

# Intégration des garanties temporelles (m,k)-firm dans les ordonnanceurs WFQ pour les réseaux temps réel

Anis Koubaa, Ye-Qiong Song

► **To cite this version:**

Anis Koubaa, Ye-Qiong Song. Intégration des garanties temporelles (m,k)-firm dans les ordonnanceurs WFQ pour les réseaux temps réel. 3ème Journées Scientifiques Des Jeunes Chercheurs En Génie Electrique et Informatique 2003 - GEI'2003, 2003, Mahdia, Tunisie, 6 p, 2003. <inria-00107713>

**HAL Id: inria-00107713**

**<https://hal.inria.fr/inria-00107713>**

Submitted on 19 Oct 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Intégration des Garanties Temporelles $(m,k)$ -Firm dans les Ordonnanceurs WFQ pour les Réseaux Temps-Réel

Anis KOUBAA  
LORIA-TRIO  
2, Avenue de la Forêt de Haye  
54516 Vandoeuvre Lès Nancy, FRANCE  
[akoubaa@loria.fr](mailto:akoubaa@loria.fr)

Ye-Qiong SONG  
LORIA-TRIO  
2, Avenue de la Forêt de Haye  
54516 Vandoeuvre Lès Nancy, FRANCE  
[song@loria.fr](mailto:song@loria.fr)

## Résumé

*Pour satisfaire les contraintes  $(m,k)$ -firm, des algorithmes d'ordonnancement tel que Distance Based Priority (DBP) et Matrix-DBP ont été développés. Ces algorithmes sont de type Best-Effort et sont basés sur la priorité dynamique relative à la distance d'un flux et ne fournissent pas des garanties temporelles bornées. L'idée développée dans ce papier consiste à intégrer les contraintes  $(m,k)$ -firm dans les ordonnanceurs à débits garantis afin d'améliorer les délais de bout en bout tout en garantissant une borne sur le temps de réponse des flux temps réels. Une analyse par simulation montre l'avantage de cette technique par rapport à l'algorithme de Weighted Fair Queueing (WFQ) standard pour garantir des délais plus étroit en tenant compte des contraintes  $(m,k)$ -firm. L'étude montre que cette technique permet de découpler les délais du taux du coefficient de partage de bande passante.*

**Mots Clés :** WFQ,  $(m,k)$ -firm, qualité de service, simulation, équité

## 1. Introduction

Le nombre des ordinateurs et des dispositifs connectés en réseaux pour bénéficier de services en ligne ne cessent de s'accroître avec le temps. Pour cette raison, les futurs réseaux ont besoin de supporter une variété de services que ce soit en mode best-effort ou en mode réservé. Pour fournir des services garantis, l'algorithme d'ordonnancement WFQ [1] a été proposé et utilisé par les protocoles qui assurent la gestion de la qualité de service tel que *IntServ* dans la communauté de l'Internet. WFQ est une approximation du modèle fluide *Generalized Processor Sharing* (GPS) [1]. Avec GPS, chaque flux servi bénéficie d'un *coefficient de partage* qui fixe le taux de la bande passante pré-allouée. Ce modèle fluide garantit un partage équitable de la bande passante sur tout intervalle de temps en

assurant le service de chaque flux actif (présent dans sa file d'attente) en proportion avec son coefficient de partage et envoyant des quantités infinitésimales de données à partir de chaque file. GPS n'étant pas implémentable en pratique, des versions en paquet qui émulent le modèle fluide ont été développées. WFQ est l'une des versions principales et transmet les paquets dans l'ordre croissant de leur temps de départ dans GPS appelé *le temps virtuel de départ*. Ce temps virtuel de départ dépend du coefficient de partage de chaque flux. C'est l'unique critère de décision dans l'ordonnancement des paquets par WFQ et ses variantes. Des études précédentes ont analysé le délai de bout en bout assuré par un serveur à débit garanti tel que WFQ. Cependant, le problème majeur est que le délai est étroitement lié à la bande passante allouée à ce flux. Si un flux veut avoir un temps de réponse plus court, il doit réserver plus de bande passante. Nous avons traité une étude de cas à ce propos dans [7] où nous avons montré par simulation les avantages et les limites de WFQ vis-à-vis de l'ordonnancement à priorité stricte.

