Deterministic and probabilistic QoS guarantees for the EF class in a DiffServ/MPLS domain

Leila Azouz Saidane, Pascale Minet, Steven Martin, Ines Korbi

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Deterministic and probabilistic QoS guarantees for the EF class in a DiffServ/MPLS domain

Leila Azouz Saidane — Pascale Minet — Steven Martin — Ines Korbi

N° 5622

July 2005

Thème COM

ISSN 0249-6399 ISRN INRIA/RR--5622--FR+ENG

Deterministic and probabilistic QoS guarantees for the EF class in a DiffServ/MPLS domain

Leila Azouz Saidane* , Pascale Minet† , Steven Martin‡ , Ines Korbi*

Thème COM — Systèmes communicants
Projets Hipercom

Rapport de recherche n° 5622 — July 2005 — 24 pages

Abstract: In the Differentiated Services (DiffServ) architecture, the Expedited Forwarding EF class has been proposed for applications requiring low end-to-end packet delay, low delay jitter and low packet loss (e.g. voice and video applications that are delay and jitter sensitive). In this paper, we focus on the quantitative Quality of Service (QoS) guarantee that can be granted to an EF flow in terms of end-to-end delay. Two approaches are presented. The deterministic one is based on a worst case analysis and leads to a deterministic bound which is infrequently reached. The probabilistic approach, based on the probability density function of the response time, is introduced to evaluate the probability of missing a given deadline. This study shows that delays much smaller than the deterministic bound can be guaranteed with probabilities close to one. An admission control derived from these results is then proposed, providing a probabilistic QoS guarantee to EF flows.

Key-words: Quality of service, probabilistic guarantee, deterministic guarantee, end-to-end response time, M/G/1 station, MPLS/DiffServ.

* ENSI, CRISTAL, Campus Universitaire Manouba, 2010, Tunis, Tunisie, *leila.saidane@ensi.rnu.tn*, *ines.korbi@softhome.net*
† INRIA Rocquencourt, 78153 Le Chesnay Cedex, France, *pascale.minet@inria.fr*
‡ Université Paris Sud, LRI, 91405 Orsay, France, *steven.martin@lri.fr*

Garanties de qualité de service déterministes et probabilistes pour la classe EF dans un domaine DiffServ/MPLS

Résumé : Dans une architecture de services différenciés (DiffServ), la classe la plus prioritaire, la classe EF (Expedited Forwarding), a été proposée pour les applications ayant des contraintes sur les délais de bout-en-bout, la gigue et le taux de perte de paquets (par exemple, les applications voix et vidéo qui sont sensibles au délai et à la gigue). Dans ce papier, nous nous intéressons à la garantie quantitative de qualité de service qui peut être fournie à un flux EF en terme de délai de bout-en-bout. Deux approches sont présentées. L'approche déterministe est basée sur une analyse pire cas et conduit à une borne déterministe. Seuls quelques paquets atteignent cette borne pire cas. L'approche probabiliste, basée sur une fonction de densité de probabilité du temps de réponse, est introduite pour évaluer la probabilité de dépasser une échéance donnée. Cette étude montre que des délais beaucoup plus petits que la borne déterministe peuvent être garantis avec des probabilités proches de 1. Un contrôle d'admission dérivé de ces résultats est alors proposé, fournissant une garantie probabiliste de qualité de service aux flux EF.

Mots-clés : Qualité de service, garantie probabiliste, garantie déterministe, temps de réponse de bout-en-bout, station M/G/1, MPLS/DiffServ.

Contents

1 Introduction

The classical Internet best-effort service model cannot provide Quality of Service (QoS) guarantees to new applications, particularly real-time applications (including voice and/or video). The low scalability of the *Integrated Services* (IntServ) architecture led the IETF to define the *Differentiated Services* (DiffServ) architecture [6]. This architecture is based on traffic aggregation in a limited number of classes. In particular, the *Expedited Forwarding* (EF) class has been proposed for applications requiring low end-to-end packet delay, low delay jitter and low packet loss (e.g. voice and video applications that are delay and jitter sensitive), the EF class being the highest priority class.

In addition, *MultiProtocol Label Switching* (MPLS) has been introduced to improve routing performance and then to obtain shorter packet response time. This technology is strategically significant for traffic engineering [3], especially when it is used with constraint-based routing. As MPLS allows to fix the path followed by all packets of a flow, it makes easier the QoS guarantee.

In this paper, we show how to provide a quantitative QoS guarantee to all the EF flows in a DiffServ/MPLS domain. More precisely, we focus on the end-to-end response time of such flows. Indeed, beyond the qualitative definition of the offered QoS, no end-to-end quantitative guarantees have been proved for the EF class [5]. Two approaches can be used: probabilistic and deterministic ones. In this paper, a mathematical model has been developed to obtain the probability of meeting the required quality of service. In fact, the model focuses, among others, on the density probability function of an EF packet response time. This yields to obtain the probability that the EF packet response time is less than a required delay. Moreover, an admission control of EF flows can be derived from these results.

The rest of this paper is organized as follows. In section 2, we define the problem and the adopted assumptions. In section 3, we show how to compute deterministic bounds on the node response time and then on the end-to-end response time for any EF flow, based on a worst case analysis. By definition, the worst case end-to-end response time of a flow is equal to the worst case end-to-end response time experimented by a packet of this flow. Section 4 presents the mathematical models providing the probability density function of the end-to-end response time for an EF packet. In section 5, we show how to provide a probabilistic QoS guarantee to EF flows, by evaluating the probability of meeting the requested QoS. An admission control, based on these results, is then proposed for the EF flows. Finally, we conclude the paper in section 6.

2 The problem

We investigate the problem of providing quantitative QoS guarantees to any EF flow, in terms of end-to-end response time and delay jitter which are defined between the ingress

node and the egress node of the flow in the considered DiffServ domain. As we make no particular assumption concerning the arrival times of packets in the domain, the feasibility of a set of EF flows is equivalent to meet constraints on the end-to-end response time and jitter, whatever the arrival times of the packets in the domain.We now detail our assumptions.

2.1 DiffServ Architecture

In the DiffServ architecture [6], traffic is distributed over a small number of classes. Packets carry the code of their class. This code is then used in each DiffServ-compliant router to select predefined packet handling functions (in terms of queuing, scheduling and buffer acceptance), called *Per-Hop Behavior* (PHB). Nodes at the boundary of the network (ingress and egress routers), perform complex treatments (packet classification and traffic conditioning) whereas nodes in the core network (core routers), forward packets according to their class code. Several per-hop behaviors have been defined:

- the *Best-Effort Forwarding* PHB is the default one.
- the *Assured Forwarding* (AF) PHB group [15]. Four classes, providing more or less resources in terms of bandwidth and buffers, are defined in the AF service.
- the *Expedited Forwarding* (EF) PHB [16]. Traffic belonging to the EF service is delivered with very low latency and drop probability, up to a negotiated rate. This service can be used for instance by IP telephony.

