

Integrating (m,k)-Firm Real-Time Guarantees into the Internet QoS Model

Anis KOUBAA, Ye-Qiong SONG, Jean-Pierre THOMESSE

LORIA – INPL – UHP Nancy I
615 rue du jardin botanique, 54602 Villers-Les-Nancy

{akoubaa, song, thomesse}@loria.fr

Abstract. Having in mind that real-time streams tolerate some deadline misses according to (m,k)-firm constraints [1], this paper presents a solution that consists in integrating (m,k)-firm temporal requirements into the *guaranteed-rate* quality of service such as IntServ Model to provide both bandwidth and delay guarantees as required by real-time applications. Analytic study using Network Calculus gives the upper bound on delay guaranteed by the proposed approach and shows better behavior compared to the classical IntServ QoS model.

1 Introduction

Guaranteed QoS of IntServ model uses guaranteed-rate servers, such as Weighted Fair Queueing (WFQ) and its variants, to provide a share-driven scheduling policy to serve active streams according to their bandwidth requirements specified by the traffic parameter known as TSPEC, which defines an arrival curve for the real-time flow. Moreover, guaranteed end-to-end delay bound is also computed from the TSPEC parameter as well as the reserved bandwidth. This delay bound is affected by the bursty nature of the flow. In fact, for a given reserved bandwidth, when the burst size is large, the delay experienced by real-time packets gets higher and may exceed the required deadline. However, increasing service share to meet the required deadline leads to inefficient resource utilization since the reserved bandwidth would be higher than the actual need of the application. A solution proposed in [2], called PWFQ, consists in integrating static priority within WFQ scheduling algorithm in order to better manage the delay bounds for various sessions. The idea consists in serving with *Static Priority* policy, packets whose virtual finish tags [13] belong to a specified *sliding window*. This technique decouples the delay from service share and provides lower delay for low-share streams without degrading the delay of other sessions. The main critical point of this technique is to find the optimal window size and the priority assignment for the stream set.

Furthermore, congestion is an additional problem for streams with large burst size. In fact, when serving several concurrent streams, the guaranteed-rate router may suffer from congestion if its internal queue reaches its capacity. The basic solution to prevent such situation consists in performing stochastic dropping using RED

mechanism; however, it could affect the QoS of real-time streams if packet drops are made inadequately. An elegant share-driven scheduling technique with drop was recently proposed by Koubâa and Song in [3], called (m,k)-WFQ, and resolves the problem of delay guarantees for low-share bursty streams as well as dropping process. This technique integrates (m,k)-firm temporal requirement [1] into WFQ scheduling algorithm and takes advantage of some deadline miss tolerance for real-time streams with the respect of (m,k)-firm timing constraints, which consists in meeting the deadlines of at least m messages among any k consecutive message. (m,k)-WFQ is described in section 2.

In our previous work [3], we have only defined the (m,k)-WFQ algorithm and evaluate its performance using a basic simulation scenario. In this paper we propose a novel analytic study using Network Calculus to compute the delay bound for a given flow described with its TSPEC parameter and its (m,k)-firm requirement in the IntServ model. For this purpose, we introduce the (m,k)-filtering concept to integrate (m,k)-firm constraint into Network Calculus framework and derive the expression of the delay upper bound.

The remainder of this paper is structured as follows. Section 2 outlines the basic steps of (m,k)-WFQ scheduling algorithm. Section 3 presents mathematical background to adapt (m,k)-firm constraints to Network Calculus framework. In section 4, we propose to integrate (m,k)-firm constraints into IntServ QoS model using (m,k)-WFQ to provide bandwidth guarantee. Section 5 concludes this paper.

2 (m,k)-WFQ Scheduling Algorithm

In this section we briefly describe the (m,k)-WFQ scheduling algorithm. We refer the reader to our technical report [4] for more details.

(m,k)-WFQ is intended to fairly serve flows according to their bandwidth requirement and with respect to their (m,k)-firm constraints. A stream is said to have (m,k)-firm constraint if at least m packets inside any window of k consecutive packets must meet their required deadlines, otherwise the temporal QoS of the stream would be transgressed. The idea behind (m,k)-WFQ is to classify packets of incoming real-time flows into two parts - *Mandatory* and *Optional* - according to their (m,k)-firm constraints. This classification is made using the κ -pattern. 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. $\kappa(i)$ denotes the i^{th} element of the κ -pattern for $1 \leq i \leq k$. The n^{th} packet of a stream is classified as mandatory when $\kappa(n \% k) = 'M'$ for $n = 1, 2, \dots$ where $\%$ is the modulus the operator.

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:

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

where F_i^k is the virtual finish time of the k^{th} packet of i^{th} stream. $V(t)$ is the virtual time when the 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 constraint 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*, then the (m,k)-WFQ server sends it immediately. Else, if the selected packet is *optional*, the scheduler checks whether its deadline would be missed after being served. If it is the case, (m,k)-WFQ drops the selected optional packet, else, if the deadline would be met, the selected optional packet is transmitted.

