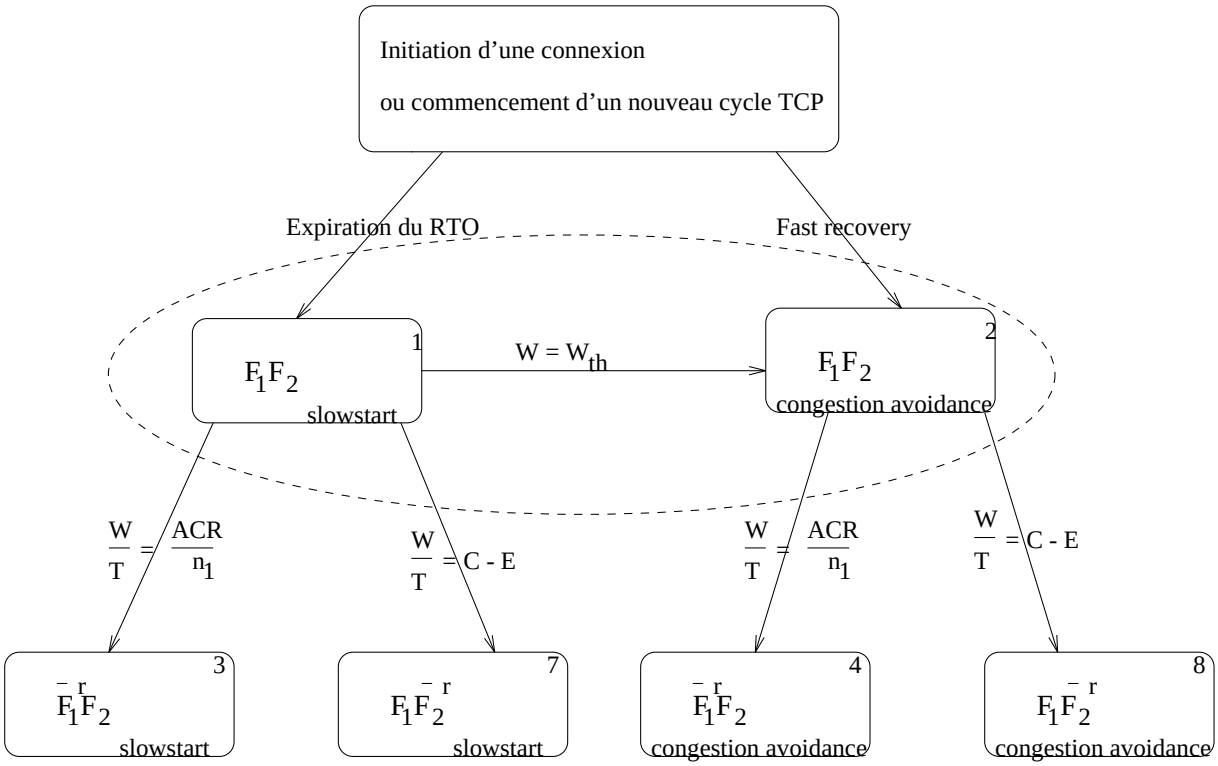


FIG. A.1 - *Diagramme d'états, lorsque F_1 et F_2 sont vides*

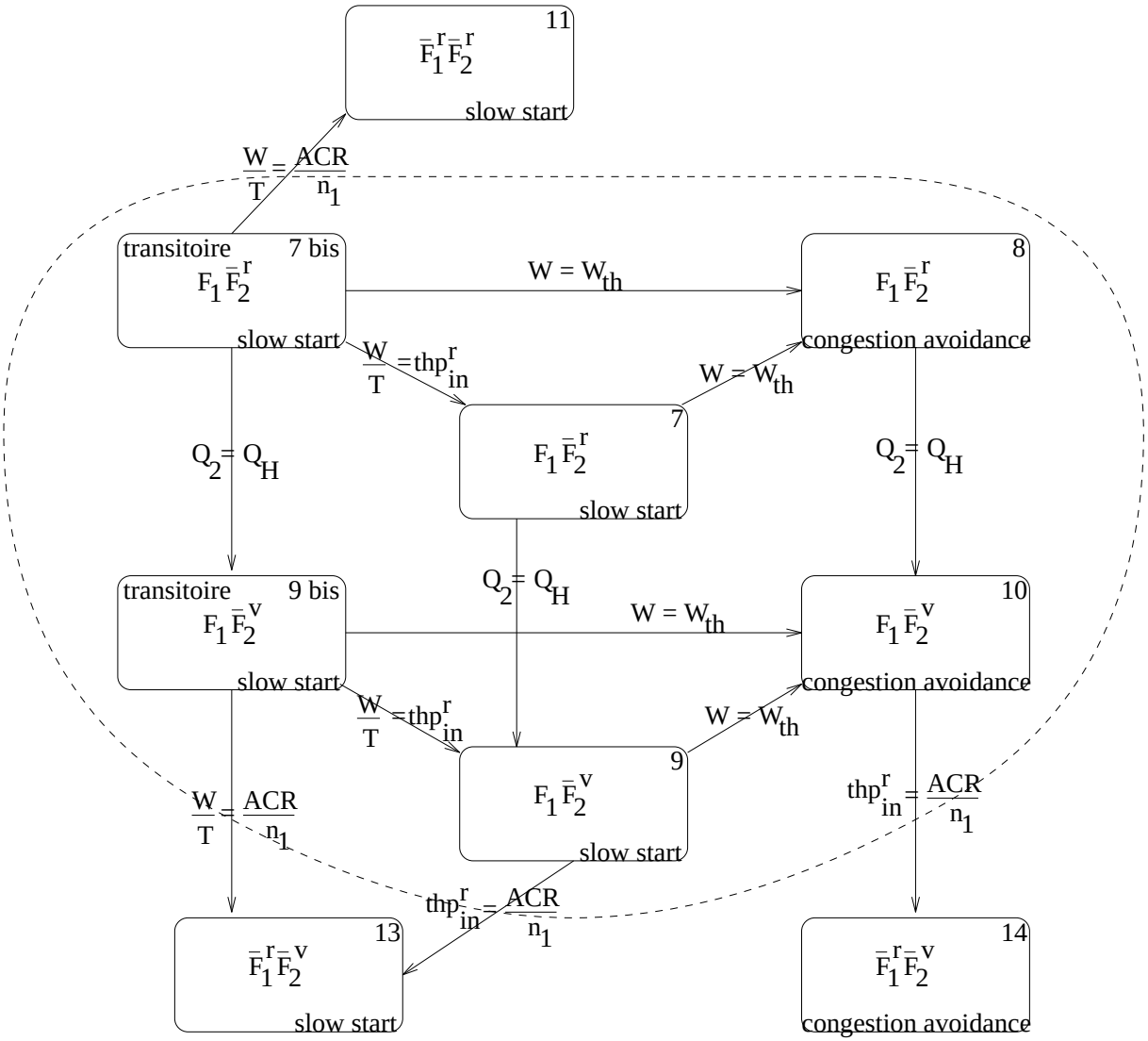


FIG. A.2 - *Diagramme d'états, lorsque F_1 est vide et F_2 non vide*

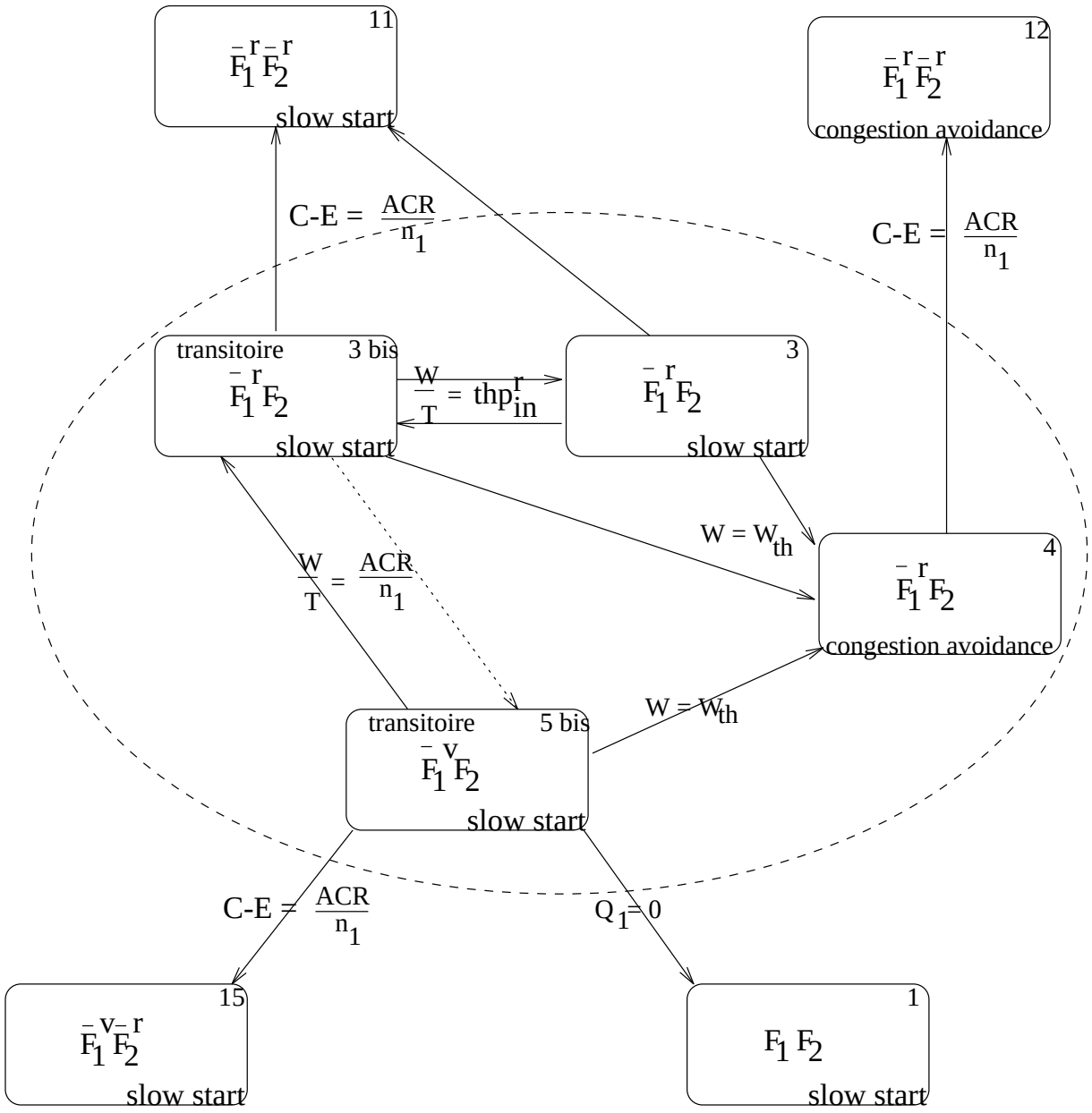


FIG. A.3 - *Diagramme d'états, lorsque F_1 est non vide et F_2 vide*

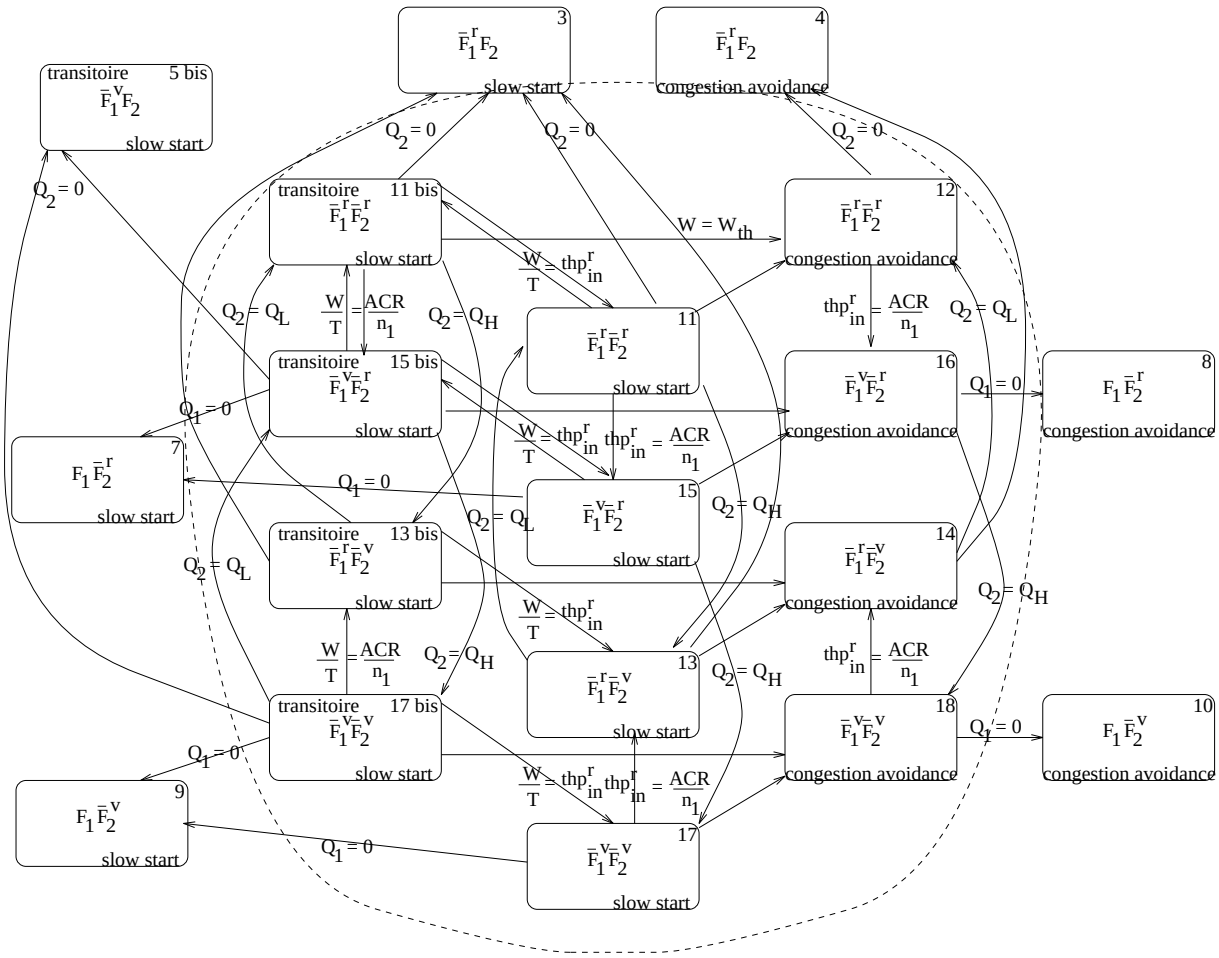


FIG. A.4 - *Diagramme d'états, lorsque F_1 et F_2 sont non vides*

Annexe B

Glossaire

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACR	Allowed Cell Rate
