
Un ordonnanceur assurant des services différenciés relatifs basés sur le délai

Geoffroy Jennes, Guy Leduc et Mudassir Tufail¹

Research Unit in Networking (RUN)

Université de Liège

Institut Montefiore - Bât 28

Grande Traverse, 10

B-4000 Liège

Belgique

Tél : +32 4 366 2691, Fax : +32 4 366 2989

jennes@run.montefiore.ulg.ac.be, leduc@montefiore.ulg.ac.be

RÉSUMÉ. Nous proposons l'ordonnanceur RD-VC garantissant des services différenciés (DIFF-SERV) relativement quantifiés basés sur le délai. Un tel ordonnanceur assure que différentes classes de service obtiennent des délais inversement proportionnels à des indices de qualité fixés, quelle que soit la charge du réseau, et sans signalisation, ni contrôle d'admission. De plus, bien que cet algorithme ne s'applique qu'aux classes de services, les délais relatifs sont également respectés pour chaque flux composant ces agrégats. Ceci permet d'assurer des qualités de service au niveau des (micro)flux sans que leur grand nombre ne conduise à un effondrement du système. L'algorithme offre de bonnes performances tout en étant très peu coûteux : le traitement par paquet est très léger et il n'est même pas nécessaire de disposer d'une file d'attente par classe de service.

ABSTRACT. We propose the RD-VC scheduler providing relative Differentiated Services (DIFF-SERV) based on delay. Such a scheduler ensures that delays in different service classes are inversely proportional to fixed quality indices, whatever the load of the network and without any signaling nor admission control. Moreover, although the algorithm is class based, relative delays are respected at the flow level. This allows us to ensure Quality of Services at (micro)flow level in a scalable way, without system degradation when the number of flows increases. The algorithm provides good performance while consuming little resource : the processing of a packet is very light and the approach does not even require to implement a queue per class.

MOTS-CLÉS : Services différenciés, services relativement quantifiés, ordonnanceur de réseau, délais relatifs, performance, simulation

KEYWORDS: Differentiated Services, DiffServ, Assured Forwarding, relatively quantified services, scheduling, relative delays, performance, simulation

1. Nouvelle adresse : Mudassir@calynet.com

1. Introduction

La croissance de l'Internet a conduit à un ensemble de réflexions sur la manière d'optimiser l'utilisation de ce réseau afin que les applications puissent bénéficier de services appropriés au lieu de subir une politique globale dite "BEST-EFFORT" fournie actuellement dans l'Internet. Les Qualités de Service (QoS) sont devenues incontournables [XIA 99] et nécessitent l'implantation de divers mécanismes dans les entités de réseau. L'intérêt énorme de la communauté scientifique pour ce problème s'est concrétisé par le développement rapide de deux grands standards.

Le premier, l'Intégration de Services ("INTSERV") [BRA 94], permet d'assurer des QoS absolues aux flux de données sur les chemins qu'ils empruntent dans le réseau. Les services intégrés nécessitent de réserver dynamiquement des ressources pour chaque flux et dans chaque nœud traversé. Le protocole "RSVP" [BRA 97, WRO 97] remplit cette tâche. Le principal inconvénient lié à cette approche est sa mauvaise adaptabilité à la croissance du nombre de flux. En effet, les ressources nécessaires dans chaque nœud sont proportionnelles au nombre de flux.

Le second standard, la Différenciation de Services ("DIFFSERV") [BOR 99, BLA 98], résiste mieux au facteur d'échelle en assurant les Qualités de Service à des classes ou agrégats plutôt qu'aux flux de données individuels. Ce faisant, le risque existe de perdre les QoS au niveau des flux, vu que ceux-ci seront traités sans discernement au sein des agrégats.

A cet égard, nous verrons que l'approche proposée dans cet article a le mérite d'assurer ces QoS aux flux individuels avec un ordonnanceur qui ne traite que des agrégats. Ceci est rendu possible par le choix de la métrique servant à exprimer les QoS. Nous verrons par exemple que le délai est parfaitement adapté.

Afin de simplifier l'architecture en évitant toute signalisation ou contrôle d'admission, nous proposons de fournir des services relatifs. Le principe est de définir plusieurs classes ordonnées par délais décroissants, en quantifiant les rapports entre les délais des classes par des indices de qualité fixés. L'ordonnanceur que nous proposons aura pour objectif d'assurer ces ratios quelle que soit la charge du réseau dans les différentes classes.

En combinant ces éléments, il est donc possible d'assurer aux classes (mais aussi indirectement à tous les flux de ces classes) des services différenciés en termes de délais relativement quantifiés.

Par souci de concision, nous parlerons simplement de *services relatifs* dans la suite de cet article pour désigner les QoS relativement quantifiées basées sur le délai.

2. Services différenciés

Dans ce type de services [BOR 99, BLA 98], la différenciation s'opère au niveau des agrégats plutôt qu'au niveau des flux eux-mêmes. Afin de proposer des Qualités

de Services différenciées de bout en bout dans un réseau, un certain nombre de mécanismes plus ou moins complexes doivent être mis en œuvre, tels que des mécanismes de contrôle d'admission qui acceptent ou non un nouveau flux suivant les ressources disponibles dans le réseau mais aussi d'autres mécanismes de contrôle de conformité sanctionnant les flux entrants s'ils ne respectent pas leur part du contrat établi. Ces mécanismes de contrôle sont implantés dans les nœuds périphériques du réseau.

