Loss-Tolerant QoS using Firm Constraints in Guaranteed Rate Networks

Anis KOUBAA,
LORIA Research Laboratory
National Polytechnic Institute of Lorraine
akoubaa@loria.fr

Ye-Qiong SONG
LORIA Research Laboratory
Henri Poincaré University, Nancy I
song@loria.fr

Abstract

Fair Queuing scheduling algorithms, such as Weighted Fair Queueing (WFQ) and its variants, are intended to provide hard real-time guarantees by making bandwidth reservation. However, hard guarantees are based on pessimistic estimation that leads to underutilize network resources. We propose in this paper a trade-off between hard and soft real-time guarantees to maintain an acceptable QoS guarantee in overload condition and maximize efficiently the utilization of network resources. The key of our solution is that many real-time applications are loss-tolerant, but the loss profile must be well defined since successive packet losses are not suitable. We use the concept of (m,k)-firm timing constraints to define a novel guaranteed loss-tolerant QoS. Therefore, we extend the basic WFQ algorithm to take into account the firm timing constraints to provide lower delay guarantees without violating bandwidth fairness or misusing network resources. The proposal is called (m,k)-WFQ. Using Network Calculus formalism, analytic study gives the deterministic delay bound provided by the (m,k)-WFQ algorithm for upper bounded arrival curve traffic. Theoretical results and simulations show a noticeable improvement on delay guarantee made by (m,k)-WFQ compared to standard WFQ algorithm without much degrading bandwidth fairness.

1 Introduction

Bandwidth guarantee has been widely used in Internet Quality of Service (QoS) architecture as well as ATM networks to make deterministic real-time guarantees for time-sensitive applications. However, hard guarantee is not without cost since it requires deterministic predictability on network delays and thus underutilizes network resources. Soft real-time guarantee, on the other hand, maximizes resources utilization but has less stringent guarantee and QoS may degrade drastically in overload condition.

For real-time applications the QoS important metrics are delay and loss. To ensure short delay and loss free (hard) guarantee, the peak-rate bandwidth reservation is commonly used especially for variable bit-rate (VBR) flows whose carried burst size is quite large. This kind of reservation overstates the bandwidth requirement and reduces network utilization. A second approach consists in making average-rate bandwidth reservation to maximize resource utilization but the guaranteed delay may be larger than the application requirement when the burst size is quite important resulting in many deadline misses.

WFQ [10] and its variants [11,12,13] are the basic algorithms used to make bandwidth guarantees. However, several studies have been interested in improving the delay guarantee provided by WFQ for real-time flows with high burst size. In [2], Wang et *al.* proposed a technique called PWFQ that combines fixed-priority assignment and WFQ scheduling algorithm in order to better manage the delay bounds for various flow sessions. The problem to resolve was how to guarantee short delays for flows with low service share. The outcome of this method is to reduce the upper bound on delay for low service-share streams by assigning them higher priorities within a *sliding window*. A window defines a (virtual) time interval in which all packets whose virtual finish times fall into are considered to have a similar finish time. The server selects the highest priority packet to be served within the sliding window. This technique leads to decouple the delay from the service share and guarantees better delay for low share flows without much degrading the delay of other flow sessions, but the choice of the optimal window size might be complex. In [7], authors proposed the Burst-based WFQ (BWFQ) algorithm that generalizes WFQ to schedule bursts instead of individual cells in an ATM switch. This extension of WFQ provides lower delay guarantee for delay-sensitive applications.

The main problem addressed in this paper is to provide short delay guarantees for bursty real-time flows in guaranteed-rate networks without underutilizing network resources by using the loss tolerance property of some real-time applications such as multimedia streams and embedded system applications. The study on the MPEG video flow presented in [9] stated that the effect of the packet loss on QoS of the application depends on when and how the loss occurs. The loss could be tolerated if it occurs occasionally and not successively. The (m,k)-firm model was first proposed in [4] to express the deadline miss tolerance schema of a real-time application and is now interesting recent research trends [3,14] to take advantage of this technique to define new QoS metrics. In [14], Striegel has proposed, in the context of media servers, a Dynamic Class-based Queue Management (DCQM) scheme that creates or terminates groups into a class of service by multiplexing individual streams of the same class into a group with similar loss-tolerance characteristics or (m,k)-firm constraints.

Our contribution is to define a loss-tolerant QoS guarantee by integrating the (m,k)-firm constraints into WFQ algorithm. This current study extends the work made in [1] and we analytically evaluate the delay bound for VBR traffic expressed by a two-segment curve as deployed in IntServ QoS model as well as ATM networks. We also simulate a multimedia network to show the performance of our proposal to provide low delay guarantee and selective loss profile with perfect use of network resources.

The rest of the paper is organized as follows. Section 2 gives an overview of the (m,k)-firm model. Section 3 shows the (m,k)-WFQ algorithm. Section 4 presents the delay bound analysis of (m,k)-WFQ and extend results for a VBR flow bounded by a two-segment curve. Simulations are presented in section 5. Section 6 concludes the paper.

2 Overview of (m,k)-Firm Constraints

A stream is said to have (m,k)-firm requirement if at least *m* packets inside any window of *k* consecutive packets must meet their required deadline. If more than *k-m* deadline misses occur in a specified window, the stream is said to be in *dynamic failure state*.

We define the concept of *κ-pattern* that specifies the organization of deadline miss/meet within a window of *k* packets. The κ-pattern of a stream having (m,k)-firm deadline requirement is the succession of k elements from the alphabet $\Delta = \{O, M\}$ where :

$$\begin{cases} 'O' & \text{Stands for an Optional packet} \\ 'M' & \text{Stands for a Mandatory packet} \end{cases}$$

and contains exactly m 'M' symbols. κ(i) denotes the i[th] element of the κ-pattern for $1 \le i \le k$.