Cependant, dans quelques applications, des flux peuvent avoir besoin des garanties de délai plus strictes et ne réservant qu'un faible taux de bande passante. Par exemple, une communication téléphonique via Internet utilisant le protocole *Voix sur IP* est caractérisée par un trafic en rafale ayant un débit moyen assez réduit ( $\approx 64$  Kbit/s) et pourtant nécessite un délai très restreint pour assurer la bonne interactivité de cette application. De plus, de tels flux temps-réel (vidéo, voix, audio, ...) peuvent tolérer quelques pertes d'échéances de paquets si cette perte se fait adéquatement. En effet, *Hamdaoui* et *Ramanathan* ont introduit dans [2,3] que les flux temps-réel peuvent tolérer la perte de quelques paquets suivant un profil déterminé, caractérisé par le couple  $(m,k)$  où  $m$  représente le nombre de paquets qui doivent respecter leur échéances parmi  $k$  paquets transmis consécutivement. Donc, on peut tolérer la perte de  $k-m$  paquets dans une fenêtre de taille  $k$  paquets. Cette contrainte définit le modèle de perte pour un flux ayant des besoins temps-réel et admettant la perte occasionnelle et ordonnée de quelques paquets. Cette

contrainte est admissible du fait que les flux temps-réels sont sensibles aux variations du délai plutôt qu'au phénomène de perte.

Cependant, les contraintes (m,k)-firm et l'algorithme d'ordonnancement WFQ ne sont pas directement compatibles. A travers ce papier, on introduit une technique pour adapter cette classe de contraintes aux serveurs à débit garanti, notamment WFQ, et on présente un nouvel algorithme d'ordonnancement pour garantir une bande passante et assurer une plus petite borne sur le délai. Les paquets d'un flux sont divisés en deux classes ; critique et optionnelle selon la contrainte (m,k)-firm sur ce flux. L'algorithme d'ordonnancement que nous présentons tient compte de cette classification pour décider de transmettre ou de rejeter le paquet sélectionné dans l'ordre croissant des temps virtuels de départ.

Le reste de ce papier est organisé comme suit. La section 2 présente un bref aperçu sur la notion des contraintes temporelle (m,k)-firm et les principes de base de WFQ. Dans la section 3 nous présentons l'algorithme (m,k)-WFQ et la section 4 sera dédiée à l'évaluation de performances de cet algorithme et la comparaison avec celles du WFQ standard et enfin nous concluons ce travail dans la section 5.

## 2. Notions préliminaires

### 2.1. Aperçu sur les garanties (m,k)-firm

Les classes des flux ayant des contraintes (m,k)-firm a été introduites dans [2,3]. On dit qu'un système requiert des garanties de type (m,k)-firm, si au moins  $m$  paquets doivent respecter leurs échéances dans n'importe quel fenêtre de  $k$  paquets consécutifs. S'il existe une fenêtre de  $k$  paquets consécutifs ayant plus que  $(k-m)$  paquets ratant leurs échéances, on dit que le système est tombé en échec.

On définit le concept du  $\mu$ -pattern comme étant le profil de perte suivant la contrainte définie par le couple  $(m,k)$ . Il spécifie l'organisation souhaitée des pertes des échéances dans une fenêtre de  $k$  paquets.

Le  $\mu$ -pattern d'un flux ayant une contrainte temporelle (m,k)-firm est défini par la succession de  $k$  éléments de l'alphabet  $\Delta = \{0,1\}$  tel que :

$$\begin{cases} 0 & \text{indique un paquet optionnel} \\ 1 & \text{indique un paquet critique} \end{cases}$$

On représente par  $\mu(i)$  le  $i^{\text{ème}}$  élément du  $\mu$ -pattern pour  $1 \leq i \leq k$ .