3 The (m,k)-Filtering Concept

We introduce the (m,k)-filtering concept to adapt (m,k)-firm constraints to Network Calculus framework, and then derive the upper bound on delay in section 4. Further details are presented in [4].

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

The following theorem gives the arrival curve of a (σ, ρ) -shaped stream that crosses an (m,k)-filter. The proof is shown in [4].

Theorem 1. 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 $\tilde{\alpha}(t) = \tilde{\sigma} + \tilde{\rho} t$ where:

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

and $t \in T = \{t_0, t_k, t_{2k}, \dots, t_{nk}, \dots\}$ where t_{nk} is $(nk)^{\text{th}}$ packet arrival time.

This curve is the **minimal arrival curve** of the stream.

4 Integrating (m,k)-Firm Constraints into IntServ QoS Model

In the IntServ QoS model, variable-bit rate flows are typically described by the quadruple TSPEC = (M, p, b, r) where M is the maximum packet size of the flow, p is the peak rate, b is the maximum burst size and r is the average long-term rate.

The cumulative arrival function of the VBR traffic is then upper bounded by the arrival curve $\alpha(t) = \min(M + p.t, b + r.t)$. In this QoS model, routers use WFQ and its variants to provide bandwidth guarantee.

By making bandwidth reservation R and a maximum latency T , the maximum delay guaranteed for a flow using WFQ scheduling [5]:

$$D_{\max} = \frac{M}{R} + \frac{b-M}{R} \left(\frac{p-R}{p-r} \right)^+ + T \quad (2)$$

This delay bound depends only on the traffic specification (M, p, b, r) and the reserved bandwidth. No temporal constraint is considered. As consequent, if the burst size b is large, the delay may exceed the required deadline of the real-time stream.

Using (m,k)-WFQ scheduler, the actual traffic transmitted would be slightly different from the TSPEC form since the scheduler would drop optional packets missing their deadlines. Hence, we need to estimate the arrival curve of the *effective flow* transmitted by the scheduler which includes all mandatory packets (bits) and the maximum number of optional packets (bits) transmitted by the scheduler. We denote by D_{req} the required delay for each packet of the flow. Then, the burst size of optional packets cannot be larger than $\sigma = D_{req} \cdot r$ since r , the average long-term rate, is the minimum reserved bandwidth. We denote λ_M (resp. λ_O) the ratio of mandatory (resp. optional) bits into the window of k consecutive packets according to its κ -pattern. Figure 1 presents the effective flow model.

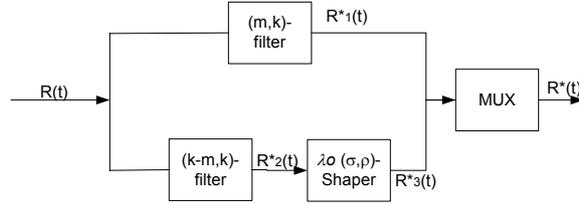


Fig. 1. The effective flow model is obtained using the (m,k)-filter.

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. Using *Theorem 1*, we show that the curve of the traffic actually served by (m,k)-WFQ, called *smoothed curve* is (c.f figure 2):

$$\alpha^*(t) = \min(M + p.t, (\lambda_M M + \lambda_O \sigma) + \rho t, (\lambda_M b + \lambda_O \sigma) + r t) \quad (3)$$

Where $\rho = \lambda_M p + \lambda_O r$.

We denote by $\theta_\sigma = \frac{\sigma - M}{p - r}$, $\theta_b = \frac{b - M}{p - r}$, $\gamma_1 = \lambda_M b + \lambda_0 \sigma$, $\gamma_2 = \lambda_M M + \lambda_0 \sigma$.

From the figure 2, the smoothed curve is made of three segments as expressed by equation 3. When time interval $[\theta_\sigma, \theta_b]$ is very short, a good approximation of this arrival curve is:

$$R^*(t) \sim \min(M + pt, (\lambda_M b + \lambda_0 \sigma) + rt) \quad (4)$$

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

If we consider the approximation of equation 4, a direct result of the delay bound, when replacing the TSPEC curve by the approximated smoothed curve, is:

$$D_{\max} = \frac{M}{R} + \frac{(\lambda_M b + \lambda_0 \sigma) - M}{R} \left(\frac{p - R}{p - r} \right)^+ + T \quad (5)$$

A finer bound is obtained by considering the three segment curve of equation 3. We show that the delay bound guaranteed by (m,k)-WFQ for a reserved bandwidth R and a latency T (Figure 2):

$$D_{\max} = \max \left[\frac{M}{R} + \left(\frac{\sigma - M}{R} \right) \left(\frac{p - R}{p - r} \right)^+, \left(\frac{\lambda_M M + \lambda_0 \sigma}{R} + \left(\frac{b - M}{R} \right) \left(\frac{p - R}{p - r} \right) \right) \right] + T \quad (6)$$

All the details and proofs leading to this result could be found in [4].