We consider that a DiffServ-compliant router implements Best-Effort, AF and EF classes. When a packet enters the node scheduler, it is scheduled with the other packets of its class waiting for processing. As illustrated by figure 1, the EF class is scheduled with a *Fixed Priority Queuing* scheme with regard to the other classes. Thus, the EF class is served as long as it is not empty. Packets in the Best-Effort and AF classes are served according to *Weighted Fair Queuing* (WFQ) [20]. In this way, EF traffic will obtain low delay thanks to *Fixed Priority Queuing* scheduler and AF traffic will receive a higher bandwidth fraction than Best-Effort thanks to WFQ. Notice that resources provisioned for the EF class that are not used are available for the other classes.

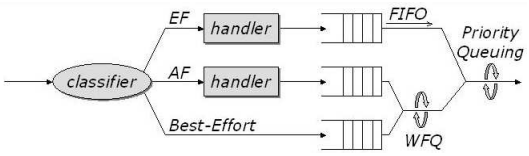

Figure 1: DiffServ-compliant router

2.2 Assumptions

• **Scheduling assumption**

Assumption 1 *On each node, the* EF *class scheduling is* FIFO *and packet scheduling is non-preemptive.*

Therefore, the node scheduler waits for the completion of the current packet transmission (if any) before selecting the next packet.

• **Network assumptions**

Assumption 2 *All routers are DiffServ-compliant and Label Switching Routers (*LSR*).*

Assumption 3 *Links interconnecting routers are supposed to be* FIFO.

Assumption 4 *The network delay between two nodes has known lower and upper bounds:* P_{min} *and* P_{max}.

Assumption 5 *Network is reliable: neither network failures nor packet losses.*

• **Traffic assumption**

Assumption 6 *We focus on a set* $\tau = \{\tau_1, ..., \tau_n\}$ *of* n EF *flows. According to* MPLS, *each flow* τ_i *follows a fixed sequence of nodes, called path.*

3 Deterministic Approach

In this worst case analysis, we compute the worst case end-to-end response time of sporadic flows. We assume that time is discrete. [4] shows that results obtained with a discrete scheduling are as general as those obtained with a continuous scheduling when all flow parameters are multiples of the node clock tick. In such conditions, any set of flows is feasible with a discrete scheduling if and only if it is feasible with a continuous scheduling.

The assumptions used in this section are those given in Section 2 plus additionnal ones specific to the deterministic approach. We first introduce the notations and definitions used throughout the paper and define the constituent parts of the end-to-end response time of any EF flow. Then, we show how to compute upper bounds on the end-to-end response time of any flow belonging to the EF class, based on a worst case analysis of FIFO scheduling.

3.1 Notations and definitions

In this section,we focus on the end-to-end response time of any packet m of any EF flow $\tau_i \in \{\tau_1, ..., \tau_n\}$ following a path $\mathcal{P}_i$ comprising $|\mathcal{P}_i|$ nodes. Moreover, we use the following notations and definitions:

Assumption 7 *The EF flows are assumed to be sporadic. A sporadic flow τ_i is defined by:*

- *T_i, the minimum interarrival time (abusively called period) between two successive packets of τ_i;*
- *C_i^h, the maximum processing time on node h of a packet of τ_i;*
- *J_i, the maximum jitter of packets of τ_i arriving in the DiffServ domain;*
- *D_i, the end-to-end delivery deadline: any packet of flow τ_i generated at time t must be delivered at the latest at time $t + D_i$.*

This characterization is well adapted to real-time flows (e.g. process control, voice and video, sensor and actuator). For any packet g belonging to the EF class, we denote $\tau(g)$ the index number of the EF flow which g belongs to. Moreover, B^h denotes the maximum processing time at node h of any packet of flow not belonging to the EF class.

a_m^h	the arrival time of packet m in node h;
R_{max_i}	the worst case end-to-end response time of flow τ_i;
$Smin_i^h$	the minimum time taken by a packet of flow τ_i to arrive on node h;
$Smax_i^h$	the maximum time taken by a packet of flow τ_i to arrive on node h;
J_i^h	the worst case jitter of flow τ_i when entering node h;
$first_{j,i}$	the first node visited by the EF flow τ_j on path $\mathcal{P}_i$;
$slow_i$	the slowest node visited by EF flow τ_i on its path, that is for any node $h \in \mathcal{P}_i$, $C_i^h \leq C_i^{slow_i}$;
$slow_{j,i}$	the slowest node visited by the EF flow τ_j on path $\mathcal{P}_i$, that is for any node $h \in \mathcal{P}_j \cap \mathcal{P}_i$, $C_j^h \leq C_j^{slow_{j,i}}$.

Definition 1 *An idle time t is a time such that all packets arrived before t have been processed at time t.*

Definition 2 *A busy period is defined by an interval $[t, t')$ such that t and t' are both idle times and there is no idle time $\in (t, t')$.*

Definition 3 *For any node h, the processor utilization factor for the EF class is denoted U_{EF}^h. It is the fraction of processor time spent by node h in processing EF packets. It is equal to $\sum_{i=1}^{n}(C_i^h/T_i)$.*

3.2 Determination of the end-to-end response time

In this section, we show how to compute the worst case end-to-end delay for EF flows visiting the DiffServ domain. Let us consider any EF flow τ_i, $i \in [1,n]$, following a path $\mathcal{P}_i$ comprising $|\mathcal{P}_i|$ nodes. The end-to-end response time of any packet m of τ_i generated at time t depends on:

- the delay due to the non-preemptive effect, denoted $\delta_i(t)$ (see subsection 3.2.2);
- the delay due to other EF packets, denoted $X_{EF}(t)$ (see subsection 3.2.1);
- network delays, bounded by $(|\mathcal{P}_i| - 1) \cdot P_{max}$.

As we consider a non-preemptive scheduling, the processing of a packet can no longer be delayed after it has started. That is why we compute the latest starting time of packet m on its last node visited. To achieve that, we adopt the trajectory approach. This approach consists in moving backwards through the sequence of nodes m visits, each time identifying preceding packets and busy periods that ultimately affect the delay of m.

- *For the sake of simplicity, we assume that $|\mathcal{P}_i| = q$ and nodes visited by flow τ_i are numbered from 1 to q.*

Moreover,we assume in a first step that any EF flow τ_j following path $\mathcal{P}_j$ with $\mathcal{P}_j \neq \mathcal{P}_i$ and $\mathcal{P}_j \bigcap \mathcal{P}_i \neq \emptyset$ never visits a node of path $\mathcal{P}_i$ after having left path $\mathcal{P}_i$ (cf. Assumption 8). In subsection 3.4, we show how to remove this assumption.