Le nouveau flux entrant doit également être classé suivant ses QoS. Servir les agrégats ainsi formés pour qu'ils reçoivent les QoS demandées nécessite des méthodes spécifiques de gestion de files d'attente et d'ordonnancements. Ces mécanismes, regroupés au sein d'un nœud, constituent le concept de "PER HOP BEHAVIOUR" (PHB).

Le groupe de travail "DIFFSERV" de l'IETF (INTERNET ENGINEERING TASK FORCE) a défini trois grandes classes de services : "EXPEDITED FORWARDING" (EF) [JAC 99], "ASSURED FORWARDING" (AF) [HEI 99] et "BEST-EFFORT" (BE).

La classe EF est proposée pour garantir des QoS quantitatives en terme de bande passante, de délai, de perte et de jigue. La classe AF, quant à elle, reçoit une certaine quantité de ressources dans chaque nœud. Elle est divisée en quatre sous-classes appelées respectivement AF1, AF2, AF3 et AF4. Pour chacune de ces sous-classes, trois niveaux de précedence sont définis et correspondent à des probabilités croissantes de perte de paquet au sein de cette sous-classe. La combinaison de ces deux aspects donne alors douze niveaux de différenciation de services. Enfin, la classe BE ne garantit aucune QoS. Elle sert les flux en utilisant les ressources restantes.

La classe à laquelle nous allons nous intéresser dans ce document est la classe AF. Deux grandes approches liées à la classe AF ont émergé. La première propose de fournir des QoS différenciées qualitatives, c'est à dire que les agrégats recevront des services non-mutuellement dépendants. Par exemple, un agrégat lié à une sous-classe recevra des services garantissant de faibles pertes, tandis qu'une autre recevra des services garantissant de faibles délais, sans pour autant les quantifier. La deuxième approche propose des services différenciés relativement quantifiés, c'est à dire que les QoS des quatre sous-classes AF seront liées entre-elles par des facteurs donnés en se basant sur une même métrique. Par exemple, une classe "A" recevra un service N fois meilleur qu'une classe "B". La valeur N étant déterminée par le rapport des indices de qualité de ces deux classes.

3. Choix d'une métrique

Dans les réseaux informatiques, trois grandes métriques sont utilisées pour définir des QoS : *la bande passante, le délai et le taux de pertes*. Nous allons étudier chacune d'elles afin de déterminer celle qui sera la plus adaptée aux deux exigences suivantes.

Nous voulons d'une part que les services offerts aux agrégats se reflètent au niveau des flux individuels, et d'autre part que la différenciation s'intègre bien dans le contexte de DiffServ.

3.1. *La bande passante*

Supposons qu'une classe de service composée de multiples flux se voie attribuer une bande passante B . Que peut-on dire des bandes passantes disponibles pour les flux composant cet agrégat ? A priori rien si on ne connaît aucune information sur celui-ci. En supposant même que les flux disposent d'une part équitable de la bande passante B , la bande passante d'un flux dépendrait toujours du nombre de flux présents dans la classe.

Il en est de même pour les services relativement quantifiés. Si deux classes se voient attribuer des bandes passantes B et $2B$, chaque flux de la deuxième n'aura pas en général deux fois plus de bande passante que chaque flux de la première. A nouveau, cela dépend du nombre de flux dans chaque classe. Pour assurer des QoS au niveau des flux, l'ordonnanceur devrait au minimum connaître ces nombres en permanence et s'adapter en conséquence.

Nous considérons que la bande passante n'est donc pas une métrique adéquate, car DiffServ considère que les nœuds intermédiaires doivent tout ignorer du contenu des classes.

3.2. *Le taux de pertes*

Lorsqu'un agrégat subit un taux de pertes p , ce taux se répercute approximativement sur chaque flux de cet agrégat. Cette approximation peut être très bonne si les pertes sont dues à des mécanismes de décongestion tels que RED "RANDOM EARLY DETECTION" [FLO 93] plutôt qu'à des pertes par débordement de mémoire. En effet, par la destruction aléatoire de paquets avant que la congestion ne soit critique, RED assure une bonne équité entre les flux d'un même agrégat en détruisant un nombre de paquets de chaque flux proportionnellement à leur charge, donc en assurant le même taux de perte p à chaque flux.

Les taux de pertes assurés au niveau des agrégats se répercutent donc naturellement sur tous les flux qui les composent. Contrairement à la bande passante, cette métrique satisfait ainsi le premier critère.

Cependant, comme nous l'avons décrit à la section 2, en plus de définir quatre sous-classes AF, le standard définit la notion de précedence. La précedence d'un paquet au sein même d'une sous-classe détermine son éligibilité à subir une perte en cas de congestion. Les paquets possédant le niveau de précedence le plus élevé seront sanctionnés les premiers. Et ainsi de suite vers les niveaux de précedence plus faible, si la congestion persiste. Le niveau de précedence d'un paquet peut être déterminé par l'application qui le génère ou par un routeur périphérique.