Using *κ*-patterns, the stream's packets are divided into *optional* and *mandatory* parts. To satisfy the (m,k)-firm constraint of a flow, it is sufficient that all mandatory packets meet their deadlines. Optional packets could be dropped or transmitted depending on the scheduling algorithm. The miss of all or some optional packets would not affect the temporal QoS of the stream. Hence, a *loss-tolerant QoS* consists in guaranteeing the deadline meet of all mandatory packets.

The n[th] packet of a stream is classified as mandatory when $\kappa(n\%k) = 'M'$ for $n = 1, 2, ...$ where % is modulus operator (remainder after division).

This technique may be used in different contexts. In [5], the classification into mandatory and optional parts has been used to enhance the Rate Monotonic scheduling process. Also in multimedia, this concept could be applied to select mandatory frames from a group of picture (GOP) using MPEG compression standard. For instance, if a stream has a GOP structure IBBPBBPB, it could be considered as a (3,8)-firm flow and assigned a *κ*-pattern such $\kappa = 'MOOMOOMO'$. This says that all B-frames are considered as optional. Scheduler should take more care of I and P frames since they are mandatory.

Also, for an audio stream, that can tolerate no more than one deadline miss, could be assigned a (2,3)-firm guarantee with a *κ*-pattern $\kappa = 'MMO'$.

The specification of a *κ*-pattern could be initiated by the real-time application that knows its loss tolerance schema. This is simply a way among several to classify packets according to (m,k)-firm constraints. This classification is static. However, we could imagine a dynamic classification made by the server according to a given algorithm. Dynamic classification is out of the scope of this current study.

3 (m,k)-Weighted Fair Queueing

(m,k)-WFQ is an extension of WFQ scheduling algorithm that integrates (m,k)-firm constraints and takes advantage of packet classification into mandatory and optional parts.

The standard WFQ scheduling algorithm is based on the computation of virtual finish time to emulate the fluid GPS system. The virtual finish tag of a packet is defined as:

$$F_i^k = \max\left\{F_i^{k-1}, V(t)\right\} + \frac{L_i^k}{\Phi_i} \qquad \textbf{(1)}$$

where F_i^k is the virtual finish time of k^{th} packet of stream *i*. $V(t)$ is the virtual time when k^{th} packet arrives, L_i^k is the packet size of k^{th} packet and Φ_i is the service share weight. This value is tagged into the packet. Then, the scheduler selects the packet with lowest finish tag. This tag doesn't consider any temporal constraint. It depends only on service share weight Φ_i and packet length L_i^k.

However, (m,k)-WFQ scheduling algorithm repairs this lack by further considering the (m,k)-firm deadline of each stream as well as the packet classification. The proposed algorithm fosters the transmission of mandatory packets to guarantee their deadline meet. In fact, (m,k)-WFQ scheduler makes the selection of the packet with lowest finish tag among mandatory packets present at the head of active queue of each served stream. Otherwise, *i.e.* -no mandatory packet is present at the head of queues- the optional packet with lowest F_i^k is then picked out for service. If the selected packet is mandatory, the (m,k)-WFQ server sends it immediately, else, for optional packet, the scheduler checks whether its deadline is missed, and then drops it, or met and then it is transmitted. Figure 1 depicts the (m,k)-WFQ scheduling process.

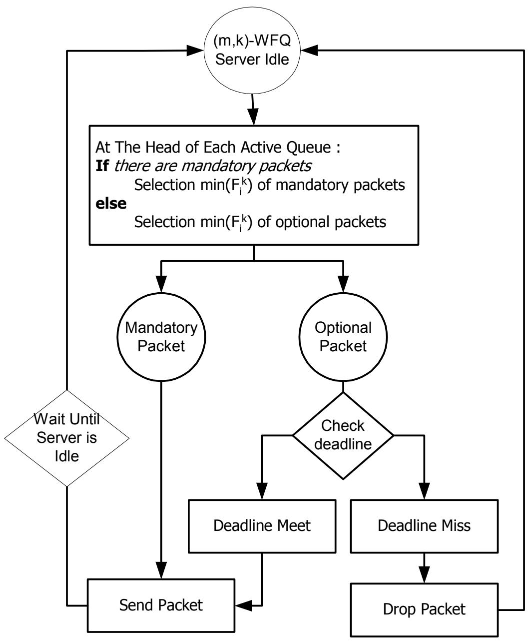


Fig. 1. (m,k)-WFQ Scheduling Algorithm

The choice of the deadline to drop optional packets has an important impact on the behavior of (m,k)-WFQ scheduler. In fact, when the specified deadline to drop optional packets gets shorter, there are increasing chances to drop optional packets and, consequently, to send in faster way mandatory packets.

The analytic model that we describe in section 4, shows an efficient technique to adjust adequately this deadline to guarantee the required delay for mandatory packets.

The advantage of (m,k)-WFQ is to make a trade-off between two QoS metrics : (1) bandwidth fairness (2) delay guarantee since it considers simultaneously these two requirements in the scheduling process.

4 Delay Bound Analysis

In this section we present the analysis of the delay bound guaranteed by a (m,k)-WFQ scheduler using Network Calculus Formalism [6]. For this purpose we introduce the concept of (m,k)-filtering to adapt Network Calculus to the use of (m,k)-firm constraints.

4.1 The Filtering Theory

Assume that we have a flow with cumulative arrival function $R(t)$ and has a (m,k)-firm deadline requirement. Let κ be the κ-pattern of this flow.

Definition 1. (m,k)-Filter device
We define a (m,k)-filter as a device, that for an arrival function $R(t)$, makes the output $\widetilde{R}(t)$ where only mandatory packets of the corresponding flow are sent according to its κ-pattern. Optional packets are discarded.

At once, if $m=k$, the output is exactly equal to the input. For $m<k$, $\widetilde{R}(t)$ is the cumulative number of mandatory packets within the interval [0,t].

Firstly, we assume that each flow has a constant packet size denoted by L. Note that $R(t)$ denotes the input in terms of number of packets. To have the number of bits (fluid model), we just need to multiply this quantity by the packet size L.