En utilisant le concept du  $\mu$ -pattern, on divise les paquets d'un flux en deux classes : optionnelle et critique. Avec cette classification, il suffit de transmettre avec succès les paquets critiques pour satisfaire cette contrainte. Cependant, il reste toujours la possibilité de transmettre des paquets optionnels sans violer cette contrainte. En général, les paquets

optionnels peuvent être transmis ou rejetés suivant les besoins de l'algorithme d'ordonnancement.

Par exemple, un flux ayant des contraintes de type (3,5) peut satisfaire plusieurs  $\mu$ -patterns (10110, 01011, 11010, ...) de longueur 5 et ayant exactement 3 paquets critiques.

De ce fait, le  $n^{\text{ème}}$  paquet d'un flux ayant des contraintes temporelles (m,k)-firm, est considéré comme étant une paquet critique si et seulement si il satisfait la relation suivante :

$$\mu(n \% k) = 1 \quad 2.1$$

avec  $n \% k$  est le reste de la division de  $n$  modulo  $k$ .

L'application de cette classification peut être utile dans le domaine multimédia. En effet, ce concept peut être appliqué à un flux vidéo pour sélectionner les trames critiques dans un GOP (Group of Picture) en utilisant le standard de compression MPEG. Par exemple, un flux compressé utilisant la structure du GOP suivante [IBBPBPIB] peut être considéré comme étant un flux ayant des contraintes temporelles de type (5,8) affecté au  $\mu$ -pattern suivant '1101010'. Ceci dit que la 3<sup>ème</sup>, 5<sup>ème</sup> et 8<sup>ème</sup> trames, qui sont des B-frames, sont considérées comme optionnelles. Par exemple, la 226<sup>ème</sup> trame est considérée comme étant critique car

$$\mu(226 \% 8) = \mu(2) = '1'$$

et la 227<sup>ème</sup> trame est considérée comme étant optionnelle car

$$\mu(227 \% 8) = \mu(3) = '0'$$

Nous précisons que cette technique de classification est statique et initiée par l'application ou l'utilisateur. Cependant, nous pouvons imaginer une classification réalisée dynamiquement par l'ordonnancement suivant un algorithme donné. Ceci peut être avantageux pour optimiser les rejets des paquets, mais il nécessite un traitement additionnel par le serveur.

Dans la suite, on utilisera uniquement une classification statique pour chaque flux.

### 2.2. Principe de l'ordonnancement WFQ

Le principe de WFQ consiste à garantir à chaque flux servi la bande passante fixée par son coefficient de partage  $\Phi$  qui lui est pré-alloué et une borne supérieure sur le temps de réponse. WFQ utilise la notion du temps virtuel pour émuler le comportement du système fluide. Le critère de sélection des paquets par WFQ est le temps virtuel de départ dans le système GPS correspondant [1].

Dans WFQ le temps virtuel de départ est défini par

$$F_i^k = \max\{F_i^{k-1}, V(t)\} + \frac{L_i^k}{\Phi_i} \quad 2.2$$

avec

$F_i^k$  : temps virtuel de départ du  $k^{\text{ème}}$  paquet de l' $i^{\text{ème}}$  flux,

$V(t)$  : le temps virtuel quand le  $k^{\text{ème}}$  paquet arrive,

$\Phi_i$  : le coefficient de partage de l' $i^{\text{ème}}$  flux,

$L_i^k$  : la taille du  $k^{\text{ème}}$  paquet de l' $i^{\text{ème}}$  flux,

$\max\{F_i^{k-1}, V(t)\}$  est considéré comme étant le temps

virtuel du début de service du  $k^{\text{ème}}$  paquet. Il est clair qu'à partir de cette équation, plus le coefficient de partage  $\Phi_i$  est petit, plus le temps virtuel de départ est grand, ce qui fait que les paquets des flux ayant plus de bande passante sont servis à l'avance puisque la sélection se fait dans l'ordre croissant des temps virtuels de départ.