Utiliser une métrique comme le taux de pertes pour garantir des services relatifs par agrégats, tout en gardant ce type de différenciation au sein des agrégats par des niveaux de précedence ne nous semble pas opportun. Cela conduirait à 12 taux de

pertes, partiellement ordonnés, qui seraient difficiles à utiliser, et à une complexité accrue des mécanismes de destruction de paquet dans les nœuds.

3.3. *Le délai*

Prendre en compte le délai comme métrique de différenciation de services s'avère être la solution la plus adaptée. En effet, assurer un délai d au niveau d'un agrégat se répercute automatiquement sur chacun de ses flux qui héritent tous du même délai d . De même, si deux classes assurent des délais relatifs d et $2d$, chaque flux de la seconde aura un délai deux fois plus grands que celui de chaque flux de la première, quelle que soient les charges respectives des classes.

Le délai est également une métrique très prisée dans les réseaux, que ce soit pour les applications interactives exigeant de faibles délais, ou pour des mécanismes de contrôle de flux en *boucle fermée* tels que TCP, qui ont de meilleures performances lorsque le délai aller-retour est petit. Le débit de TCP par exemple est, selon [MAT 97], approximativement inversement proportionnel à ce délai.

Dès lors, nous baserons notre différenciation de service sur cette métrique qui répond à nos deux exigences.

Nous étudierons d'abord quelques ordonnanceurs en mesure d'assurer de tels délais relatifs, quelles que soient les charges des classes et sans faire intervenir de contrôle d'admission.

4. État de l'art sur les ordonnanceurs basés sur les délais

Quelques algorithmes d'ordonnement assurant des délais relatifs ont déjà été proposés.

Nous allons les classer en deux grandes catégories. La **catégorie A** reprend les propositions imposant l'implantation de files d'attente séparées pour chacune des classes de trafic, tandis que la **catégorie B** reprend les propositions se contentant d'une seule file d'attente pour toutes les classes de trafic.

Deux algorithmes peuvent être classés dans la catégorie A : "BACKLOG PROPORTIONAL RATE" (BPR) [DOV 99] et "WAITING TIME PRIORITY" (WTP) [KLE 67, DOV 99].

Dans la catégorie B, nous présenterons l'algorithme "EXTENDED VIRTUAL-CLOCK" (Ex-VC) [TUF 99a, TUF 99c].

A noter qu'il existe d'autres types d'algorithmes également basés sur les délais, (par exemple : "EARLIEST DEADLINE FIRST" (EDF) [DER 74, GUE 98]) mais celui-ci n'a pas été développé dans l'optique de fournir des services relatifs. C'est pourquoi nous ne l'avons pas étudié dans ce document.

4.1. Catégorie A : ordonnanceurs requérant plusieurs files d'attente

Cette section présente les deux ordonnanceurs BPR et WTP nécessitant une file d'attente par classe.

4.1.1. "BACKLOG PROPORTIONAL RATE" (BPR)

L'idée initiale de cet ordonnanceur est de se baser sur le modèle de distribution de bande passante GPS [PAR 93] mais en implantant la spécificité suivante : réajuster dynamiquement la bande passante fournie aux différentes classes en se basant sur sa charge et sur l'indice de qualité lui étant attribué. L'indice de qualité est une information propre à la différenciation de service relative qui détermine le ratio entre les délais des différentes classes.

Cet ordonnanceur BPR est discrétisé au niveau des paquets et se base sur une fonction de service définie pour chaque file d'attente, donc pour chaque classe. Cette fonction tente d'approximer le service que le paquet en tête de la file devrait recevoir s'il était servi de manière fluide à l'aide de l'ordonnanceur GPS.

Pour cet ordonnanceur BPR, il est nécessaire de re-calculer la fonction de service à chaque fois qu'un paquet est sélectionné pour traitement.

4.1.2. "WAITING TIME PRIORITY" (WTP)

L'ordonnanceur WTP, initialement développé par Kleinrock [KLE 67], est un ordonnanceur se basant sur la priorité associée à un paquet. Avec cette approche, la priorité du paquet augmente en fonction de son temps d'attente dans le système [DOV 99]. Pour cela, l'algorithme se base sur le temps d'arrivée du paquet. Il est donc nécessaire de les estampiller dès leur arrivée.

A chaque fois que l'algorithme WTP doit sélectionner un paquet, chaque file d'attente active est remise à jour en calculant les priorités des paquets en tête de chaque file, ce qui augmente la complexité de cette approche.

Une fois les priorités calculées, l'ordonnanceur choisit de servir la file contenant le paquet de plus haute priorité.

A noter que cet ordonnanceur ne tient pas compte de la taille du paquet pour calculer les priorités.

4.1.3. BPR versus WTP

Une étude comparative des deux algorithmes montre que les deux approches sont à même de fournir des services relatifs basés sur les délais [DOV 99, CNO 99]. Il en ressort toutefois que l'algorithme WTP, tout en étant plus simple, donne de bien meilleurs résultats. Dès lors, les prochaines comparaisons se baseront sur WTP.

4.2. Catégorie B : ordonnanceur ne requérant qu'une seule file d'attente

Cette catégorie reprend un algorithme, appelé "EXTENDED VIRTUAL-CLOCK" (Ex-VC) [TUF 99a, TUF 99c].

L'idée initiale sur laquelle se base Ex-VC est de réajuster dynamiquement la bande passante fournie aux différentes classes en se basant sur la charge et sur l'indice de qualité de la classe. Cependant, cette approche ne tente pas d'approximer le modèle de service GPS [PAR 93], mais se base sur l'ordonnanceur "VIRTUAL-CLOCK" [ZHA 91].