Assumption 8 *For any EF flow τ_j following path $\mathcal{P}_j$ with $\mathcal{P}_j \neq \mathcal{P}_i$ and $\mathcal{P}_j \bigcap \mathcal{P}_i \neq \emptyset$, if there exists a node $h \in \mathcal{P}_j \bigcap \mathcal{P}_i$ such that τ_j visits $h' \neq h+1$ immediately after h, then τ_j never visits a node $h'' \in \mathcal{P}_i$ after.*

According to the trajectory approach, we focus on the busy period bp^q in which m is processed on node q and we define $f(q)$ as the first EF packet processed in bp^q. But $f(q)$ does not necessarily come from node $q-1$ since the EF flow that $f(q)$ belongs to may follow a path different from $\mathcal{P}_i$. Hence, to move backwards through the sequence of nodes m traverses, each time identifying preceding packets and busy periods that ultimately affect the delay of m, we have to consider an additional packet on node q, that is $p(q-1)$: the first EF packet processed between $f(q)$ and m on node q and coming from node $q-1$. We denote bp^{q-1} the busy period in which $p(q-1)$ has been processed on node $q-1$ and $f(q-1)$ the first EF packet processed in bp^{q-1}. We then define $p(q-2)$ as the first EF packet processed between $f(q-1)$ and m on node $q-1$ and coming from node $q-2$. And so on until the busy period of node 1 in which the packet $f(1)$ is processed. We have thus determined the busy periods on nodes visited by τ_i that can be used to compute the end-to-end response time of packet m (see Figure 2).

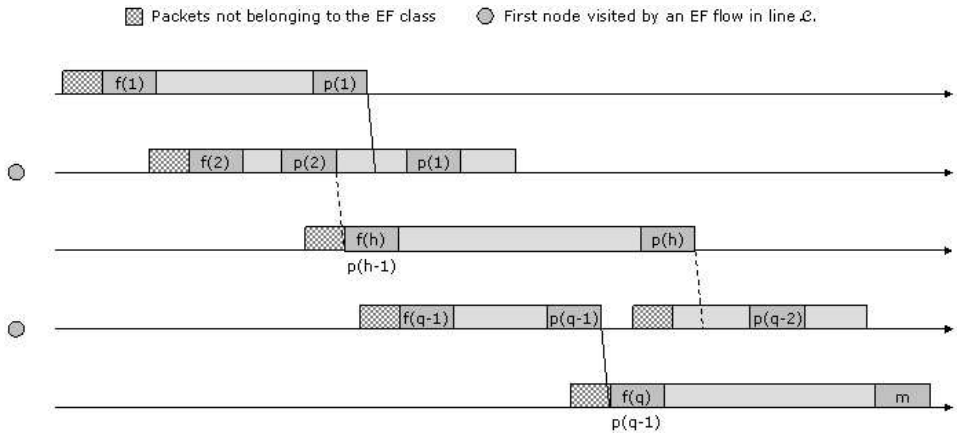

Figure 2: Response time of packet m

The latest departure time of packet m from node q is then equal to:

$a^1_{f(1)}$ $+(q-1)\cdot P_{max} + \delta_i(t)$

+ the processing time on node 1 of packets $f(1)$ to $p(1)$

+ the processing time on node 2 of packets $f(2)$ to $p(2) - (a^2_{p(1)} - a^2_{f(2)})$

...

+ the processing time on node q of packets $f(q)$ to $m - (a^q_{p(q-1)} - a^q_{f(q)})$.

Hence, the end-to-end response time of packet m is bounded by:

$$a^1_{f(1)} - t + \sum_{h=1}^{q} \left(\sum_{g=f(h)}^{p(h)} C^h_{\tau(g)} \right) + (q-1)\cdot Pmax + \delta_i(t) - \sum_{h=2}^{q} \left(a^h_{p(h-1)} - a^h_{f(h)}\right).$$

We recall that $first_{j,i}$ denotes the first node visited by the EF flow τ_j on path $\mathcal{P}_i$. We can notice that on any node h of path $\mathcal{P}_i$, if there exists no EF flow τ_j such that $h = first_{j,i}$, then $p(h-1) = f(h)$ and so $a^h_{p(h-1)} - a^h_{f(h)} = 0$. In other words, if $p(h-1) \neq f(h)$, then there exists an EF flow τ_j such that $h = first_{j,i}$. In such a case, by definition of $p(h)$, all the packets in $[f(h), p(h-1))$ cross path $\mathcal{P}_i$ for the first time at node h. We can then act on their arrival times. Postponing the arrivals of these messages in the busy period where $p(h-1)$ is processed, would increase the departure time of m from node q. Hence, in the worst case, $p(h-1) \in bp^h$ and $a^h_{p(h-1)} = a^h_{f(h)}$.

By numbering consecutively on any node h the EF packets processed after $f(h)$ and before $p(h)$ (with $p(q) = m$), we get an upper bound on the end-to-end response time of packet m, that is:

$$a^1_{f(1)} - t + \sum_{h=1}^{q} \left(\sum_{g=f(h)}^{p(h)} C^h_{\tau(g)} \right) + \delta_i(t) + (q-1)\cdot P_{max}.$$

3.2.1 Evaluation of the maximum delay due to EF packets

We now consider the term $X_{EF} = a^1_{f(1)} - t + \sum_{h=1}^{q} \left(\sum_{g=f(h)}^{p(h)} C^h_{\tau(g)} \right)$, that is the maximum delay due to EF packets and incurred by m. By definition, for any node $h \in [1, slow_i)$, $p(h)$ is the first packet belonging to the EF class, processed in bp^{h+1} and coming from node h. Moreover, $p(h)$ is the last packet considered in bp^h. Hence, if we count packets processed in bp^h and bp^{h+1}, only $p(h)$ is counted twice. In the same way, for any node $h \in (slow_i, q]$, $p(h-1)$ is the first packet belonging to the EF class, processed in bp^h and coming from node $h-1$. Moreover, $p(h-1)$ is the last EF packet considered in bp^{h-1}. Thus, $p(h-1)$ is the only packet counted twice when counting packets processed in bp^{h-1} and bp^h.

In addition, for any node $h \in [1, q]$, for any EF packet g visiting h, the processing time of g on node h is less than $C^{slow_{\tau(g),i}}_{\tau(g)}$, where $slow_{j,i}$ is the slowest node visited by τ_j on path $\mathcal{P}_i$. Hence, if on any node $h \in [1, slow_i)$ (resp. $h \in (slow_i, q]$), we bound $p(h)$ (resp. $p(h-1)$) by $C^h_{max} = \max_{j \in \tau}(C^h_j)$, then we get:

$$X_{EF} \leq a^1_{f(1)} - t + \sum_{g=f(1)}^{m} C^{slow_{\tau(g),i}}_{\tau(g)} + \sum_{\substack{h=1 \\ h \neq slow_i}}^{q} C^h_{max}.$$

We now evaluate the quantity $\sum_{g=f(1)}^{m} C^{slow_{\tau(g),i}}_{\tau(g)}$. This quantity is bounded by the maximum workload generated by the EF flows visiting at least one node of path $\mathcal{P}_i$.