Theorem 1.
Consider a constant packet-size flow with a cumulative arrival function R(t) and a κ-pattern κ that crosses an (m,k)-filter. $\widetilde{R}(t)$ is the output of the (m,k)-filter if and only if,

$$\widetilde{R}(t) = m \cdot \left\lfloor \frac{R(t)}{k} \right\rfloor + \Pi\left(R(t)\right)$$

$$with\ \Pi\left(R(t)\right) = \sum_{n=\left(\left\lfloor \frac{R(t)}{k} \right\rfloor . k\right)+1}^{R(t)} \overline{\kappa}(n\%k)$$

$$and\ \forall 1 \le i \le k \quad \overline{\kappa}(i) = 1 \ \ if\ \ \kappa(i) = 'M' \ \ else\ \overline{\kappa}(i) = 0$$

Refer to appendix 1 for the proof. In general case, when the packet size is variable, we assume that there exists a constant λ_M that represents the ratio of mandatory bits into the window of k consecutive packets according to its κ-pattern. It is simple to verify that, for $t \in \mathrm{T} = \{t_0, t_k, t_{2k}, ..., t_{nk}, ...\}$ where t_{nk} is $(nk)^{th}$ packet arrival time, $\widetilde{R}(t) = \lambda_M R(t)$.

As an example of this assumption, if we mark as mandatory the I and P frames in an MPEG stream, λ_M represents the ratio of mandatory bits (of I and P frames) within a Group of Picture (GoP). For the case of a constant packet size, $\lambda_M = \frac{m}{k}$.

The following theorem gives the arrival curve of a (σ,ρ)-bounded stream that crosses an (m,k)-filter. We do remind that a stream with an arrival function $R(t)$ is said to be upper bounded by the typical arrival curve $(\sigma + \rho.t)$ if $R(t) - R(s) \le \sigma + \rho(t - s)$, $\forall\ 0 \le s \le t$.

Theorem 2. Application to a leaky bucket stream.
Consider a stream S with arrival function $R(t)$ upper constrained by the arrival curve $\alpha(t) = \sigma + \rho.t$ and crosses an (m,k)-filter device. Set λ_M the ratio of mandatory packets into the window of k consecutive packets according to its κ-pattern.
The output produced by the (m,k)-filter is bounded by the arrival curve $\widetilde{\alpha}(t) = \widetilde{\sigma} + \widetilde{\rho}.t$ where

$$\begin{cases} \widetilde{\sigma} = \lambda_M \cdot \sigma \\ \widetilde{\rho} = \lambda_M \cdot \rho \end{cases}$$

and $t \in \mathrm{T} = \{t_0, t_k, t_{2k}, ..., t_{nk}, ...\}$ where t_{nk} is $(nk)^{th}$ packet arrival time. We call this curve as the minimal arrival curve of the stream.

Refer to appendix 2 for the proof.

4.2 Delay Bound for a Leaky Bucket Constrained Flow

In [1], using results of the filtering theory, we have determined the delay bound for a flow constrained by one-segment linear arrival curve $\alpha(t) = \sigma + \rho.t$. We denote by λ_M (λ_O) the ratio of mandatory (optional) bits into a window of k consecutive packets. Then, for a bandwidth reservation $R \ge \rho$ made by the (m,k)-WFQ scheduler, the delay bound is :

$$D^*_{\max} = \lambda_M \cdot \frac{\sigma}{R} + \lambda_O \cdot \frac{b}{R} + \frac{L_{\max}}{C} \qquad \textbf{(2)}$$

Where C denotes the server capacity and b denotes the maximum burst size of optional packets eligible for serving. This burst is defined by $b = D_{op}.\rho$ and represents all optional packets that do not violate their required deadline D_{op} . D_{op} is less or equal to the delay requirement of the flow. $L_{\max}$ is the maximum packet size among all served streams.

It is easy to observe that this delay bound is lower than that given by the standard fair queuing scheduler:

$$D^*_{stdFQ} = \frac{\sigma}{R} + \frac{L_{\max}}{C} \qquad \textbf{(3)}$$

Since $b \le \sigma$.

Hence, for the same resources reservation, we have better quality of service in terms of delay guarantee. With (m,k)-WFQ, it is easy to control the loss tolerance by skipping transmission of optional packets in overload situation to foster the transmission of mandatory ones.

4.3 Delay Bound of VBR flows

In this section we generalize the results made in [1] and evaluate the delay bound of VBR flows typically expressed by the two-segment linear curve $\alpha(t) = \min(M + p.t, b + r.t)$ where M is the maximum packet length, p is the peak-rate, b is the maximum burst size and r is the average long-term rate. This representation of VBR is commonly used in IntServ QoS model as well as AMT networks.

Let κ denotes the κ-pattern of the VBR flow and λ_M (λ_O) be the ratio of mandatory (optional) bits into the window of k consecutive packets according to its κ-pattern.

To determine the delay bound we need to estimate the arrival curve of the *effective flow* transmitted by the scheduler. The curve of the effective flow includes all mandatory bits and the maximum number of optional bits transmitted by the scheduler.

So, the main problem is to compute efficiently the effective flow curve from the two-segment arrival curve of the VBR flow and its timing constraints.

4.3.1 The Effective Flow Curve

According to (m,k)-WFQ scheduling algorithm, all optional packets, whose deadline exceeds the desired value, are dropped. Note by δ the desired deadline. Then, the burst size of optional packets cannot be larger than $\sigma = \delta.r$ since *r,* the average long-term rate*,* is the minimum reserved bandwidth. The effective flow model is presented by figure 2.

The mandatory part of *R(t)* is the output of the (m,k)-filter $R_1^*(t)$. The optional part of *R(t)* is obtained when the flow crosses the *(k-m,k)-filter* according to the *reverse κ-pattern* of the stream. The output is denoted by $R_2^*(t)$. Finally, to get the maximum number of optional packets processed by the scheduler (not dropped), the flow $R_2^*(t)$ is shaped by a $\lambda_O(\sigma,r)$ leaky bucket controller to select only optional packets whose deadlines are lower than σ/r. The output $R^*(t)$ represents then the effective flow. Denote by $\alpha^*(t)$ its arrival curve.