Comme indiqué précédemment, le réajustement dynamique de la bande passante pour une classe de trafic se base sur sa charge propre, celle-ci étant mesurée à partir de la taille de sa file d'attente et sur celles des autres classes. De plus, ce réajustement dynamique est pondéré en fonction de l'indice de qualité spécifié.

Une fois estampillés, les paquets sont servis dans l'ordre croissant de leur estampille.

Ex-VC ne souffre pas des inconvénients des algorithmes de la catégorie A et le coût par paquet reste constant. Cependant, dans le document [TUF 99b], il est montré que cet ordonnanceur ne réagit pas parfaitement en fonction de la charge de chacune des classes. Ce manque de performance est lié au fait que l'estimation de la charge d'une classe, et donc l'estampille, se base sur la taille de sa file d'attente, au moment où le paquet arrive, alors que l'état du système a changé au moment où le paquet est servi.

Dans la suite, nous allons développer un nouvel ordonnanceur utilisant une seule file d'attente comme Ex-VC, mais offrant de meilleures performances que celui-ci.

5. Proposition d'un nouvel ordonnanceur

Par souci de simplicité, nous recherchons un algorithme d'ordonnancement peu coûteux en traitement par paquet, et qui puisse n'utiliser qu'une file d'attente commune aux différentes classes.

Le principe est d'estampiller une seule fois chaque paquet à son arrivée. Les paquets seront alors servis par ordre d'estampilles croissantes. Toute la difficulté est d'allouer correctement celles-ci afin de respecter quantitativement les délais relatifs entre classes.

Cet ordonnanceur sera appelé RD-VC (Relative Delay Virtual Clock).

5.1. Description formelle de RD-VC

Considérons le cas général de N classes actives. Définissons q_i comme étant l'indice de qualité de la classe i et imposons un ordre croissant des indices de qualité.

$$q_1 \leq q_2 \leq q_3 \dots \leq q_N \quad [1]$$

Cette relation 1 définit que la classe N a l'indice de qualité le plus élevé ; ce qui signifie que cette classe N devra recevoir la meilleure Qualité de Service.

En considérant que l'indice de qualité q_i représente le service relatif d'une classe i , et que d_i représente le délai par paquet de cette même classe i , nous pouvons dire que notre objectif est d'assurer la contrainte suivante :

$$\frac{q_i}{q_j} = \frac{d_j}{d_i} \quad [2]$$

Pour simplifier l'exposé, nous supposerons que chaque classe a sa propre file d'attente, même si une telle implantation n'est pas nécessaire comme nous le verrons. Considérons l'état du système au temps t : soit b_i la taille de la file d'attente de la classe i (ou simplement de la file i), et soit

$$B = \sum_i^N b_i \quad [3]$$

Le délai d_i associé au dernier paquet de la file i vaudra en fait

$$d_i = \frac{b_i}{r_i} \quad [4]$$

où r_i est le taux de service moyen (futur) de la file i pendant le temps d_i qui est nécessaire pour vider la file i des paquets présents au temps t .

A partir des équations 3, 4 et 2, nous pouvons déduire que

$$B = \sum_i^N b_i = \sum_i^N r_i d_i = \sum_i^N r_i d_1 \frac{q_1}{q_i} \quad [5]$$

Ceci permet d'extraire le délai associé au dernier paquet de la file 1, et donc aussi la valeur de l'estampille à appliquer à un paquet de cette classe : $d_1 = \frac{B}{q_1 \sum_i^N \frac{r_i}{q_i}}$.

Plus généralement, nous avons

$$d_i = \frac{B}{q_i \sum_j^N \frac{r_j}{q_j}} \quad [6]$$

Ces équations ne sont pas encore utilisables, car les r_i sont inconnus. Heureusement, il est possible de bien les approximer.

Si aucun nouveau paquet n'arrivait dans le système et si les estampilles des derniers paquets de chaque file étaient identiques, on aurait immédiatement $\frac{r_i}{C} = \frac{b_i}{B}$ où C est la capacité de la ligne de sortie. En effet, toutes les files seraient servies dans le même temps et le taux moyen de service de chaque file serait proportionnel au contenu de chacune d'elles.

Supposons maintenant qu'un paquet arrive dans chaque classe à peu près au même instant, l'estampille du dernier paquet de la classe 4 aura forcément une valeur inférieure à celles des derniers paquets des autres classes, étant donné que l'ordonnanceur favorise la classe 4. De plus, certains paquets futurs de la classe 4 seront même servis avant certains paquets déjà présents dans les autres files, pour la même raison.

Soit b_i le nombre de paquets présents dans la file i au temps t , considérons alors la classe 4. Pendant le temps nécessaire pour qu'elle se vide de tous les paquets présents au temps t (b_4 paquets), la classe 1 ne se videra approximativement que de $b_1 \frac{q_1}{q_4}$ si l'ordonnanceur sert bien les classes 1 et 4 selon les indices de qualité souhaités. Plus généralement, pendant ce temps, la classe i se videra de $b_i \frac{q_i}{q_4}$. Dès lors, $\frac{r_4}{C} = \frac{b_4}{\sum_i b_i \frac{q_i}{q_4}}$ plutôt que $\frac{r_4}{C} = \frac{b_4}{B}$.