On node h, a packet of any EF flow τ_j visiting at least one node visited by τ_i can delay the execution of m if it arrives on node $first_{j,i}$, $first_{j,i}$ denoting the first node of $\mathcal{P}_i$ common with $\mathcal{P}_j$, at the earliest at time $a^{first_{j,i}}_{f(first_{j,i})}$ and at the latest at time $a^{first_{j,i}}_m$. Indeed, if a packet of τ_j arrives on node $first_{j,i}$ after $a^{first_{j,i}}_m$, it will be processed on this node after m due to the FIFO-based scheduling of the EF class. So, if the next node visited by m is also the next node visited by the packet of τ_j, this packet will arrive after m on this node, and so on. Therefore, it will not delay packet m.

Thus, the packets of τ_j that can delay the execution of m are those arrived on node $first_{j,i}$ in the interval $[a^{first_{j,i}}_{f(first_{j,i})} - Smax^{first_{j,i}}_j - J^1_j \ , \ a^{first_{j,i}}_m - Smin^{first_{j,i}}_j]$, where $Smax^{first_{j,i}}_j$ (respectively $Smin^{first_{j,i}}_j$) denotes the maximum time (respectively the minimum time) taken by a packet of flow τ_j to arrive on node $first_{j,i}$. The maximum number of packets of τ_j that can delay m is then equal to:

$$1 + \left\lfloor \frac{a^{first_{j,i}}_m - Smin^{first_{j,i}}_j - \left(a^{first_{j,i}}_{f(first_{j,i})} - Smax^{first_{j,i}}_j - J^1_j \right)}{T_j} \right\rfloor.$$

By definition, $Smax_j^{first_{j,i}} - Smin_j^{first_{j,i}} + J_j = J_j^{first_{j,i}}$, where $J_j^{first_{j,i}}$ denotes the delay jitter experienced by τ_j between its source node and node $first_{j,i}$. Applying this property to all the EF flows visiting at least one node visited by τ_i, we get:

$$\sum_{g=f(1)}^{m} C_{\tau(g)}^{slow_{\tau(g),i}} \leq \sum_{h=1}^{q} \left(\sum_{\substack{j=1 \\ first_{j,i}=h}}^{n} \left(1 + \left\lfloor \frac{a_m^h - a_{f(h)}^h + J_j^h}{T_j} \right\rfloor \right) \cdot C_j^{slow_{j,i}} \right).$$

As $a_m^h - a_{f(h)}^h \leq a_m^h - t + t - a_{f(1)}^1$, we get $a_{f(1)}^1 - t + \sum_{g=f(1)}^{m} C_{\tau(g)}^{slow_{\tau(g),i}}$ bounded by:

$$a_{f(1)}^1 - t + \sum_{h=1}^{q} \left(\sum_{\substack{j=1 \\ first_{j,i}=h}}^{n} \left(1 + \frac{a_m^h - t + t - a_{f(1)}^1 + J_j^h}{T_j}\right) \cdot C_j^{slow_{j,i}} \right)$$

$$\leq \sum_{h=1}^{q} \left(\sum_{\substack{j=1 \\ first_{j,i}=h}}^{n} \left(1 + \frac{Smax_i^h + J_j^h}{T_j}\right) \cdot C_j^{slow_{j,i}} \right) + \left(t - a_{f(1)}^1\right) \cdot \left(\sum_{h=1}^{q} \left(\sum_{\substack{j=1 \\ first_{j,i}=h}}^{n} \frac{C_j^{slow_{j,i}}}{T_j} \right) - 1 \right).$$

Hence, if $\sum_{h=1}^{q} \left(\sum_{\substack{j=1 \\ first_{j,i}=h}}^{n} (C_j^{slow_{j,i}}/T_j) \right) \leq 1$, the maximum delay due to EF packets and incurred by m is bounded by:

$$X_{EF} \leq \sum_{h=1}^{q} \left(\sum_{\substack{j=1 \\ first_{j,i}=h}}^{n} \left(1 + \frac{Smax_i^h + J_j^h}{T_j}\right) \cdot C_j^{slow_{j,i}} \right) + \sum_{\substack{h=1 \\ h \neq slow_i}}^{q} C_{max}^h.$$

3.2.2 Non-preemptive effect

As packet scheduling is non-preemptive, whatever the scheduling algorithm in the EF class and despite the highest priority of this class, a packet from another class (i.e. Best-Effort or AF class) can interfere with EF flows processing due to non-preemption. Indeed, if any EF packet m enters node $h \in \mathcal{P}_i$ while a packet m' not belonging to the EF class is being processing on this node, m has to wait until m' completion.

Generally, the non-preemptive effect may lead to consider EF packets with a priority higher than m arrived after m, but during m' execution. As we consider in this paper that the scheduling of the EF class is FIFO, no EF packet has priority over m if arrived after m. Hence, we get the following property.

Property 1 *The maximum delay due to packets not belonging to the EF class incurred by any packet of the* EF *flow τ_i generated at time t meets: $\delta_i(t) \leq \sum_{h \in \mathcal{P}_i} (B^h - 1)$, where B^h denotes the maximum processing time at node h of any packet of flow not belonging to the* EF *class. By convention, $B^h - 1 = 0$ if there is no such packets.*

Proof: Let us show that if m is delayed on any node h of path $\mathcal{P}_i$ by a packet m' not belonging to the EF class, then in the worst case the packet m arrives on node h just after the execution of m' begins. Indeed, by contradiction, if m arrives on the node before or at the same time as m', then as m belongs to the EF class and this class has the highest priority, m will be processed before m'. Hence a contradiction. Thus, in the worst case, m arrives on node h one time unit after the execution of m' begins, because we assume a discrete time. The maximum processing time for m' on h is B^h. By convention, if all packets visiting node h belong to the EF class, we note $B^h - 1 = 0$. Hence, in any case, the maximum delay incurred by m on h directly due to a packet not belonging to the EF class is equal to $B^h - 1$. As we make no particular assumption concerning the classes other than the EF class, in the worst case packet m is delayed for $B^h - 1$ on any visited node h. ■

It is important to notice that this bound can be optimized if we have additionnal information concerning flows not belonging to the EF class.

3.2.3 Worst case end-to-end response time of an EF packet

From previous subsections, we have R_{max_i}, the maximum end-to-end response time of EF flow τ_i, $i \in [1, n]$, bounded by:$X_{EF} + \sum_{h=1}^{q}(B^h - 1) + (q-1) \cdot P_{max}$. Hence, considering the above defined bound for X_{EF}, we finally get:

Property 2 *If the condition* $\sum_{h=1}^{q}\left(\sum_{\substack{j=1 \\ first_{j,i}=h}}^{n}(C_j^{slow_{j,i}}/T_j)\right) \leq 1$ *is met, then the worst case end-to-end response time of any packet of* EF *flow* τ_i *is bounded by:*

$$R_{max_i} \leq \sum_{h \in \mathcal{P}_i}\left(\sum_{\substack{j=1 \\ first_{j,i}=h}}^{n}\left(1 + \frac{Smax_i^h + J_j^h}{T_j}\right) \cdot C_j^{slow_{j,i}}\right) + \sum_{\substack{h \in \mathcal{P}_i \\ h \neq slow_i}} C_{max}^h + \sum_{h \in \mathcal{P}_i}(B^h - 1) + (|\mathcal{P}_i| - 1) \cdot P_{max}.$$

where $slow_{j,i}$ *is the slowest node visited by* τ_j *on path* $\mathcal{P}_i$ *and* $first_{j,i}$ *is the first node visited by* τ_j *on path* $\mathcal{P}_i$.