Donc, le délai est étroitement en liaison à la bande passante réservée ainsi que la taille maximale de rafale du flux. Pour un flux  $S_i$  ayant un débit moyen réservé  $r_i$  et une taille maximale de rafale  $b_i$ , le délai garanti par WFQ à ce flux est borné par :

$$D_{i,\max} = \frac{b_i}{r_i} + \frac{L_{\max}}{C} \quad 2.3$$

Où,  $L_{\max}$ , est la taille maximale du paquet parmi tous les paquets dans tous les flux et  $C$  étant la capacité de traitement du serveur. Cette borne augmente avec la taille de rafale et en diminuant le taux de bande passante réservée.

Le but de notre contribution est d'alléger l'effet pointu de la taille de rafale et de la bande passante sur le temps de réponse au pire cas en tenant compte en plus des contraintes temporelles (m,k)-firm et les échéances des flux.

### 3. Présentation de (m,k)-WFQ

Dans cette section, nous présentons notre algorithme (m,k)-WFQ. Nous intégrons dans cet algorithme les contraintes temporelles de type (m,k)-firm. Pour ce fait, nous partons de l'hypothèse que chaque flux demande à l'ordonnanceur WFQ la réservation de la bande passante dont il a besoin et fournit sa contrainte (m,k)-firm ainsi que son modèle de perte souhaité, *i.e.* son  $\mu$ -pattern. Aussi, chaque flux doit proposer à l'ordonnanceur son échéance souhaitée.

L'algorithme est composé de deux parties :

- *Processus d'affectation de priorité,*
- *Processus de service.*

Ces deux processus sont légèrement modifiés par rapport à ceux de WFQ pour être adaptés aux garanties de type (m,k). L'idée est d'ajouter un autre critère de sélection dans WFQ qui fait intervenir le couple (m,k) et le  $\mu$ -pattern de chaque flux. Ce critère indique la priorité des paquets au sein d'un même flux, critique ou optionnel.

Lors de l'arrivée d'un paquet, le processus d'affectation de priorité estampille le paquet reçu par son temps virtuel de départ en appliquant l'équation 2.2 et il marque aussi sa priorité dans le flux (critique ou optionnel) en se basant sur le  $\mu$ -pattern du flux correspondant et en appliquant l'équation 2.1.

Table 1. Algorithme (m,k)-WFQ

#### Entrées

Flux  $S_i = \{(Période \text{ ou Débit}), \text{Echéance Désirée}, (m,k), (Gigue \text{ ou Rafale}), \text{Taille de Paquet}\}$

#### Affectation de priorité

```
A l'arrivée du aième du flux [i] {
  si ( $\mu((a\%k_i)+1)=1$ ) alors {
    Marquer le paquet comme critique;
  }
  sinon {
    Marquer le paquet comme optionnel;
  }

  Calculer le temps virtuel de départ  $F_i^k$ ;

  Estampiller le paquet avec  $F_i^k$ ;
  Mettre le paquet dans sa fille d'attente;
}
```

#### Discipline de Service

```
Serveur = libre;
Tant que (la file est non vide) {
  si (serveur!= occupé) {
    Sélectionner le paquet ayant le plus
    petit temps virtuel de départ
  si (paquet est critique) {
    Envoyer le paquet;
    Serveur = occupé;
  }
  sinon { //Paquet Optionnel
    si (l'échéance serait ratée){
      Rejet du paquet;
      Serveur = libre;
    }
    sinon {
      Envoyer le paquet;
      Serveur = occupé;
    }
  }
}
si (serveur== occupé) {
  attendre jusqu'à transmission
totale
du paquet;
Serveur= libre;
}
}
```