De façon similaire, considérons la classe 3. Le raisonnement ci-dessus est le même vis-à-vis des classes 1 et 2. Cependant, pendant le temps nécessaire pour que la classe 3 se vide de tous les paquets présents au temps t (b_3 paquets), la classe 4, non seulement se videra entièrement mais, aura en plus le temps d'écouler des paquets non encore arrivés à l'instant t . Si nous supposons, par extrapolation, que les taux d'arrivée des paquets dans un futur proche seront approximativement les mêmes que les taux récents, nous obtenons que la classe 4 aura le temps d'écouler $b_4 \frac{q_4}{q_3}$, c'est-à-dire plus que son contenu au temps t . Ce raisonnement peut être répété pour les classes 1 et 2, et nous conduit à la formule suivante : $\frac{r_i}{C} = \frac{b_i}{\sum_j b_j \frac{q_j}{q_i}}$

Ce qui peut encore s'écrire :

$$r_i = \frac{b_i q_i C}{\sum_j^N b_j q_j} \quad [7]$$

De 6 et 7, nous obtenons

$$d_i = \frac{B}{q_i \sum_k^N \frac{b_k C}{\sum_j^N b_j q_j}} = \frac{\sum_j^N b_j q_j}{q_i C} \quad [8]$$

Tout nouveau paquet arrivant dans la classe i sera donc estampillé comme suit : $stamp_i = a_i + d_i$, où a_i est le temps d'arrivée de ce nouveau paquet. Vu la dynamique du système et afin d'éviter que deux paquets successifs d'une même file ne se voient

attribuer des estampilles décroissantes, on ajoute la contrainte que l'estampille du dernier paquet de chaque file doit au moins être égale à celle du paquet qui le précède. Les paquets seront alors servis dans l'ordre croissant des estampilles.

Nous avons effectué une série de simulations afin de valider notre approche. Nous avons considéré les q_i suivants : $q_1 = 1$, $q_2 = 2$, $q_3 = 3$ et $q_4 = 4$.

Le graphique gauche de la figure 1 montre les taux d'arrivée utilisés pour chacune des quatre classes de trafic. Les caractéristiques du trafic d'arrivée changent toutes les 500 unités de temps¹ afin de compliquer le scénario de simulation.

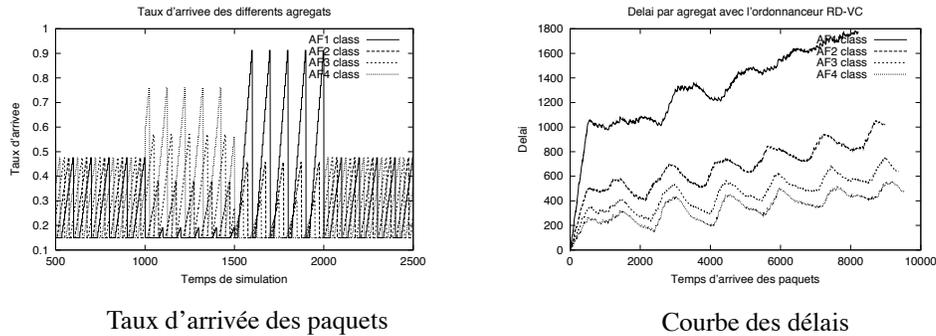


FIG. 1. Courbe des délais par classes pour le modèle basé sur l'équation 8.

La partie droite de la figure 1 montre le délai par classes obtenu en utilisant l'équation 8. Nous remarquons que les délais relatifs sont assurés et que les courbes de délai sont assez lisses² ; ce qui signifie que la jigue est relativement faible pour chacune des classes, malgré les fluctuations importantes des débits d'arrivée.

Dans la section suivante, nous allons montrer par d'autres simulations que cet ordonnanceur a un excellent comportement, compte tenu du fait que les estampilles sont fixées une fois pour toutes lors de l'arrivée des paquets. De plus, cet algorithme ne nécessite pas une file par classe. Lorsqu'une seule file est utilisée, il suffit qu'elle soit triée par estampilles croissantes et que l'ordonnanceur conserve dans quelques registres les nombres de paquets de chaque classe dans cette file unique, ainsi que les valeurs des dernières estampilles attribuées à chaque classe.

6. Résultats des simulations

Les simulations présentées dans cette section sont des simulations réalisées en injectant des flux de trafic "fluides", c'est-à-dire avec un temps inter-arrivée entre pa-

1. Une unité de temps représente le temps qu'il faut pour transmettre complètement un paquet. De plus les paquets utilisés pour cette première simulation sont de taille fixe.
2. Les fluctuations des délais sont en fait inévitables vu les conditions de charge du système, celles-ci changeant toutes les 500 unités de temps.

quets constant. Cependant, différentes matrices de trafics seront utilisées, modifiant la charge d'entrée pour chacune des quatre classes.

Le but de ces simulations est de vérifier et comparer les performances des différents ordonnanceurs offrant des services relatifs et cela suivant différentes configurations de charge du système. Pratiquement nous allons simuler quatre classes AF : AF1, AF2, AF3, et AF4. Ces classes ont respectivement les indices de qualité suivant : $q_1 = 1$, $q_2 = 2$, $q_3 = 3$ et $q_4 = 4$. Cela signifie que la classe AF4 recevra une qualité de service 4 fois meilleure que la classe AF1, se concrétisant par un délai par paquet 4 fois moindre.