We can notice that this bound on the worst case end-to-end response time is more accurate than the sum of the maximum sojourn times on the visited nodes, plus the sum of the maximum network delays. This shows the interest of the trajectory approach comparatively to the holistic one.

Remark: Particular case of the same path

In the particular case of a single path consisting of q nodes numbered from 1 to q, we notice that:

- the condition given in Property 2 becomes $\sum_{j=1}^{n} C_j^{slow}/T_j \leq 1$, that is a necessary condition for the feasibility of a set of flows;
- the bound on the end-to-end response time given in Property 2 becomes:

$$R_{max_i} \leq \sum_{j=1}^{n} \left(1 + \frac{J_j}{T_j}\right) \cdot C_j^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^{q} C_{max}^{h} + \sum_{h=1}^{q} (B^h - 1) + (q-1) \cdot P_{max}.$$

3.3 Computation algorithm

For any EF flow τ_i, we define $first_i$ and $last_i$ as respectively the first and the last nodes visited by τ_i. Moreover, we define $pre_i(h)$ the node visited by τ_i before node h. Finally, we denote $Rmax_i^h$ the maximum delay for τ_i to visit nodes $first_i$ to h.

To compute the worst case end-to-end response time of any flow τ_i when Assumption 8 is met, we apply the following algorithm:

- we first determine the set $\mathcal{S}_i$ of EF flows crossing directly or indirectly flow τ_i, that is any EF flow τ_j belongs to $\mathcal{S}_i$ iff τ_j directly crosses τ_i or an EF flow $\tau_k \in \mathcal{S}_i$. By definition, $\tau_i \in S_i$;
- we then initialize for the iteration $a = 1$ the value of $Smax_j^h(a)$ to $\sum_{h'=first_j}^{pre_j(h)} (C_j^{h'} + P_{max})$ for any flow $\tau_j \in S_i$ and any node h such that there is a flow τ_k with $h = first_{k,j}$;
- we proceed iteratively:

```
a = 0
Repeat
  a = a + 1 /* iteration a*/
  for any flow τj ∈ Si
    for any node h = firstj to lastj
      if (h = lastj) or (∃ τk ∈ Si crossing τj such that h = prej(firstk,j)) then
          compute Rmax_j^h using Property 2, with Smax_j^h(a)
          and J_k'^h for any flow k' such that h = first_k',j
          if ∃ τk ∈ Si such that h = prej(firstk,j) then
             Smax_j^{first_k,j}(a + 1) = Rmax_j^h + Pmax

Until (∃τj ∈ Si, Rj > Dj)
   or (∀τj and τk ∈ Si, ∀h = prej(firstk,j), Smax_j^{first_k,j}(a + 1) = Smax_j^{first_k,j}(a))
```

3.4 Generalization

Property 2 can be extended to the general case by decomposing an EF flow in as many independent flows as needed to meet Assumption 8. To achieve that, the idea is to consider an EF flow crossing path $\mathcal{P}_i$ after it left $\mathcal{P}_i$ as a new EF flow. We proceed by iteration until meeting Assumption 8. We then apply Property 2 considering all these flows.

3.5 Example and discussion

Let us consider the following example with six EF flows whose characteristics are given in Table 1. These flows visit four nodes. Flows not belonging to the EF class, denoted $\overline{\text{EF}}$, represent a workload of 20% on each node. In this example, the worst case end-to-end response time of any EF flow is equal to 66 time units.

Class	Flow	Execution duration in time unit	Interarrival time in time unit
EF	τ_1	3	20
	τ_2	4	50
	τ_3	8	30
	τ_4	7	100
	τ_5	10	90
	τ_6	2	40
$\overline{EF}$		3	15

Table 1: Traffic parameters

The deterministic approach may lead to a bound that can be reached infrequently. A network dimensioning based on this bound can be expensive in terms of resources. Moreover, many applications do not require such guarantees. That is why, we are interested in probabilistic QoS guarantees concerning the end-to-end response time. That is QoS is met with a given probability. An admission control, based on these results, can be derived.

4 Probabilistic Approach

The probabilistic approach aims to alleviate flow constraints obtained with the deterministic approach. For instance if we consider the end to end delay, the probabilistic approach will allow each flow to miss its end to end deadline with a certain probability. In this section, we consider a study based on queuing theory to derive the end to end response time distribution for the set of real time flows described in Section 3 by $\{\tau_1, \tau_2, ..., \tau_n\}$ aggregated over the EF class in the MPLS/DiffServ network. The assumptions used for this study are those given in Section 2 plus additionnal ones specific to the probabilistic approach.

Through each node, EF flows will be served by a non preemptive scheduler in a strict priority over other DiffServ classes. All the packets belonging to non EF class will be referred by $\overline{\text{EF}}$ packets. $\overline{\text{EF}}$ is a virtual class defined in the probabilistic model to evaluate the non preemptive effect of the node scheduler.

4.1 Mathematical Model

We focus on the set $S = \{\tau_1, \tau_2, ..., \tau_n\}$ of the n *EF* flows. To derive the end-to-end response time distribution for these flows, we make the following simplifying assumption:

Assumption 9 *At network entries, the real-time flows arrive according to a Poisson process. Each flow τ_i is a Poisson process with parameter λ_i.*

In fact, the arrival process of the Internet traffic, at a packet scale, does not correspond to a Poisson process (On-off sources for example). However, it can be modeled by a Poisson process [7]. Since we are interested in providing QoS guarantees, considering a Poisson arrival process for real time flows packets, is a justified assumption.

The network is composed of a set Q of edge and core routers interconnected by links. Each link is characterized by its bandwidth. Links bandwidths are described by the matrix (V_{ab}), where $V_{aa} = 0$ for $a = 1...Q$ and $V_{ab} \neq 0$ if there is a link between nodes a and b.

The n flows circulate through the Q nodes of the network. Each flow τ_i, is characterized by its packet processing time, its packet interarrival time and its relative deadline. Packets belonging to the same flow, inherits from its characteristics. According to Assumption 6, all the packets of a same flow go through the same path. These paths are defined by the matrix $F = (F_{ia})_{i=1...n, a=1...Q}$ where F_{ia} gives for the flow τ_i, the row of the node a in its path. If τ_i doesn't cross the node a, then $F_{ia} = 0$. Hence we define δ^i_{ab} by

$$\delta^i_{ab} = \begin{cases} 1 \text{ if } F_{ib} - F_{ia} = 1 \quad \text{with } F_{ib} > 0, F_{ia} > 0 \text{ and } V_{ab} > 0 \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

That is $\delta^i_{ab} = 1$ if the link ab (node a to node b) belongs to the path of the flow τ_i.