We show that, using *Theorem 2* and results in chapter 3 in [6] :

$$R^*(t) \sim \alpha^*(t) = \min\begin{pmatrix} M + p.t, \\ (\lambda_M M + \lambda_O.\sigma) + (\lambda_M p + \lambda_O r).t, \\ (\lambda_M b + \lambda_O.\sigma) + r.t \end{pmatrix} \quad \textbf{(4)}$$

The proof is presented in Appendix 3.

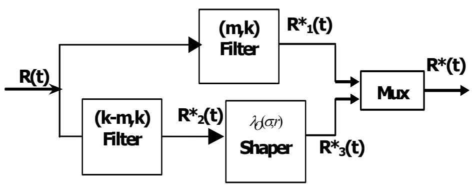
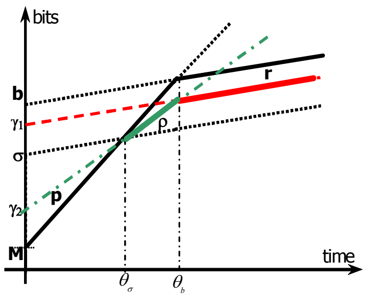


Fig. 2. Effective Flow Model

We denote by

$$\theta_\sigma = \frac{\sigma - M}{p - r}, \quad \theta_b = \frac{b - M}{p - r}, \quad \rho = \lambda_M p + \lambda_O r$$

$$\gamma_1 = \lambda_M b + \lambda_O \sigma, \quad \gamma_2 = \lambda_M M + \lambda_O.\sigma$$



Effective Flow Arrival Curve

From the figure above, the effective flow curve is made of three segments as expressed by equation 4. When time interval $[\theta_\sigma, \theta_b]$ is very short, a good approximation of this arrival curve is:

$$R^*(t) \sim \min\left(M + p.t, (\lambda_M b + \lambda_O \sigma) + r.t\right) \quad \textbf{(5)}$$

The outcome of this curve is to take into account the timing constraints of the flow. In fact, λ_M and σ stand for the (m,k)-firm and the deadline requirements, respectively. The value of the burst σ is adjustable to control the dropping process of optional packets and then enforce the required delay bound for mandatory packets.

4.3.2 Delay Bound Analysis

We assume that a service curve $\beta_{R,T}(t) = R.(t - T)$ is guaranteed to the effective flow. We propose to derive the delay bound experienced by the effective flow.

If we consider the approximated curve, a direct result of the delay bound is:

$$D_{\max} = \frac{M}{R} + \frac{(\lambda_M b + \lambda_O \sigma) - M}{R}\left(\frac{p - R}{p - r}\right)^+ + T \quad \textbf{(6)}$$

A finer bound is obtained by considering the three segments of figure 3 as a flow arrival curve.

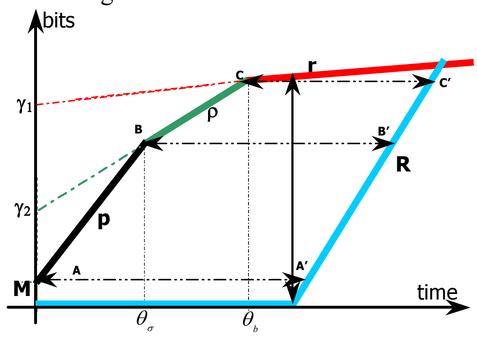


Fig. 3. Computation of Delay Bound

The guaranteed delay bound is represented by the maximum horizontal deviation between the arrival curve $\alpha^*(t)$ and the service curve $\beta(t)$. The maximal horizontal distance is reached at an angular point. In the figure it is either marked as AA', BB' or CC'. Then, the delay bound is given by :

$$D_{\max} = \max\left[\frac{M}{R}, \frac{\alpha(\theta_\sigma)}{R} - \theta_\sigma, \frac{\alpha(\theta_b)}{R} - \theta_b\right] + T$$

using Max-Plus algebra we can rearrange this formula as follows :

$$D_{\max} = \max\begin{bmatrix} \frac{M}{R} + \left(\frac{\sigma - M}{R}\right)\left(\frac{p - R}{p - r}\right)^+, \\ \left(\frac{(\lambda_M M + \lambda_O \sigma)}{R} + \left(\frac{b - M}{R}\right)\left(\frac{\rho - R}{p - r}\right)\right) \end{bmatrix} + T \quad \textbf{(7)}$$

It can be shown that the delay given by equation 7 is always lower than that approximated in equation 6.

We show that those delay bounds provided by the (m,k)-WFQ scheduler are lower than that provided by standard WFQ reservation expressed in [6] by:

$$D_{stdFQ2} = \frac{M}{R} + \frac{b - M}{R}\left(\frac{p - R}{p - r}\right)^+ + T \quad \textbf{(8)}$$

Moreover, from equation 6 or 7, we can adjust the deadline to drop optional packets in order to make D_{max} equal to the required delay.

Thus, according to equation 6, for example, to make delay guarantee for mandatory packets no more than the required delay D_{req}, the maximum allowed optional packet burst-size eligible for serving is:

$$\sigma = \left(\frac{R}{\lambda_O}\left(\frac{p-r}{p-R}\right)^+\right).\left(D_{req} - \left(\frac{M}{R} + \frac{\lambda_M b - M}{R}\left(\frac{p-R}{p-r}\right)^+ + T\right)\right) \quad \textbf{(9)}$$

From equation 7, after some computations, the maximum allowed optional packet burst-size eligible for serving is:

$$\sigma = \min\left\{\frac{R.\left(D_{req}-T\right) - M\left(1-\frac{(p-R)^+}{(p-r)}\right)}{\frac{(p-R)^+}{(p-r)}}, \frac{R.\left(D_{req}-T\right) - \lambda_M M + (b-M)\left(\frac{\rho - R}{p-r}\right)}{\lambda_O}\right\} \quad \textbf{(10)}$$

Hence, the maximum delay to serve an optional packet is not more than $\delta = \sigma.r$.