Une période de chargement du système est configurée afin d'atteindre un état de charge globale raisonnable. Durant cette période, le taux d'arrivée des paquets sera deux fois plus élevé que le taux de service de l'ordonnanceur. Après quoi, nous imposerons un taux d'arrivée égal au taux de service, afin de stabiliser le système. De ce fait nous serons capables de voir le temps mis par l'ordonnanceur pour atteindre l'état stable souhaité.

Pour ce qui est du taux d'arrivée des paquets, trois scénarios sont définis : un scénario dit *symétrique*, un deuxième dit *uniforme* et un troisième dit *asymétrique*. Ces trois scénarios sont représentés à la figure 2.

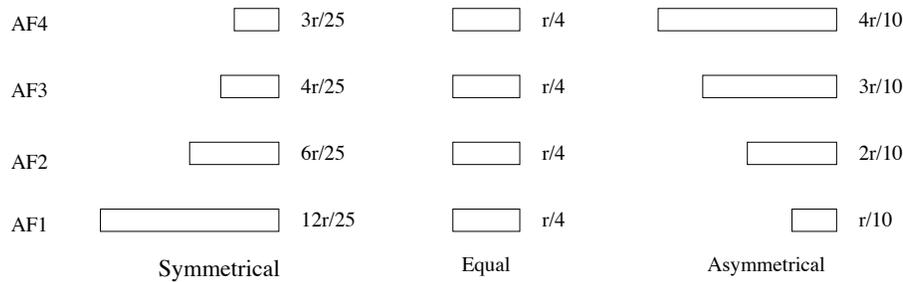


FIG. 2. Présentations des trois scénarios décrivant le taux d'arrivée pour chacune des 4 classes

En voici une description plus détaillée :

- **Le scénario *symétrique*** : les différentes classes sont chargées proportionnellement à leur garantie relative de délai. Par exemple, la classe AF4 aura un taux d'arrivée quatre fois plus faible que la classe AF1. Cette situation est une situation a priori favorable pour l'ordonnanceur.
- **Le scénario *uniforme*** : les taux d'arrivée sont identiques pour chacune des quatre classes indépendamment de leur indice de qualité, et donc de leur garantie en terme de délai relatif.
- **Le scénario *asymétrique*** : c'est a priori le cas le plus défavorable pour l'ordonnanceur. Le taux d'arrivée sera inversement proportionnel à l'indice de qualité.

Ainsi, la classe AF4 aura un taux d'arrivée quatre fois plus élevé que la classe AF1.

Les périodes de charge et de stabilité feront en général l'objet de scénarios différents afin de simuler un ensemble représentatif de cas. Lors du passage entre les 2 périodes, l'ordonnanceur devra alors s'adapter de manière plus ou moins importante en fonction de la situation à laquelle il est confronté. Nous ne présenterons toutefois que les situations les plus intéressantes : principalement les simulations réalisées avec une période de charge *uniforme*. Ce choix se justifie par le fait qu'il permet de placer le système dans une situation intermédiaire à partir de laquelle on peut étudier tous les scénarios possibles pour la période de stabilité. Les paramètres de la simulation sont : une période de charge de 500 unités de temps et une période de stabilité de 10,000 unités de temps.

6.1. Étude comparative du délai par paquet pour les trois ordonnanceurs

La chose la plus importante à remarquer à la figure 3 est que chacun des trois ordonnanceurs offre une différenciation de service relative basée sur le délai ; cette différenciation étant définie par les indices de qualité associés aux différentes classes AF.

Toutefois, il faut noter que les trois ordonnanceurs s'adaptent différemment pour atteindre un état stable.

Les résultats obtenus peuvent se résumer comme suit :

- **L'ordonnanceur *Ex-VC*** : Les performances de cet ordonnanceur sont les moins bonnes. L'algorithme fluctue assez bien avant d'atteindre un état stable suivant le scénario de charge qu'on lui applique [TUF 99b]. Ce manque de performance est principalement dû à son mode d'adaptation aux fluctuations de charge (section 4.2). En effet, il se base sur la taille moyenne des files d'attente des différentes classes pour s'auto-réguler plutôt que de se baser directement sur une notion de délai. De plus, les paquets étant estampillés dès leur arrivée, l'état du système, en terme de taille de file d'attente, peut donc changer durant le transit du paquet jusqu'en tête de file. Cependant l'algorithme est très simple puisqu'il n'a besoin que d'une seule file d'attente pour fonctionner. Il pourra mémoriser les variables sur lesquelles il se base (typiquement les tailles des files d'attente des classes) dans de simples registres. La simplicité est un atout non négligeable pour de tels algorithmes destinés à être implantés dans des routeurs à haut débit.
- **L'ordonnanceur *WTP*** : Cette approche apparaît directement comme étant la plus performante quant à sa rapidité de stabilisation. Par contre, une chose importante à noter est qu'elle ne respecte pas très bien les indices de qualité durant la période de charge. Ce phénomène se remarque sur la partie b de la figure 3, durant les 500 premières unités de temps de la simulation (période de charge). Typiquement, lors d'une forte charge du système, WTP favorise beaucoup trop