Le processus de service est activé quant au moins un paquet existe dans la file d'attente du système. Le serveur sélectionne le paquet ayant le plus petit temps virtuel de départ parmi tous les paquets critiques présents en tête de leur files. Si aucun paquet critique existe, le choix sera fait parmi les paquets optionnels. Après, si le paquet sélectionné est critique, il est

transmis directement par le serveur, sinon, si le paquet est optionnel, l'ordonnanceur vérifie avant la transmission si ce paquet pourrait éventuellement satisfaire son échéance. Si l'échéance souhaitée est ratée, le serveur rejette le paquet et refait une nouvelle sélection, sinon, il l'envoie.

L'avantage de l'algorithme proposé est qu'il permet de garantir une bande passante à un flux tout en intégrant les propriétés temporelles dans le processus d'ordonnement ce qui résulte à gérer les flux plus efficacement. En effet, le rejet des paquets optionnels qui ne satisfont pas leurs échéances permet au serveur de donner la main plus rapidement aux paquets critiques en attente. Cette perte ne dégrade pas les performances des flux servis tant que leurs contraintes (m,k)-firm sont satisfaites. Ainsi, (m,k)-WFQ diminue forcément les bornes sur les temps de réponse des flux temps réels par rapport à WFQ standard.

#### 4. Evaluation de performances

Dans cette section, nous présentons une étude par simulation pour évaluer les performances de l'algorithme d'ordonnement proposé dans le paragraphe précédent.

##### 4.1. Présentation du système

La topologie du réseau est constituée de trois sources de trafic comme le présente la figure Fig. 4.1.

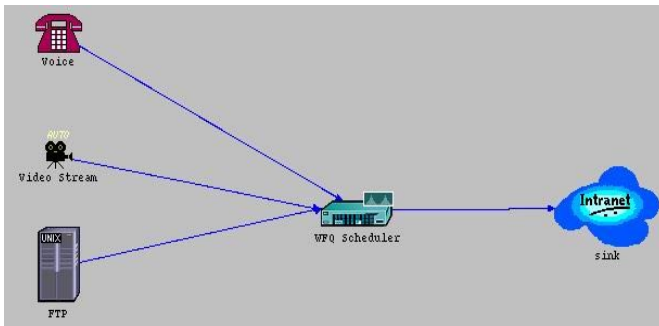


Fig. 4.1. Topologie du système

Ces trois sources partagent un lien de 10 Mb/s selon leurs coefficients de réservation. Dans cette simulation, on considère une taille fixe à tous les paquets des trois flux de 8 Kbits.

La première source génère un flux de voix selon le modèle de trafic ON/OFF [5]. Les périodes d'activité ON et de silence OFF sont exponentiellement distribuées avec les moyennes  $1/\mu_{ON} = 500ms$  et  $1/\mu_{OFF} = 755ms$  avec une période de génération de paquets dans la période d'activité de 50 ms. Donc, le débit moyen du flux est de 64 Kb/s. Les contraintes temporelles sont de type (4,5) et l'échéance souhaitée d'un paquet est fixée à 10 ms.

La deuxième source est une source CBR (Constant Bit Rate) qui génère un flux vidéo pseudo-périodique de 2 Mb/s. L'échéance des paquets est fixée à 4 ms avec une garantie de type (3,5).

La troisième source est un agrégat de flux FTP qui consomme le reste de la bande passante ayant donc un débit de 7,936 Mb/s. Ce trafic fonctionne en mode Best-Effort. Donc, il ne possède pas de propriétés temporelles strictes comme dans le cas des deux sources temps-réel : Voix et Vidéo. Par conséquent, nous fixons une garantie de type (0,1) pour le flux FTP. Pour cela, tous les paquets issus de cette application sont considérés comme optionnels.