Therefore, integrating (m,k)-firm timing constraints provides an important flexibility for real-time applications and makes guarantees on both bandwidth and delay with efficiently maximizing network resources.

5 Performance Evaluation

In this section we present a case study based on simulation of a multimedia network for voice, audio and video delivery depicted by figure 4. We use OPNET simulator [16] for this study.

The scenario consists in different multimedia sources and best-effort flow as background traffic.

The main goal is to evaluate the performance of our proposed scheduling algorithms by comparing the standard WFQ scheduling technique and on the other side, enhanced (m,k)-WFQ scheduling technique that considers as well packet classification into mandatory and optional parts - in terms of the following metrics (1) end-to-end delay, (2) loss ratio.

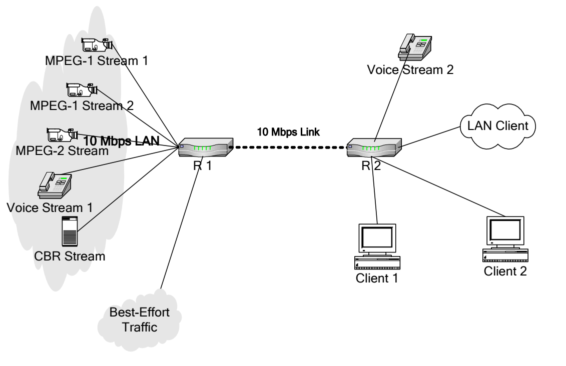


Fig. 4. Simulation Scenario

Table 1 summarizes the total workload.

Table. 1. Simulation Workload

	Rate	Traffic Model			κ-pattern or GoP Structure
		I-size (Kb)	P-size (Kb)	B-Size (Kb)	
MPEG-1 Stream1	1Mb/s	Log(129,20)	Log(53,10)	Log(11,2)	IBBPBBPBBPBBPB B
MPEG-1 Stream2	1Mb/s	Log(129,20)	Log(29,10)	Log(11,3)	IBPBPBPBPB
MPEG-2 Stream3	2Mb/s	Log(156,10)	Log(80,20)	Log(27,1)	IBPBPBPBPB
Voice Stream1	64 kb/s	ON/OFF (Exp(352)/Exp(650)/Const(20)) ms			MOMOM
Audio-CBR Stream	2Mb/s	CBR traffic / Packet Size (6Kb) / Inter-arrival (3ms)			MOOM
Best-Effort	3.936 Mb/s	Self Similar Traffic : Multiplexed pareto distributed ON/OFF sources			Scenario 2: 'O' Scenario 3: 'M'

The video sources produce MPEG streams, that approximate the measurement of actual streams used in [9], according to the following stochastic model [17,18,19]. Each stream generates I-frames, P-frames and B-frames according to a fixed GoP structure at a specific average rate of 30 frames/second. A large frame is fragmented into packets to adapt it to the maximum transmission unit (MTU) of the router. Frame length of each frame type is lognormal-distributed [17,18], denoted $Log(u,v)$ where u is the average frame size and v is the variance. In this simulation study, we consider that all B packets are optional whereas I and P packets are mandatory.

The voice source is modeled using a bursty ON-OFF model with standard parameters [15]. Specifically, the ON and the OFF times are exponentially distributed with means 352 ms and 650 ms respectively. When a stream is in the ON state the inter-arrival time of voice packets is 20 ms.

The Audio source is a 2 Mbps-CBR traffic.

Finally the background best effort traffic is modeled as a multiplexed set of ON-OFF sources to model a network behavior that swings between activity and silence. ON and OFF periods are pareto-distributed and packet arrival distribution within an ON period is Poisson. This generator model results in a self-similar traffic with long-range dependence [20].

Real-time flows are described in table 2.

Table. 2. Flow Description

Stream ID	Source	Destination	Required Delay (ms)
Stream1	MPEG-1 Stream 1	Client 1	50
Stream2	MPEG-1 Stream 2	LAN Client	50
Stream3	MPEG-2 Stream	Client 2	50
Stream4	Voice Stream 1	Voice Stream 2	100
Stream5	CBR Stream	LAN Client	50

We evaluate the performance of WFQ and (m,k)-WFQ through three scenarios presented in table 3.

Table. 3. Simulation Scenarios

Scenario	Shared Link R1→R2	
1	WFQ	
2	(m,k)-WFQ	The Best-Effort flow is set to be "Optional"
3	(m,k)-WFQ	The Best-Effort flow is set to be "Mandatory"

We focus on the MPEG-1 Stream 1 delivery. The behavior of the other MPEG sources is similar to this stream. We obtain the two-segment curve parameters of this MPEG stream using a shaping device developed with OPNET. The parameters are presented in table 4.

Table. 4. MPEG Stream Specification

M	Maximum Packet Size	11.5 kb
p	Peak-Rate	4.2 Mbps
b	Maximum Burst Size	112 kb
r	Average Rate	1 Mbps

According to the GoP structure and frame sizes, the ratio of mandatory bits (of I and P Frames) in the GoP of MPEG-1 Stream 1 is $\lambda_M = 0.747$ and the ratio of optional ones (of B frames) is $\lambda_O = 0.253$. Optional packets whose deadlines exceed 50 ms are dropped. Hence, we derive in table 5 the analytic delay bounds using a bandwidth guarantee $R = 1\ Mbps$ by the application of equations 7 and 8.

Table. 5. Analytic Delay Bounds

	Delay
Scenario 1	112 ms
Scenarios 2 and 3	96 ms

The delay bound guaranteed by (m,k)-WFQ is lower than that provided by standard fair queueing. However, the analytic bounds are quite high to efficiently express the actual behavior of the system. We assume that the end-to-end delay required for those MPEG streams is 50 ms and we propose to simulate the system behavior with both scheduling strategies. A packet that comes over this required delay is considered to be missed or dropped.

Figure 5 depicts the loss ratio of each frame type of the MPEG stream. With Standard WFQ, the loss means that the packet comes later than its required deadline. However, with (m,k)-WFQ the loss may refer to a deadline miss or also to a dropped optional packet missing its deadline at the intermediate router.