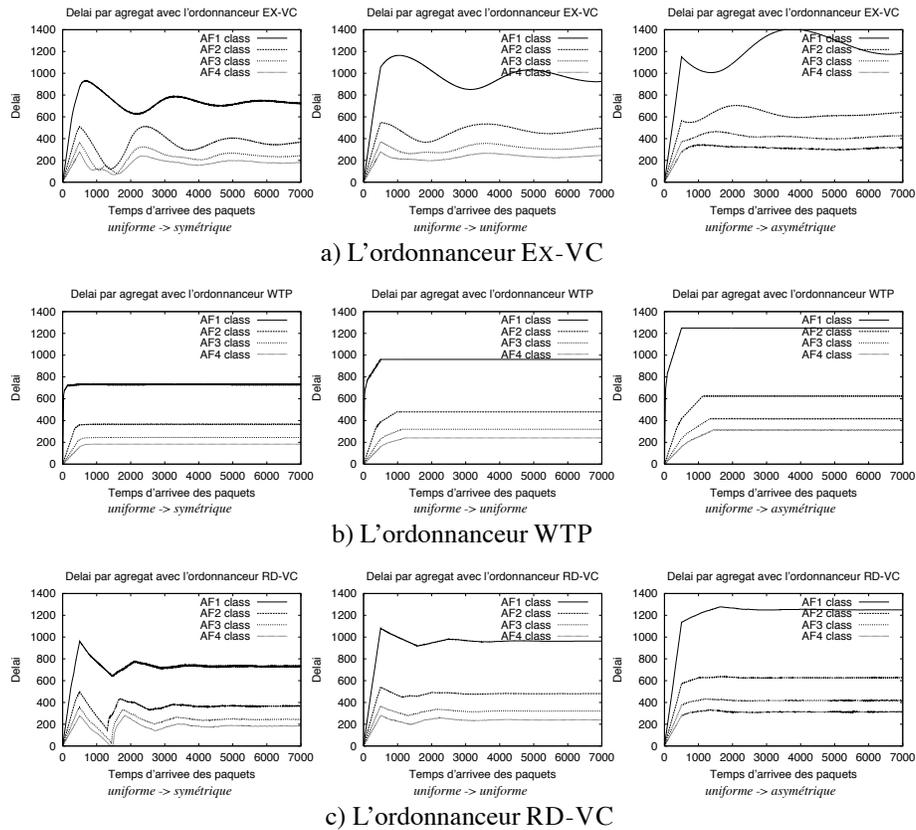


FIG. 3. Résultats des simulations : présentation du délai par paquet pour les trois ordonnanceurs

les classes requérant les meilleures qualités, laissant ainsi de côté les classes les moins exigeantes. De plus, cette approche souffre de deux autres inconvénients. Le premier est sa complexité. En effet cet algorithme requiert une file d'attente séparée pour chacune des classes de trafic. Le deuxième inconvénient est son coût élevé de traitement par paquet. En effet, un paquet est traité au minimum deux fois. Une fois à son arrivée quand on l'estampille avec son temps d'arrivée, et une seconde fois quand il atteint la tête de sa file d'attente et que l'on calcule sa priorité afin qu'il soit servi selon les indices de qualité fixés.

- **L'ordonnanceur RD-VC** : Notre algorithme se distingue par sa simplicité et par ses performances. En effet, contrairement à WTP, durant la période de charge, les ratios entre les délais des différentes classes respectent parfaitement les indices de qualité fixés. Lors du changement de période, après un délai d'adaptation, l'algorithme se stabilise rapidement et garantit à nouveau les qualités requises. A cet égard, il est moins bon que WTP, mais ce résultat n'est pas sur-

prenant. En effet, RD-VC, comme Ex-VC, estampille les paquets à leur arrivée en fonction de l'état du système à cet instant. Au moment où les paquets sont servis, les conditions ont changé et, contrairement à WTP, on s'interdit pour des raisons d'efficacité de recalculer les estampilles. Compte tenu de cette différence importante, il est remarquable que RD-VC s'en sorte si bien, sachant qu'il fait en quelque sorte l'hypothèse que le futur proche sera semblable au passé proche. Il existe cependant des cas critiques où cette hypothèse est assez loin d'être vérifiée, ce qui explique les fluctuations de l'algorithme. Il faut toutefois remarquer que nous avons soumis les algorithmes à des conditions de fluctuations extrêmes des charges, qui sont certainement bien supérieures à la réalité. En effet, les charges des classes fluctuent moins que les charges individuelles des flux, et sont donc plus lisses en pratique.

7. Conclusion

Dans ce document, nous avons tout d'abord présenté ce qu'étaient des Qualités de Service relatives. Après avoir proposé trois métriques sur lesquelles se baser, celle du délai ressortait comme étant la plus adaptée à de tels services. Trois ordonnanceurs existants ont alors été introduits et comparés. Puis, nous avons développé de manière détaillée notre approche appelée RD-VC. Une série de simulations ont été réalisées et présentées graphiquement.

Parmi les trois ordonnanceurs, il ressort que l'approche WTP s'adapte plus rapidement que les autres. Toutefois, cette dernière ne respecte pas bien les indices de qualité pendant les périodes de charge importante et est nettement plus complexe à implanter dans des routeurs à haut débit.