Cependant, nous configurons l'ordonnanceur (m,k)-WFQ de ne rejeter que les paquets optionnels des sources temps-réel, voix et vidéo. En effet, le flux FTP est vulnérable à la perte de paquets. Ainsi, comme tous les paquets de ce flux sont optionnels, le serveur est averti de faire passer tous ses paquets. Ceci est réalisé en mettant une échéance infinie pour les flux best-effort.

Le tableau suivant récapitule les paramètres de Simulation pour chacun des flux :

Table 2. Paramètres de Simulation

	(m,k)	Débit	Trafic	$\mu$ - pattern	Echéance Désirée
<b>Voix</b>	(4,5)	64 kb/s	ON/OFF (500/755/50)ms	11011	10 ms
<b>Vidéo</b>	(3,5)	2Mb/s	Pseudo Périodique ~2Mb/s	10110	4 ms
<b>FTP</b>	(0,1)	7,936 Mb/s	Pseudo Périodique ~7.936 Mb/s	0	Inf

Nous simulons ce système avec WFQ et (m,k)-WFQ pour une étude comparative.

##### 4.2. Etude comparative

Nous avons utilisé le simulateur à événements discrets OPNET/MODELER [4]. L'expérience consiste à simuler l'ordonnement du trafic généré par les trois sources précitées avec les serveurs WFQ standard et (m,k)-WFQ.

Le tableau 3 montre les bornes mesurées sur le temps de réponse des paquets pour chacun des flux fournies par les deux serveurs.

Table3. Bornes sur les délais (ms)

	(m,k)-WFQ	WFQ
<b>Voix</b>	9,769	4776,83
<b>Vidéo</b>	3,999	41,084
<b>FTP</b>	3,837	18,048

Comme prévu, (m,k)-WFQ fournit une garantie plus étroite sur le délai pour les flux temps-réel. Dans ce scénario, il est remarquable que le délai maximal garanti par WFQ au trafic de la voix soit assez grand. Ce résultat découle de deux facteurs majeurs (cf. équation 2.3):

- Le faible taux de bande passante réservée.
- La taille importante de la rafale.

Cependant, l'algorithme (m,k)-WFQ permet de réduire considérablement les bornes sur les temps de réponse en sacrifiant quelques paquets optionnels selon les contraintes temporelles (m,k)-firm de chaque flux. Le rejet des paquets optionnels ne satisfaisant pas leurs échéances, améliore nettement le délai des paquets critiques.

D'autres part, le modèle (m,k)-firm permet de distinguer entre les flux temps-réel et les flux best-effort. En effet, selon les contraintes (m,k) que nous avons défini, l'ordonnanceur (m,k)-WFQ sert les flux temps-réel avant le flux FTP puisque le critère de sélection primaire se fait sur les paquets critiques et nous avons défini les paquets du trafic FTP comme étant tous optionnels. Donc, (m,k)-WFQ assure le service des paquets à contraintes temporelles avant ceux du flux best-effort. Ce qui n'est pas le cas avec WFQ standard car il ne peut pas distinguer entre les différents types de flux servis.

Les bornes sur les délais dépendent de l'échéance requise de chaque flux. En effet, nous avons changé l'échéance requise pour le trafic vidéo à  $D_{vidéo} = 40ms$  (2<sup>ème</sup> scénario). Nous obtenons les résultats suivants :

*Table 4. Bornes sur les délais (ms)*

(2 <sup>ème</sup> Scénario)	
(m,k)-WFQ	
<b>Voix</b>	3,177
<b>Vidéo</b>	37,621
<b>FTP</b>	26,111

Nous observons que la borne du temps de réponse du flux de la voix a diminué par rapport au 1<sup>ère</sup> scénario et la borne sur le délai du trafic vidéo a augmenté mais reste toujours inférieur à son échéance désirée. Ce résultat est étroitement lié aux taux de rejet. Le tableau 5 montre les taux de rejets pour chacun des flux temps-réel dans les deux scénarios.