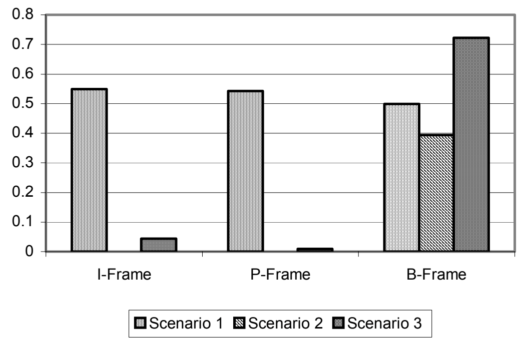


Fig. 5. Loss Ratio

It can be seen that (m,k)-WFQ algorithm has a better behavior than standard WFQ in overload condition to ensure graceful QoS degradation. We see that (m,k)-WFQ outperforms standard WFQ in terms of differentiation between MPEG packets. In fact, in the first scenario the frame miss ratio is quite similar for the three frame types and is around 54%.The loss of an I-frame or a P-frame has more sever impact than the loss of a B-frame according to the compression mechanism. Using packet classification enables the scheduler to make selective send/discard of real-time packets dynamically and depending on load conditions. In the second scenario (m,k)-WFQ has transmitted successfully all I-frames and P-frames. In the third scenario, only a reduced amount of important frames has been lost. Therefore, the selective frame discard by (m,k)-WFQ is very efficient to ensure an acceptable user's perceivable QoS when minimizing the loss of important frames and guaranteeing their delivery at their required playback delay.

Figures 6 and 7 show the instant delay and the average delay of the MPEG stream measured at the receiving application, respectively.

The (m,k)-WFQ algorithm guarantees lower delay for the MPEG stream than that performed by standard WFQ. In fact, (m,k)-WFQ takes profit of packet classification into mandatory and optional parts by skipping the transmission of optional packets missing their deadlines, which fosters the transmission of mandatory ones. We can observe in figure 5 that the instant delay of received packets in the second and the third scenarios does not exceed the 50 ms required deadline (some exceptions occur in the third scenario leading to few mandatory packets missing their deadlines), whereas in the first scenario, this delay reaches 80 ms. Hence, in the first scenario the client application must reserve more buffer space than it should reserve in the second and third scenarios to avoid sever degradation of playback quality. The second scenario gives lower average delay than that given by the third scenario. In fact, in the 2nd scenario mandatory frames are sent before best-effort flow regarded as optional.

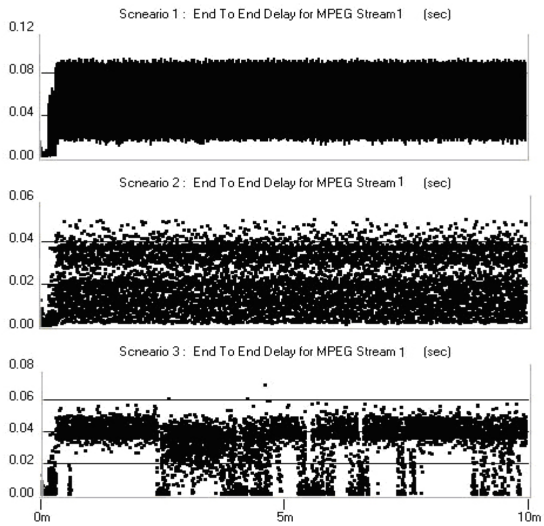


Fig. 6. Instant Delay of MPEG-1 Stream 1

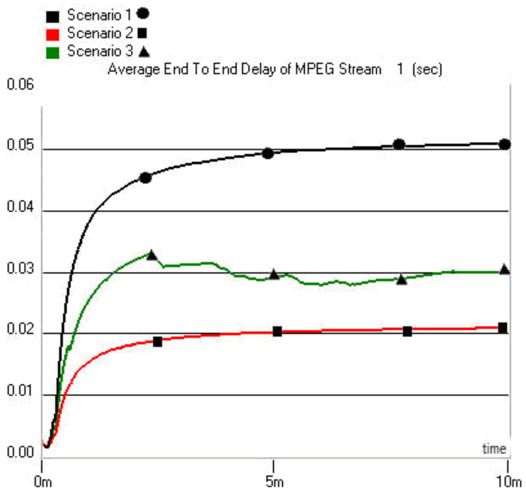


Fig. 7. Average Delay of MPEG-1 Stream 1

Moreover, the average delay bound depicted in figure 6, shows the avail of selective packet discard to have steadier and lower delay guarantee. Hence, we notice that for better delivery of VBR flows in a resource constrained guaranteed-rate network, using selective packet discard is an efficient technique to provide lower delay guarantee without increasing service share or buffering space.

With the same bandwidth reservation, the loss-tolerant QoS ensured by the (m,k)-WFQ algorithm makes better guarantees in terms of loss ratio and delay guarantee compared with standard fair queueing.

6 Conclusion

We have presented in this paper an extended technique of bandwidth reservation to make real-time guarantee with deterministic loss profile and efficient bandwidth resource management. We have proposed an extension of WFQ algorithm to consider (m,k)-firm constraints in the scheduling process. Analytic study and simulations show good performance of the proposed (m,k)-WFQ algorithm to provide lower delay guarantee for VBR flows. This mechanism provides the so called loss-tolerant deterministic QoS. In fact, in overload condition (m,k)-WFQ discards selectively optional packets to enhance transmission of mandatory ones to ensure a graceful QoS degradation. We are currently working to integrate (m,k)-firm guarantees into current DiffServ architecture to define loss-tolerant service classes using these timing constraints.

7 References

[1] A. Koubâa , Y.Q. Song "(m,k)-WFQ : Integrating (m,k)-firm Real-Time Constraints into Guaranteed Rate Networks" *Proceedings Real Time Systems Conference RTS'2004* Paris (France) 30 Mars-31 Mars 2004

[2] S. Wang, Y. Wang, K. Lin, "Integrating Priority with Share in the Priority-Based Weighted Fair Queueing Scheduler for Real-Time Networks" *Journal of RTS* pp. 119-149, Vol 22, 2002.