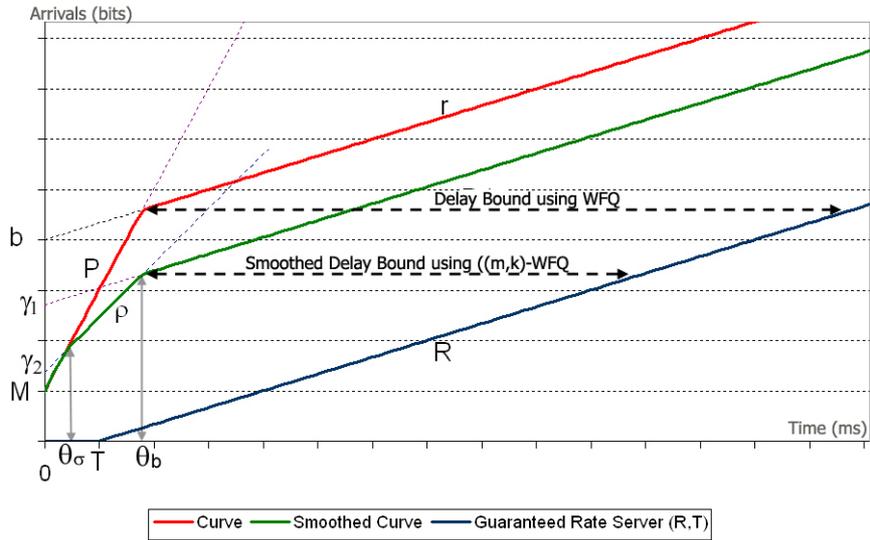


Fig. 2. This figure shows the delay bounds for a flow served by WFQ and (m,k)-WFQ with a bandwidth reservation R and a maximum latency T. The smoothed curve denotes the curve of the effective flow. The delay bound guaranteed by (m,k)-WFQ is lower than that guaranteed by WFQ due to the skipping of optional packets missing their deadlines.

The delay bound guaranteed by (m,k)-WFQ is always lower than that guaranteed by WFQ. Moreover, we can adjust the deadline to drop optional packets in order to make D_{max} equal to the required delay D_{req} . Hence to have $D_{max} = D_{req}$, according to equation 6, the maximum optional burst-size eligible for serving is:

$$\sigma = \min \left\{ \frac{R \cdot (D_{req} - T) - M \left(1 - \frac{(p-R)^+}{(p-r)} \right)}{\frac{(p-R)^+}{(p-r)}}, \frac{R \cdot (D_{req} - T) - \lambda_M M + (b-M) \left(\frac{p-R}{p-r} \right)}{\lambda_O} \right\} \quad (7)$$

Consequently, the maximum delay to serve an optional packet is not more than $D_{op} = \sigma / r$. Therefore, integrating (m,k)-firm timing constraints provides an important flexibility for real-time applications and makes guarantees on both bandwidth and delay. In fact, based on the TSPEC parameter and the (m,k)-firm requirement, an intermediate router makes the bandwidth reservation and adjusts, by using equation 7, the maximum allowed deadline to send an optional packet in order to guarantee the required delay for mandatory packets. Moreover, simulation study made in [4] shows the practical interest of using (m,k)-WFQ for guaranteeing loss-tolerant QoS for MPEG streams.

5 Conclusion

In this paper, we have presented a new approach of QoS that consists in integrating (m,k)-firm real-time guarantees into the Internet QoS model to provide delay guarantee for real-time applications that tolerate some deadline misses. Analytic study shows that (m,k)-WFQ provides lower delay guarantees than that provided by WFQ according to (m,k)-firm constraints.

We are currently working towards integrating (m,k)-firm guarantee into DiffServ QoS architecture to define loss-tolerant service classes using these timing constraints.

References

- [1] M. Hamdaoui and P. Ramanathan. "A Dynamic Priority Assignment Technique for Streams with (m, k)-firm Deadlines". *IEEE Trans. on Computers*, 44 (4), 1443–1451, Dec.1995.
- [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] A. Koubaa , Y.Q. Song "Amélioration des Délais dans les Réseaux à Débits Garantis pour des Flux Temps-Réel Sous Contrainte (m,k)-Firm" SETITE'03 Mars 2003 Sousse, Tunisie.
- [4] A. Koubaa, Y.Q. Song "Integrating (m,k)-Firm Real-Time Guarantees in the Internet QoS Model" INRIA-LORIA Technical Report February 2004.
- [5] J.Y. Le Boudec, P. Thiran, *Network Calculus: A Theory of Deterministic Queueing Systems for the Internet* Springer Verlag, July 2002.