RD-VC semble donc apporter le meilleur compromis entre simplicité et performance. Il n'a besoin que d'une seule file d'attente et limite fortement le coût de traitement par paquet en calculant directement la valeur d'estampille du paquet dès son arrivée dans le système.

Des études complémentaires sont en cours afin d'affiner les performances de RD-VC, notamment au niveau de la variation des délais. De plus, des simulations avec des sources telles que TCP, générant typiquement du trafic en rafale (trafic de type VBR), seront menées afin de vérifier ces résultats encourageants dans des cas plus représentatifs de la réalité et mettant en œuvre des topologies de réseau plus complexes. Enfin, certains mécanismes de décongestion tels que RED "RANDOM EARLY DETECTION" [FLO 93] pourront être implantés sur la file unique au sein du nœud afin de contrôler la charge de celui-ci tout en gardant un taux de perte équitable entre les classes. Le contrôle la charge des classes peut être également réalisé en mettant en œuvre un mécanisme de contrôle d'admission. Il sera alors possible de borner les délais subis dans un nœud implantant RD-VC tout en respectant les Qualités de Service relatives.

8. Bibliographie

- [BLA 98] BLAKE S., BLACK D., CARLSON M., DAVIS E., WANG Z., WEISS W., An Architecture for Differentiated Services , *Internet RFC 2475*, Dec. 1998.
- [BOR 99] BORAM Y., BINDER J., BLAKE S., CARLSON M., CARPENTER B. E., KESHAVE S., DAVIES E., OHLMAN B., VERMA D., WANG Z., WEISS W., A Framework for Differentiated Services , *Internet draft : draft-ietf-diffserv-framework-02.txt*, Feb. 1999.
- [BRA 94] BRADEN R., CLARK D., SHENKER S., Integrated Services in the Internet Architecture : an overview , *Internet RFC 1633*, 1994.
- [BRA 97] BRADEN R., ZHANG L., BERSON S., HERZOG S., JAMIN S., Resource ReSerVation Protocol(RSVP) – Version 1 Functional Specification , *Internet RFC 2205*, Sept. 1997.
- [CNO 99] CNODDER S. D., PAUWELS K., Relative delay priorities in a differentiated services network architecture , *Alcatel Alsthom CRC (Antwerp, Belgium) deliverable*, 1999.
- [DER 74] DERTOUZOS M. L., Control robotics : the procedural control of physical processes , *Proceedings IFIP Congress*, 1974, p. pp 807-813.
- [DOV 99] DOVROLIS C., STILIADIS D., Proportional Differentiated Services : Delay Differentiation and Packet Scheduling , *Proceedings of ACM SIGCOMM-99*, (<http://www.cae.wisc.edu/dovrolis/>), 1999.
- [FLO 93] FLOYD S., JACOBSON V., Random Early Detection Gateways for Congestion Avoidance , *IEEE/ACM Transactions on Networking*, 1993.
- [GUE 98] GUERIN R., PERIS V., Quality of Service in Packet Networks - Basic Mechanims and Directions , *Computer Networks and ISDN Systems, special issue on multimedia communications over packet based networks*, 1998.
- [HEI 99] HEINANEN J., FINLAND T., BAKER F., WEISS W., WROCLAWSKI J., Assured Forwarding PHB Group , *Internet RFC 2597*, June 1999.
- [JAC 99] JACOBSON V., NICHOLS K., PODURI K., An Expedited Forwarding PHB , *Internet RFC 2598*, June 1999.
- [KLE 67] KLEINROCK L., A Delay Dependent Queue Discipline , *Journal of the ACM*, Vol. 14, No. 2, 1967, p. 242-261.
- [MAT 97] MATHIS M., SMEKE J., MAHDAVI J., OTT T., The macroscopic behaviour of the TCP congestion avoidance algorithm , *Computer Communication Review*, July 1997.
- [PAR 93] PAREKH A. K., GALLAGER R. G., A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks : The Single-Node Case , *IEEE/ACM Transactions on Networking*, vol. 1, June 1993, p. 344-357.
- [TUF 99a] TUFAIL M., JENNES G., LEDUC G., A scheduler for delay-based service differentiation among AF classes , *IFIP Broadband Communications'99, Convergence of Network Technologies*, Nov. 1999, p. 93-102.
- [TUF 99b] TUFAIL M., JENNES G., LEDUC G., Providing a DiffServ like service in ATM networks , *Alcatel Alsthom CRC (Antwerp, Belgium) IWT Deliverable*, Oct. 1999.
- [TUF 99c] TUFAIL M., JENNES G., LEDUC G., Attaining per flow QoS with class-based differentiated services , *SPIE Symposium on Voice, Video and Data Communications, Conf : Broadband Network, Sept 1999.*, vol. 3842, Sept. 1999, p. 101-112.
- [WRO 97] WROCLAWSKI J., The Use of RSVP with IETF Integrated Services , *Internet RFC 2210*, Sept. 1997.

- [XIA 99] XIAO X., NI L. M., Internet QoS : the Big Picture , *IEEE Network Magazine*, March 1999.
- [ZHA 91] ZHANG L., VirtualClock : A new traffic control algorithm for packet switching , *ACM Transactions on Computer Systems*, vol. 9(2), May 1991, p. 101-124.