[3] Donglin Liu, Xiaobo Sharon Hu, Michael D. Lemmon, Qiang Ling, "Firm Real-Time System Scheduling Based on a Novel QoS Constraint" *Proceedings of the 24th IEEE International Real-Time Systems Symposium* (RTSS'03) pp. 386-395, Mexico, Dec. 2003

[4] M. Hamdaoui and P. Ramanathan. "A Dynamic Priority Assignment Technique for Streams with (m, k)-firm Deadlines". *IEEE Transactions on Computers*, Vol . 44 (4), pp. 1443–1451, Dec.1995.

[5] 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, pp 549-559, June 1999.

[6] J.Y. Le Boudec, P. Thiran, 'Network Calculus: A Theory of Deterministic Queueing Systems for the Internet' Springer Verlag, July 2002.

[7] A.T. Chronopoulos, C. Tang, E. Yaprak, " An Efficient ATM Network Swicth Scheduling", *IEEE Transactions on Broadcasting*, Vol. 49, N° 3, September 2003

[8] B. Furth "Handbook of Multimedia Computing" CRC Press 1999.

[9] J.M. Boyce, R.D. Gaglianello, "Packet Loss Effects on MPEG Video Sent Over the Public Internet" *Proceedings of the 6th ACM International Conference on Multimedia* September 1998.

[10] A.Demers, S.Keshavt, S.Shenker, "Analysis and Simulation of Fair Queuing Algorithm", *Proceedings ACM SigComm 89*, pp 3-12, 1989.

[11] S. J. Golestani, "A Self-Clocked Fair Queueing Scheme for Broadband Applications," *In Proceedings of IEEE INFOCOM 1994*, pp. 636-646.

[12] J. Bennet, H. Zhang "WF2Q: Worst-case Fair Weighted Fair Queueing", *Proc. Of IEEE Infocom '96*, March, 1996.

[13] P. Goyal, H.Vin, "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Service Packet Switching Networks" *IEEE/ACM Trans. On Networking*, Oct. 1997

[14] A. Striegel, G. Manimaran, "Dynamic Class-Based Queue Management for Scalable Media Servers", *Journal of Systems and Software*, vol.66, no.2, pp.119-128, May 2003.

[15] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE Journal of Selected Areas on Communications*., vol. SAC-4, pp. 833--846, Sept. 1986.

[16] OPNET Simulator http://www.opnet.com

[17] M. Krunz, H. Hughes "A Traffic Model for MPEG-Coded VBR Streams" *In Proc. ACM Sigmetrics'95* pp 47-55 May 1995

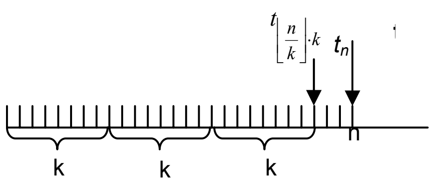
[18] O. Rose "Statistical properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems" Technical report No. 101 University of Wuerzburg, Germany 1995

[19] G. Ramamurthy, B. Sengupta "Modeling and Analysis of a Variable Bit Rate Video Multiplexer" *in Proc. Of IEEE INFOCOM'92* vol.2, pp. 817-827, 1992

[20] G. Kramer "On generating self-similar traffic using pseudo-Pareto distribution" A short Tutorial-Like, Network Research Lab, Department of Computer Science – University of California 2000

Appendix 1 - Proof of theorem 1

$\Rightarrow$ We denote by t_n the instant of n^{th} packet arrival.
$\forall n, \forall t_n \le t < t_{n+1}, \quad R(t) = n$.



- ***If n is multiple of k***: then it exists u such that $n = u \cdot k$. In this case, there are u consecutive and separate sets of k consecutive packets. Since only m packets out of any k packets exit the filtering device toward the network, then there is exactly $m \cdot u$ mandatory packets that leave the *(m,k)-filter*. So, $\left\lfloor \frac{R(t)}{k} \right\rfloor = u$ and

$$\Pi\left(R(t)\right) = \sum_{n=(u.k)+1}^{u.k} \overline{\kappa}(n\%k) = 0$$

and *Theorem 1* is satisfied since $\tilde{R}(t) = m.u$

- ***If n is not a multiple of k***: it exists $u \in \mathbb{N};\ \frac{n}{k} - 1 \le u \le \frac{n}{k}$ and $v \in \mathbb{N};\ 1 \le v \le k-1$ such that $n = u \cdot k + v$.

By definition, $u = \left\lfloor \frac{n}{k} \right\rfloor$ and $v = n - u.k$. At time $t_{u.k}$ there are $m \cdot u$ packets that leave the *(m,k)-filter*. From $t_{u.k}$ until t_n the filter sends only mandatory packets according to the defined κ-pattern κ. The n^{th} packet is mandatory if and only if $\overline{\kappa}\left((n\%k)+1\right) = 1$, in the time interval $\left[t_{u.k}, t_n\right]$, the number of mandatory packets is $\sum_{a=(u.k)+1}^{(u.k)+v} \kappa(n\%k)$ $\in [0, m]$. Then ,

$$\tilde{R}(t) = m \cdot u + \sum_{a=(u.k)+1}^{(u.k)+v} \overline{\kappa}(n\%k)$$

which satisfies *theorem 1*.