To compute the end-to-end response time distribution, we must first focus on a node response time distribution.

4.2 Node response time distribution

A node can be considered as a set of queuing systems. Arriving packets are stocked in a first queue to be processed and switched over the appropriate link. Each link ab corresponds to a queuing system, where the service is the transmission of a packet over this link. By supposing that the processing time at the first queue is instantaneously, the node response time, for a packet going through node a to node b, corresponds to the response time of the

queuing system modelling the link ab. To simplify this study, we introduce this assumption.

Assumption 10 *We suppose that packet arrivals of a flow τ_i to a link ab also follow a Poisson process with parameter*

$$\lambda_{ab}^i = \delta_{ab}^i \lambda_i \tag{2}$$

where δ_{ab}^i is given by eq.(1).

Packets of flow τ_i are served at the link ab with an average rate $\mu_{ab}^i = \frac{1}{X_{ab}^i}$. $\overline{X_{ab}^i}$ corresponds to the average service time of packets of a flow τ_i at the link ab and is given by

$$\overline{X_{ab}^i} = \frac{\beta_i}{V_{ab}}$$

where β_i is the average packet size of that flow.

At each link ab, $\rho_{ab}^{\overline{EF}}$ denotes the utilization factor of that link by non EF packets. $\rho_{ab}^{\overline{EF}}$ is a model parameter. We can therefore characterize the arrival process of $\overline{EF}$ flow packets at ab link server.

Assumption 11 *We suppose at each link ab that $\overline{EF}$ flow packets arrive according to a Poisson process with parameter $\lambda_{ab}^{\overline{EF}}$,where $\lambda_{ab}^{\overline{EF}} = \rho_{ab}^{\overline{EF}} \mu_{ab}^{\overline{EF}}$,and $\mu_{ab}^{\overline{EF}} = \frac{1}{X_{ab}^{\overline{EF}}}$. $\overline{X_{ab}^{\overline{EF}}}$ corresponds to the average service time of packets of the flow $\overline{EF}$ at the link ab and is given by*

$$\overline{X_{ab}^{\overline{EF}}} = \frac{\beta_{ab}^{\overline{EF}}}{V_{ab}}$$

and $\beta_{ab}^{\overline{EF}}$ is the average packet size of that flow.

According to the traffic description above, each link ab can be modelled by an $M/G/1$ station with $n+1$ classes of customers : the n real time flows and the flow $\overline{EF}$. The scheduling discipline is the non preemptive Head Of Line (HOL) discipline. Packets of class i (flow τ_i), $i \in [1,n]$ are served at each link ab, with the highest priority over $\overline{EF}$ packets.

Let us summarize the different parameters of this model:

n : the number of real time flows belonging to the EF class.
λ_{ab}^i : the average arrival rate of class i customers (packets) at the link ab.
$\lambda_{ab}^{\overline{EF}}$: the average arrival rate of class $\overline{EF}$ customers at the link ab.
μ_{ab}^i : the average service rate of class i customers at the link ab.
$\mu_{ab}^{\overline{EF}}$: the average arrival rate of class $\overline{EF}$ customers at the link ab.
$\rho_{ab}^{\overline{EF}}$: the utilization factor of the link sever ab by the customers of class $\overline{EF}$.

$\widetilde{x_{ab}^i}$: the random variable : the service time of class i customers at the link ab and $\overline{\left(X_{ab}^i\right)^m}$ its m^{th} moment.

$\widetilde{x_{ab}^{\overline{EF}}}$: the random variable : the service time of packets of class $\overline{EF}$ at the link ab and $\overline{\left(X_{ab}^{\overline{EF}}\right)^m}$ its m^{th} moment.

λ_{ab} : the average arrival rate of packets to the link ab. We have

$$\lambda_{ab} = \sum_{i=1}^{n} \lambda_{ab}^i + \lambda_{ab}^{\overline{EF}} \tag{3}$$

ρ_{ab}^i : the utilization factor of the link sever ab by the customers of class i. It's given by

$$\rho_{ab}^i = \lambda_{ab}^i \overline{X_{ab}^i} \tag{4}$$

ρ_{ab} : the utilization factor of the link sever ab. Hence

$$\rho_{ab} = \sum_{i=1}^{n} \rho_{ab}^i + \rho_{ab}^{\overline{EF}} \tag{5}$$

$\widetilde{w_{ab}^i}$: the random variable : the waiting time of class i customers at the link ab.

$\widetilde{s_{ab}^i}$: the random variable : the stay time of class i customers at the link ab.

The node response time distribution for class i customers at the link ab is obtained by inspecting its Laplace transform which will be denoted by $\left(S_{ab}^i\right)^*(s)$ and is given by

$$\left(S_{ab}^i\right)^*(s) = \left(W_{ab}^i\right)^*(s) \ \left(B_{ab}^i\right)^*(s) \tag{6}$$

where $\left(B_{ab}^i\right)^*(s)$ is the Laplace transform of the service time probability density function of class i customers at the link ab and $\left(W_{ab}^i\right)^*(s)$ is the Laplace transform of the waiting time density of of class i customers at the same link. We first focus on the computation of $\left(W_{ab}^i\right)^*(s)$. A packet belonging to the class i must wait for:

- packets of classes j, $j \in [1, n]$ found in the queue upon the arrival of our tagged packet,
- and the packet found in service upon the arrival of our tagged packet.

For a packet of class i, we define as in [24], two categories of packets:

- $\mathfrak{C}_{ab}^+$ which represents flow packets of high priority belonging to the customer classes $1..n$.
- $\mathfrak{C}_{ab}^-$ which represents the packets of low priority belonging to the customer class $\overline{EF}$.

The Poisson arrival rates of these two packets categories are given by

$$\lambda_{ab}^{+} = \sum_{i=1}^{n} \lambda_{ab}^{i} \tag{7}$$

$$\lambda_{ab}^{-} = \lambda_{ab}^{\overline{EF}} \tag{8}$$

The server utilization factors for these two packets categories are

$$\rho_{ab}^{+} = \sum_{i=1}^{n} \rho_{ab}^{i} \tag{9}$$

and

$$\rho_{ab}^{-} = \rho_{ab}^{\overline{EF}} \tag{10}$$

The Laplace transforms of the service time densities are respectively

$$\left(B_{ab}^{+}\right)^{*}(s) = \sum_{i=1}^{n} \frac{\lambda_{ab}^{i}}{\lambda_{ab}^{+}} \left(B_{ab}^{i}\right)^{*}(s) \tag{11}$$

and

$$\left(B_{ab}^{-}\right)^{*}(s) = \left(B_{ab}^{\overline{EF}}\right)^{*}(s) \tag{12}$$

Note that the waiting time of packets of class i, is invariant to the change in the order of service. Let $\left(W_{ab}^{+}\right)^{*}(s)$ the Laplace transform of the waiting time density of high priority packets (packets belonging to $\mathfrak{C}_{ab}^{+}$). $\left(W_{ab}^{+}\right)^{*}(s)$ is given by [24],[17]

$$\left(W_{ab}^{+}\right)^{*}(s) = \frac{(1-\rho_{ab})\, s + \lambda_{ab}^{-}\left(1 - \left(B_{ab}^{-}\right)^{*}(s)\right)}{s - \lambda_{ab}^{+} + \lambda_{ab}^{+}\left(B_{ab}^{+}\right)^{*}(s)} \tag{13}$$

where ρ_{ab} is given by eq.(4).