*Table 5. Taux de rejets (ms)*

	1 <sup>er</sup> Scénario	2 <sup>ème</sup> Scénario
<b>Voix</b>	6,8%	15%
<b>Vidéo</b>	5,5%	0%

Nous remarquons que dans le deuxième scénario le taux de rejet des paquets vidéo est nul. C'est à dire, tous les paquets critiques et optionnels de ce flux sont

servis, ce qui explique la hausse de la borne sur le délai du trafic FTP, dont tous les paquets sont optionnels. En effet, dans le 1<sup>er</sup> scénario le rejet des paquets optionnels ayant dépassés leurs échéances du flux vidéo accélère l'envoi des paquets du flux « *gourmant* » de FTP. Cependant, la diminution mesurée sur la borne du délai dans 2<sup>ème</sup> scénario pour le trafic de la voix, s'explique par l'augmentation du taux de rejet des paquets optionnels relatifs à ce flux de 6,8% à 15%. Cette augmentation est légère et la simulation montre que la contrainte temporelle (4,5)-firm est satisfaite dans les deux scénarios, puisque dans les deux cas la borne mesurée sur le délai est inférieure à celle demandée.

Les temps moyens d'attente dans les files des trois flux ordonnancés par (m,k)-WFQ et WFQ standard sont montrés par les figures Fig. 4.2 et Fig. 4.3 respectivement.

On note que l'irrégularité du trafic généré par la source de la voix (trafic en rafale) a un effet négatif sur les temps d'attente avec WFQ standard tandis que ce temps est plus stable avec l'algorithme (m,k)-WFQ. Ce dernier algorithme lisse parfaitement l'effet des rafales et garantit des délais moyens plus courts. WFQ standard pourrait satisfaire un délai plus court au trafic de la voix en lui réservant une bande passante plus grande que son débit moyen. Cependant, cette technique n'est pas efficace car ce gain dégrade les temps de réponse des autres flux temps-réel. L'apport de l'algorithme (m,k)-WFQ est de garantir des bornes plus petites sur les délais de tous les flux servis sans favoriser le service d'un flux sur un autre. L'équité de (m,k)-WFQ est garantie sur le plan de la bande passante et aussi, sur les contraintes temporelles des flux.

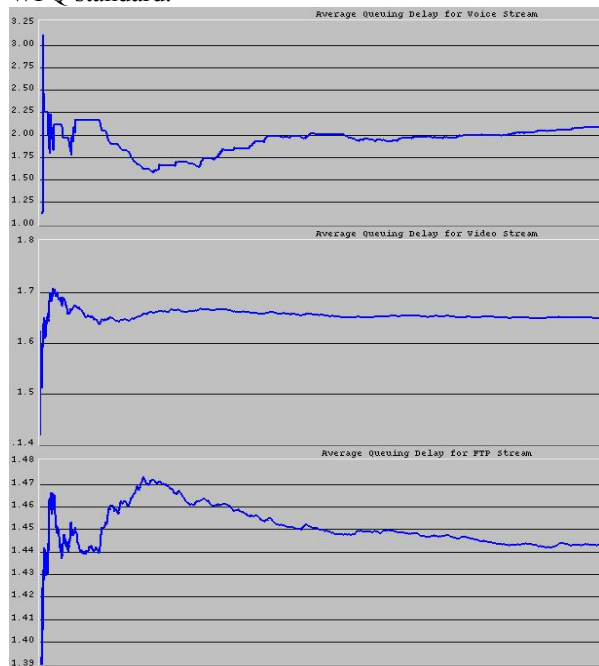
## 5. Conclusion

Nous avons présenté dans ce papier une nouvelle proposition pour l'ordonnancement équitable à débit garanti appelée (m,k)-WFQ. Cet algorithme intègre les propriétés temporelles de type (m,k)-firm dans le processus d'attribution de priorité et de service de WFQ standard.