ADTF	ACR Decrease Time Factor
ATM	Asynchronous Transfer Mode
BP	Bande Passante
BRM	Backward RM cell
CAC	Connection Admission Control
CBR	Constant Bit Rate
CCR	Current Cell Rate
CDF	Cutoff Decrease Factor
CDV	Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CER	Cell Error Ratio
CI	Congestion Indication
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CMR	Cell Misinsertion Rate
CRM	Missing RM-cell count
CTD	Cell Transfer Delay
CV	Circuit Virtuel
DBR	Deterministic Bit Rate
DES	Destination End System
DGCRA	Dynamic Generic Cell Rate Algorithm
DIR	DIRection
ECN	Explicit Congestion Notification

EFCI	Explicit Forward Congestion Indication
EPD	Early Packet Discard
EPRCA	Enhanced Proportional Rate Control Algorithm
ER	Explicit Rate
ERICA	Explicit Rate Indication for Congestion Avoidance
FIFO	First In First Out
FRM	Forward RM cell
FRP-DT	Fast Reservation Protocol with Delayed Transmission
FRP-IT	Fast Reservation Protocol with Immediate Transmission
FRTT	Fixed Round Trip Time
GCRA	Generic Cell Rate Algorithm
GFR	Guaranted Frame Rate
ICR	Initial Cell Rate
IP	Internet Protocol
LAN	Local Area Network
LANE	LAN Emulation
MAN	Metropolitan Area Network
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MPOA	Multi-Protocol Over ATM
MPLS	Multi-Protocol Label Switching
MSS	Maximum Segment Size
MTU	Maximum Transfer Unit
NI	No Increase
NNI	Network to Node Interface
OAM	Operation And Maintenance
PACTA	Programme d'Analyse d'une Connexion TCP sur ABR
PBS	Partial Buffer Sharing
PCR	Peak Cell Rate
PO	Push Out
PPD	Partial Packet Discard
PRQCA	Proportional Rate and Queue Control Algorithm
QoS	Quality of Service
RDF	Rate Decrease Factor