By coming back to the original system, the Laplace transform of the waiting time density function for class i customers at the link ab, $\left(W_{ab}^{i}\right)^{*}(s)$, is the same for all the customers of classes $j, j \in [1, N]$ since all these flows are served with FIFO discipline. It's equal to $\left(W_{ab}^{+}\right)^{*}(s)$. Hence, $\left(W_{ab}^{i}\right)^{*}(s)$ equals

$$\left(W_{ab}^{i}\right)^{*}(s) = \left(W_{ab}^{+}\right)^{*}(s) \tag{14}$$

The Laplace transform of the node sojourn time distribution for class i at ab, $\left(S_{ab}^{i}\right)^{*}(s)$ is then given by

$$\left(S_{ab}^{i}\right)^{*}(s) = \left(B_{ab}^{i}\right)^{*}(s) \frac{(1-\rho_{ab})\, s + \lambda_{ab}^{\overline{EF}}\left(1 - \left(B_{ab}^{\overline{EF}}\right)^{*}(s)\right)}{s - \sum_{i=1}^{n} \lambda_{ab}^{i} + \sum_{i=1}^{n} \lambda_{ab}^{i}\left(B_{ab}^{i}\right)^{*}(s)} \tag{15}$$

The node response time distribution is obtained by inspecting its Laplace transform given by eq.(15).

4.3 End-to-end response time distribution

Let $\widetilde{s_i}$ the random variable : the end-to-end response time for a packet of class of customer i and $S_i^*(s)$ the Laplace transform of its probability density function. The end-to-end response time corresponds to the time needed to go from the ingress node to the egress one. The random variable $\widetilde{s_i}$ corresponds to the sum of the nodes response times crossed by the packet while going through the network and the sum of the transmission delays between the different nodes.

$$\widetilde{s_i} = \sum_{a=1}^{Q}\sum_{b=1}^{Q}\delta_{ab}^{i}\left(\widetilde{s_{ab}^{i}} + \widetilde{L}\right) \tag{16}$$

where $\widetilde{L}$ is the random variable corresponding to the propagation delays between two nodes.

The random variables corresponding to these different durations being independent, we obtain

$$S_i^*(s) = \left(L^*(s)\right)^{\sum_{a=1}^{Q}\sum_{b=1}^{Q}\delta_{ab}^{i}} \prod_{a,b\in[1..Q]} \left(\left(\widehat{S_{ab}^{i}}\right)^*(s)\right) \tag{17}$$

where

$$\left(\widehat{S_{ab}^{i}}\right)^*(s) = \begin{cases} 1 & \text{if } \delta_{ab}^{i} = 0 \\ \left(S_{ab}^{i}\right)^*(s) & \text{else} \end{cases} \tag{18}$$

and $L^*(s)$ is the Laplace transform of $\widetilde{L}$ probability density function. According to assumption 3,

$$L^*(s) = \frac{e^{-sPmin} - e^{-sPmax}}{s\left(P_{max} - P_{min}\right)} \tag{19}$$

Figure 3 presents the end-to-end response time distribution, obtained for a network composed of six nodes with three real time flows, ($n = 3$), fixed paths and two priority levels EF and $\overline{EF}$. Inter arrival packet times are exponentially distributed. The curves deal with the packets of an EF test flow τ_1. They are obtained for different utilization factors of the network bottleneck link $3 \rightarrow 4$, denoted by $\rho_{34} = (0.7, 0.8, 0.9)$, and a fixed load in class $\overline{EF}$, $\rho_{34}^{\overline{EF}} = 60\%$.

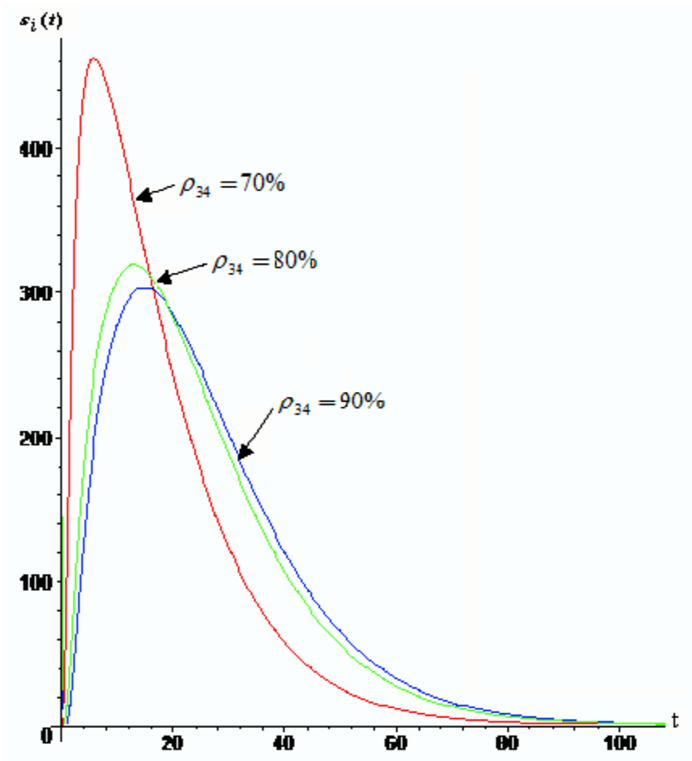

Figure 3: End-to-end response time distribution

These curves show that the end-to-end response time distribution becomes very small from a given value of t. We can then anticipate and provide the range of flow deadline values for which the flow can go through the network without missing its deadline.

5 Probabilistic QoS guarantee and admission control

The end-to-end response time distribution enables to determine, for a given configuration, the probability that a flow packet won't stay in the network more than a given duration. Indeed, a packet belonging to the flow τ_i and arriving with a relative deadline D_i won't miss its deadline with the probability $P_{success}(D_i)$ and will miss its deadline with the probability $P_{miss}(D_i) = 1 - P_{success}(D_i)$

$$P_{success}(D_i) \quad = \quad P\left[\widetilde{s_i} < D_i\right] = \int_0^{D_k} s_i(t)dt \tag{20}$$

where $s_i(t)$ is the end-to-end response time distribution obtained by inspecting its Laplace transform.

If we consider the configuration described in Section 3, Table 2 gives the missing deadline probabilities for the flow τ_5,which has the maximum service time, for different loads in class $EF = (10\%, 20\%, 30\%)$ and the same load for class $\overline{EF}$ equal to 20%. We recall that delay bound obtained with the deterministic approach is equal to 66 time units. With the probabilistic approach, this bound is guaranteed with a probability of $6 \cdot 10^{-4}$ for a load of 10%. In fact, as a Poisson arrival process has been considered, the response time cannot in any case equal zero. However, the network can guarantee a delay of 60 time units with a probability of $7.7 \cdot 10^{-3}$, a delay of 55 time units with a probability of $7.3 \cdot 10^{-2}$. These probabilities can be satisfying for some applications.