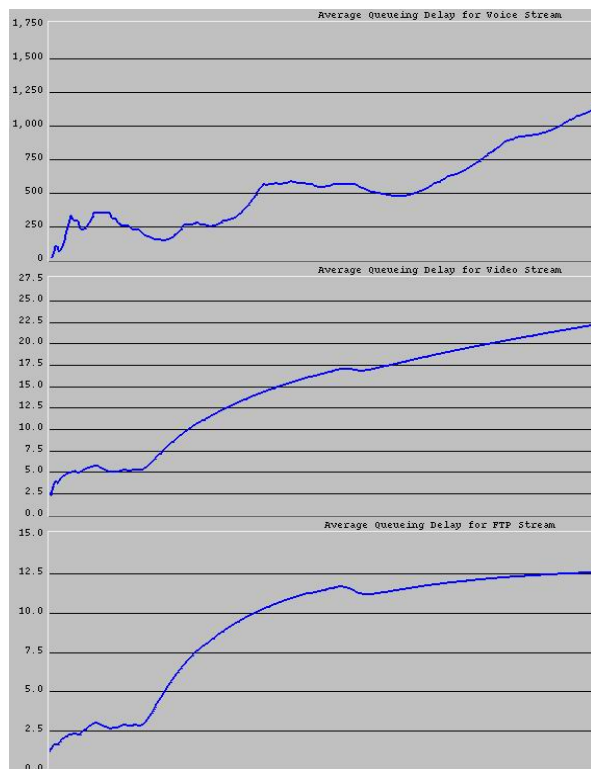
Aussi, nous avons analysé dans l'étude comparative par simulation les performances de (m,k)-WFQ vis à vis de celles de WFQ standard. Nous remarquons une amélioration sur les bornes du temps de réponse des flux temps-réel. Ceci résulte du rejet des paquets optionnels de ces flux quand ils dépassent leurs échéances. Le taux de rejet par cette politique est très satisfaisant et garanti les contraintes de type (m,k) de chacun des flux temps-réel.

Cependant, il reste à prouver cette performance par une étude analytique. Nous sommes en train de développer une étude analytique en utilisant la théorie de *Network Calculus* [6] pour fournir analytiquement des bornes déterministes sur le délai et prouver la

réduction des délais comparés avec ceux fournis par WFQ standard.



**Fig. 4.2. Temps moyen d'attente avec (m,k)-WFQ (1<sup>er</sup> scénario)**



**Fig. 4.3. Temps moyen d'attente avec WFQ**

## 6. Bibliographie

- [1] A.Demers, S.Keshav, S.Shenker, "Analysis and Simulation of Fair Queuing Algorithm", Proceeding ACM SigComm 89, pp 3-12, 1989
- [2] M. Hamdaoui and P. Ramanathan. "A dynamic priority assignment technique for streams with (m, k)-firm deadlines". *IEEE Transactions on Computers*, 44(4), 1443–1451, Dec.1995.
- [3] P. Ramanathan "Overload Management in Real-Time Control Applications Using (m,k)-Firm Guarantees" *IEEE Transactions on Parallel and Distributed Systems* Vol 10 No 6, p549-559, June 1999
- [4] OPNET Technologies / MODELER '[www.opnet.com](http://www.opnet.com)'
- [5] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 833--846, Sept. 1986.
- [6] JY Le Boudec , P Thiran, "Network Calculus : A Theory of Deterministic Queueing Systems for the Internet" Springer Verlag July 2002
- [7] Anis KOUBAA, Aref Jarraya, Ye-Qiong SONG "SBM protocol for providing real-time QoS in Ethernet LANs" 1<sup>st</sup> Intl. Workshop on Real-Time LANs in the Internet Age (Satellite Event to ECRTS'02) Austria, Juin 2002
- [8] G. Bernat , A. Burns. And A. Lamosi "Weakly Hard Real-Time Systems", *IEEE Transactions on Computers*, 50(4),pp.308-321,2001.