$\Leftarrow$ Evident

□

Appendix 2 – Proof of Theorem 2

Let us consider a constant-packet size stream with cumulative arrival function *R(t)* shaped with a leaky bucket controller with σ as maximum allowed burst and ρ as the average long-term rate. Then,

$$R(t) - R(s) \le \sigma + \rho\left(t - s\right) \ , \qquad \forall\ 0 \le s \le t$$

Also, we assume that this stream has (m,k)-firm constraints given by its κ-pattern κ. What kind of arrival curve would have the stream after crossing an (m,k)-firm filter, is the purpose of this paragraph *(see*

Fig5.1). The output $\tilde{R}(t)$ of the stream satisfies *Theorem 1*.So, $\forall\ 0 \le s \le t$

$$\tilde{R}(t)-\tilde{R}(s)=m\cdot\left\{\left\lfloor \frac{R(t)}{k}\right\rfloor-\left\lfloor \frac{R(s)}{k}\right\rfloor\right\}+\left\{\Pi\left(R(t)\right)-\Pi\left(R(s)\right)\right\}$$

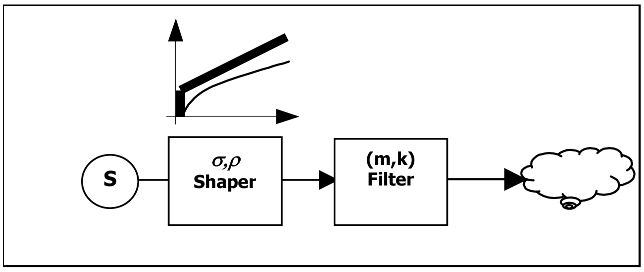


Fig A2.1 . System Description

We aim to determine an arrival curve at the output of the (m,k)-filter which is independent from any κ-pattern. However, the function $\pi(t-s)$ defined as

$$\pi(t-s)=\left\{\Pi\left(R(t)\right)-\Pi\left(R(s)\right)\right\}$$

is non-increasing and non-monotonic function and depends on κ-pattern setting of the stream. For this reason, we consider a new discrete sampling time $\mathrm{T}=\{t_0,t_k,t_{2k},\ldots,t_{nk},\ldots\}$ where t_{nk} represent the *(nk)*th packet arrival time. According to [6], this mapping results in some loss of information but it is sufficient to get an arrival curve for $\tilde{R}(t)$. With the new sampling time,

$$\forall\ s,t\ \in T,\ 0 \le s \le t;\ \ \pi(t-s)=0$$

and

$$\left\lfloor \frac{R(t)}{k}\right\rfloor=\frac{R(t)}{k};\ \ \forall t\in \mathrm{T}\ .$$

Then,

$$\forall s,t\in \mathrm{T},\ s\le t,\qquad \tilde{R}(t)-\tilde{R}(s)=m\cdot\left(\frac{R(t)}{k}-\frac{R(s)}{k}\right)$$

So, the output flow is constrained as follows:

$$\tilde{R}(t)-\tilde{R}(s)\le \frac{m}{k}\cdot\left(\sigma+\rho\cdot t\right)$$

This result leads to theorem 2.

□

Appendix 3

From Figure 2:

$$R^*(t)\sim\begin{pmatrix}\min\left(\lambda_M\left(M+p.t\right),\lambda_M\left(b+r.t\right)\right)+\\ \min\left(\lambda_O\left(M+p.t\right),\lambda_O\left(\sigma+r.t\right)\right)\end{pmatrix}\qquad \mathbf{A3.1}$$

Denote by

$x=\lambda_M\left(M+p.t\right)$, $y=\lambda_M\left(b+r.t\right)$, $z=\lambda_O\left(M+p.t\right)$, $t=\lambda_O\left(\sigma+r.t\right)$

so,

$$R^*(t)\sim\left(\min\left(x,b\right)y+\min\left(z,t\right)\right)=\min(x+z,x+t,y+z,y+t)$$

$\Rightarrow$

$$R^*(t)\sim\min\begin{pmatrix}\left(M+p.t\right),\left(\lambda_M M+\lambda_O.\sigma\right)+\left(\lambda_M p+\lambda_O r\right).t,\\ \left(\lambda_O M+\lambda_M b\right)+\left(\lambda_O p+\lambda_M r\right).t,\\ \left(\lambda_M b+\lambda_O\sigma\right)+r.t\end{pmatrix}\qquad \mathbf{A3.2}$$

Suppose $\theta_\sigma=\frac{\sigma-M}{p-r}$, $\theta_b=\frac{b-M}{p-r}$, $\gamma_1=\lambda_M b+\lambda_O\sigma$ and $\gamma_2=\lambda_M M+\lambda_O.\sigma$ and $\rho=\lambda_M p+\lambda_O r$.

The curve of $R^*(t)$ has the form $\min(a,b,c,d)$ with $a=x+z$, $b=x+t$, $c=y+z$ and $d=y+t$

From Equation A3.1

- $a-b=\lambda_O\left[\left(M-\sigma\right)+\left(p-r\right).t\right]$
  so $a\le b$ iff $t\le\theta_\sigma$
- $c-d=\lambda_O\left[\left(M-\sigma\right)+\left(p-r\right).t\right]$
  so $c\le d$ iff $t\le\theta_\sigma$
- $a-c=\lambda_M\left[\left(M-b\right)+\left(p-r\right).t\right]$
  so $a\le c$ iff $t\le\theta_b$
- $b-d=\lambda_M\left[\left(M-b\right)+\left(p-r\right).t\right]$
  so $b\le d$ iff $t\le\theta_b$

for $t\le\theta_\sigma\le\theta_b$ we have $a\le b$ and $a\le c$ $\Rightarrow\min(a,b,c,d)=\min(a,d)$

for $t\ge\theta_b\ge\theta_\sigma$ we have $d\le c$ and $d\le b$ $\Rightarrow\min(a,b,c,d)=\min(a,d)$

for $\theta_\sigma\le t\le\theta_b$ we have $a\le b$, $d\le c$, $a\le c$ and $b\le d$ then $\min(a,b,c,d)=b$

so, $\Rightarrow\min(a,b,c,d)=\min(a,b,d)\quad\forall t\in\mathbb{R}^+$

Finally, the arrival curve is

$$R^*(t)\sim\min\begin{pmatrix}M+p.t,\\ \left(\lambda_M M+\lambda_O.\sigma\right)+\left(\lambda_M p+\lambda_O r\right).t,\\ \left(\lambda_M b+\lambda_O\sigma\right)+r.t\end{pmatrix}$$