RED	Random Early Detection
RIF	Rate Increase Factor
RM	Resource Management
RNIS-LB	Réseau Numérique à Intégration de Services-Large Bande
RTO	Retransmission Time Out
RTT	Round Trip Time
SBR	Statistical Bit Rate
SCR	Sustainable Cell Rate
SECBR	Severely Errored Cell Block Ratio
SES	Source End System
SSCS	Service Specific Convergence Sublayer
STCP	Simulated Transport Control Protocol
TBE	Transient Buffer Exposure
TCP	Transmission Control Protocol
TCR	Tagged Cell Rate
THP	Throughput
UBR	Unspecified Bit Rate
UIT	Union Internationale des Télécommunications
UNI	User Network Interface
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VC	Virtual Connection
VD	Virtual Destination
VP	Virtual Path
VS	Virtual Source
WAN	Wide Area Network
WFQ	Weighted Fair Queueing

Résumé

L'avènement du mode de transfert temporel asynchrone (ATM) dans les réseaux de télécommunications a nécessité la spécification de nouveaux concepts relatifs au contrôle de trafic. Toutefois l'apparition de nouveaux mécanismes se heurte à l'utilisation intensive des schémas de contrôle de flux classiques développés pour le transfert de données sur les réseaux de paquets, tels que ceux implémentés dans le protocole TCP. Afin de résoudre les problèmes liés aux interactions entre ces deux générations de mécanismes et de favoriser le déploiement d'un contrôle de trafic adapté aux réseaux numériques à intégration de services "large bande", de nouvelles recherches consacrées à l'étude des performances des applications de transfert de données sur ATM sont apparues nécessaires. Ces recherches font l'objet de cette thèse.

Nous décrivons dans un premier temps les mécanismes de contrôle de trafic proposés dans la littérature et déterminons les solutions qui s'adaptent le mieux au mode de transfert ATM. Dans le même temps, nous présentons d'un côté le protocole TCP et son schéma de contrôle de flux de bout en bout fondé sur une fenêtre, et de l'autre côté la capacité de transfert ABR qui utilise un mécanisme de contrôle de trafic par débit sur la partie ATM du réseau. Ce premier aperçu nous conduit à proposer une étude analytique des performances de la capacité de transfert ABR lors du transport de flux TCP/IP sur un réseau ATM. Nous modélisons le comportement d'une connexion TCP sur un circuit virtuel ATM-ABR en présence d'un trafic exogène incontrôlé. L'analyse fluide de ce système permet d'une part l'étude des interactions entre le protocole TCP et le service ABR, et d'autre part d'expliquer le comportement de TCP sur ABR. Les résultats obtenus témoignent de l'attrait de la capacité de transfert ABR pour le transport de données.

Nous poursuivons notre étude par l'analyse numérique des performances de TCP sur ABR. A partir du modèle précédent, nous mesurons la robustesse des paramètres ABR dans diverses configurations de réseaux, en tenant compte de l'influence du protocole TCP et des autres connexions ou services. Nous déterminons ainsi un paramétrage optimal de la capacité ABR. L'objectif final de cette thèse est de proposer un recueil de recommandations à l'attention de l'opérateur France Télécom en vue de l'optimisation des performances des applications de transfert de données sur ATM. Cet objectif est lié à la volonté de France Télécom d'offrir une plus grande variété de services à ses clients. Dans ce cadre, nous motivons le choix de la capacité de transfert ABR conjuguée à d'autres mécanismes de contrôle de trafic.

Mots clés : mode de transfert ATM, capacité de transfert ABR, protocole TCP, contrôle de trafic et de congestion, théorie des files d'attente, analyse de performances