	D_i (ms)				
U_{EF}	45	50	55	60	66
10%	$8.4 \cdot 10^{-1}$	$3.9 \cdot 10^{-1}$	$7.3 \cdot 10^{-2}$	$7.7 \cdot 10^{-3}$	$6.1 \cdot 10^{-4}$
20%	$9.1 \cdot 10^{-1}$	$5.2 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$	$3.3 \cdot 10^{-2}$	$5.3 \cdot 10^{-3}$
30%	$9.4 \cdot 10^{-1}$	$6.2 \cdot 10^{-1}$	$2.6 \cdot 10^{-1}$	$8.8 \cdot 10^{-1}$	$1.8 \cdot 10^{-2}$

Table 2: Deadline miss probability with regard to relative deadline.

A probabilistic admission control can be defined, it provides a probabilistic QoS guarantee to each accepted flow. A new EF flow τ_i is accepted if and only if:

- The QoS requested by flow τ_i can be met with a probability p computed from the flow characteristics (i.e. its arrival rate and deadline) according to the method presented previously in this Section.
- The probabilistic QoS of already accepted EF flows is not compromised by the acceptance of flow τ_i.

A negotiation can take place between the admission control and the user submitting the new flow. For instance, if the user is not pleased with the probability computed, the admission control can propose other deadline values with the probabilities associated. The user selects the most appropriate deadline.

6 Conclusion

In this paper, we have proposed a solution to provide quantitative end-to-end real-time guarantees for flows in the EF class of the DiffServ model, assuming that this class has the highest priority and packets are scheduled FIFO within the EF class. The EF class is well adapted for flows with real-time constraints such as voice or video flows. We have computed deterministic bounds on the end-to-end response time of any EF flow. This bound, obtained in the worst case scenarios, can be reached infrequently and leads to network overdimensioning. That is why we have developed mathematical models to evaluate the probability of

meeting a specified end-to-end delay for any EF flow in the DiffServ/MPLS domain.

The MPLS technology reduces forwarding delays because of simpler processing and allows to indicate in a label the service class of the packet. This study has shown that we can guarantee delays much smaller than the deterministic bound with a probability close to one.

Moreover, this probabilistic approach yields to define an admission control based on a probabilistic guarantee of the required deadline. As a further work, we will study an Earliest Deadline First, (EDF), scheduling policy within the EF class instead of FIFO to favor flows having small deadlines.

References

[1] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda, T. Przygienda, *QoS routing mechanisms and OSPF extensions*, RFC 2676, August 1999.

[2] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, *RSVP-TE : Extensions to RSVP for LSP tunnels*, Internet draft, August 2001.

[3] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManusWang, *Requirements for traffic engineering over MPLS*, RFC 2702, September 1999.

[4] S. Baruah, R. Howell, L. Rosier, *Algorithms and complexity concerning the preemptive scheduling of periodic real-time tasks on one processor*, Real-Time Systems, 2, p 301-324, 1990.

[5] J. Bennett, K. Benson, A. Charny, W. Courtney, J. Le Boudec, *Delay jitter bounds and packet scale rate guarantee for Expedited Forwarding*, INFOCOM'2001, Anchorage, USA, April 2001.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, *An architecture for Differentiated Services*, RFC 2475, December 1998.

[7] T. Bonald, A. Proutière, J. Roberts, *Statistical performance guarantees for streaming flows using expedited forwarding*, in Proc. of IEEE INFOCOM'2001, March 2001.

[8] R. Braden, D. Clark, S. Shenker, *Integrated services in the Internet architecture: an overview*, RFC 1633, June 1994.

[9] A. Charny, J. Le Boudec, *Delay bounds in a network with aggregate scheduling*, QoFIS, Berlin, Germany, October 2000.

[10] F. Chiussi, V. Sivaraman, *Achieving high utilization in guaranteed services networks using early-deadline-first scheduling*, IWQoS'98, Napo, California, May 1998.

[11] L. Georgiadis, R. Guérin, V. Peris, K. Sivarajan, *Efficient network QoS provisioning based on per node traffic shaping*, IEEE/ACM Transactions on Networking, Vol. 4, No. 4, August 1996.

[12] L. George, S. Kamoun, P. Minet, *First come first served: some results for real-time scheduling*, PDCS'01, Dallas, Texas, August 2001.

[13] L. George, D. Marinca, P. Minet, *A solution for a deterministic QoS in multimedia systems*, International Journal on Computer and Information Science, Vol.1, N3, September 2001.

[14] M. Gerla, C. Casetti, S. Lee, G. Reali, *Resource allocation and admission control styles on QoS DiffServ networks*, QoS-IP 2001, Rome, Italy, 2001.

[15] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, *Assured Forwarding PHB group*, RFC 2597, 1999.

[16] V. Jacobson, K. Nichols, K. Poduri, *An Expedited Forwarding PHB*, RFC 2598, June 1999.

[17] L. Kleinrock, *Queuing Systems, Volume II:Computer Applications*, Wiley Interscience, pp. 100-120, 1976.

[18] J. Le Boudec, P. Thiran, *A note on time and space methods in network calculus*, Technical Report, No. 97/224, Ecole Polytechnique Fédérale de Lausanne, Swiss, April 11, 1997.

[19] F. Le Faucheur, L. Wu, S. Davari, et al. *MPLS support of Differentiated Services*, Internet draft, 2000.

[20] A. Parekh, R. Gallager, *A generalized processor sharing approach to flow control in integrated services networks: the multiple node case*, IEEE ACM Transactions on Networking, Vol.2, N2, 1994.

[21] D. Raz, Y. Shavitt, *Optimal partition of QoS requirements with discrete cost functions*, INFOCOM'2000, Tel-Aviv, Israel, March 2000.

[22] V. Sivaraman, F. Chiussi, M. Gerla, *Traffic shaping for end-to-end delay guarantees with EDF scheduling*, IWQoS'2000, Pittsburgh, June 2000.

[23] V. Sivaraman, F. Chiussi, M. Gerla, *End-to-end statistical delay service under GPS and EDF scheduling: a comparaison study*, INFOCOM'2001, Anchorage, Alaska, April 2001.

[24] H. Takagi, *Queueing Analysis: A Foundationof Performance Evaluation Vol. 1. Vacation and Priority Systems. Part 1*, North-Holland, Amsterdam, pp. 287-292, 1991.

[25] K. Tindell, J. Clark, *Holistic schedulability analysis for distributed hard real-time systems*, Microprocessors and Microprogramming, Euromicro Journal, Vol. 40, pp. 117-134, 1994.

[26] M. Vojnovic, J. Le Boudec, *Stochastic analysis of some expedited forwarding networks*, INFOCOM'2002, New York, June 2002.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399