FP/FIFO scheduling: coexistence of deterministic and probabilistic QoS guarantees

Pascale Minet[1] and Steven Martin[2]
and Leila Azouz Saidane[3] and Skander Azzaz[3]

[1]*INRIA Rocquencourt, 78153 Le Chesnay, France*
[2]*LRI, Université Paris-Sud, 91405 Orsay, France*
[3]*ENSI, CRISTAL, 2010, Tunis, Tunisia*

received June 2006, *revised January 2007*, *accepted April 2007*.

In this paper, we focus on applications with quantitative QoS (*Quality of Service*) requirements in their end-to-end response time (or jitter). We propose a solution allowing the coexistence of two types of quantitative QoS guarantees, deterministic and probabilistic, while providing high resource utilization. Our solution combines the advantages of both the deterministic and the probabilistic approaches. The deterministic approach is based on a worst case analysis. The probabilistic approach uses a mathematical model to obtain the probability of the response time exceeding a given value. We assume that flows are scheduled according to non-preemptive FP/FIFO. The packet with the highest fixed priority is scheduled first. If two packets share the same priority, the packet that arrives first is scheduled first. We make no particular assumptions about the flow priority and the nature of the QoS guarantee requested by the flow. An admission control derived from these results is then proposed, allowing each flow to receive a quantitative QoS guarantee adapted to its QoS requirements. An example illustrates the merits of the coexistence of deterministic and probabilistic QoS guarantees.

Keywords: QoS, Fixed Priority scheduling, FP/FIFO, deterministic guarantee, probabilistic guarantee, admission control

1 Context and motivations

We are interested in providing quantitative QoS (*Quality of Service*) guarantees to various types of applications in their end-to-end response time (or jitter). Accordingly, the goal of this paper is to achieve quantitative QoS guarantees to such applications while providing high resource utilization. Two types of guarantees can be granted to a flow:

- a deterministic guarantee, which ensures that no packet of this flow will encounter an end-to-end response time exceeding a given deadline. For example, the delivery of an alarm in a command and control application requires a bounded delay.

- a probabilistic guarantee, which ensures that the end-to-end response time of any packet of this flow will not exceed a given deadline with a probability higher than a given value. For example, a video flow can tolerate a few packet losses.

We propose a solution offering a good trade-off between the deterministic and the probabilistic approaches. Indeed, the deterministic approach is based on a worst case analysis and can lead to a low resource utilization. However, providing only probabilistic guarantees is not acceptable for applications requiring hard deadlines. That is why we have investigated a solution allowing both guarantee types to coexist. Such a solution should lead to a better resource utilization. The admission control presented in this paper will allow us to accept more flows and will offer to each of them a quantitative QoS guarantee in accordance with its requirements.

Our solution is based on the *Fixed Priority* scheduling [1, 2] that exhibits interesting properties:

- It favors flows with the highest fixed priority. Then, the fixed priority of a flow can be easily assigned to reflect the flow's degree of importance.
- The impact of a new flow τ_i is limited to flows having priorities smaller than that of τ_i.
- It is easy to implement.
- It is well adapted for service differentiation: flows with high priorities have smaller response times.

In this paper, we focus on non-preemptive Fixed Priority scheduling. Indeed, with regard to flow scheduling, the assumption generally admitted is that packet transmission is not preemptive. Moreover, in many cases, several flows may have to share the same priority, for example when:

- the number of fixed priorities available on a processor is less than the flow number;
- the priority of a flow is determined by external constraints and cannot be chosen arbitrarily;
- flows are processed by class of service and the flow priority is that of its class.

In the state of the art, the worst case analysis assumes that flows sharing the same priority are arbitrarily scheduled. However, FIFO is the policy generally used by the *Fixed Priority* implementations to schedule flow packets having the same fixed priority. In this paper, we consider that such packets are scheduled FIFO, and, unlike the state of the art, we take this scheduling into account to compute deterministic and probabilistic guarantees. The resulting scheduling policy is called FP/FIFO. Our solution enables us to improve the worst case response times of such flows. Indeed, a packet cannot be delayed by other packets of the same priority released after it. Notice that there is no relationship between the nature of the QoS guarantee required by a flow (deterministic or probabilistic) and its fixed priority.
The paper is organized as follows. In Section 2, we define the problem we address. In Section 3, we show how to conduct a worst case analysis to provide deterministic end-to-end response times to flows requiring firm QoS guarantees. In Section 4, we present a mathematical model to obtain, for any flow requesting probabilistic QoS guarantee, the probability that its response time does not exceed a given value. We derive from deterministic and probabilistic results an admission control, presented in Section 5. The mathematical study is validated in an example given in Section 6. An extended example to illustrate these results is presented in Section 7. Finally, we give some perspectives in Section 8 and conclude the paper in Section 9.

2 Problematic of providing quantitative QoS guarantees

We investigate the problem of providing a quantitative end-to-end response time guarantee to any flow in a distributed system. This guarantee can be either deterministic or probabilistic depending on the flow QoS requirements. We do not make any assumption concerning the scheduling priority of flows with deterministic QoS requirements versus flows with probabilistic QoS requirements.

2.1 Scheduling model

We adopt the following assumption concerning the scheduling model.

Assumption 1 *Flows are scheduled according to* FP/FIFO. *With* FP/FIFO*, packets are first scheduled according to their fixed priority. Packets with the same fixed priority are scheduled according to their arrival order on the node considered.*

Notice that this solution has no particular requirement regarding the priority of flows requesting deterministic QoS guarantees versus flows requesting probabilistic ones.

Assumption 2 *Packet scheduling is non-preemptive: the scheduler of the node considered waits for the completion of the current packet transmission (if any) before selecting the next packet.*

2.2 Network model

We adopt the following assumptions concerning the network considered.

Assumption 3 *Links interconnecting nodes are* FIFO.

Assumption 4 *The network delay between two nodes has known lower and upper bounds:* $Lmin$ *and* $Lmax$.

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

2.3 Traffic model

We consider a set $\{\tau_1, \tau_2, ..., \tau_n\}$ of n flows and adopt the following assumptions.

Assumption 6 *Each flow* τ_i *follows a fixed*[(i)] *route* $\mathcal{H}_i$ *that is an ordered sequence of nodes whose first node is the ingress node of the flow.*

Assumption 7 *Flows are characterized by sporadic arrivals. Hence, each flow* τ_i *is defined by:*

- T_i*, the minimum interarrival time (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*. This parameter depends on the maximum packet size and the capacity of its output link;*
- J_i*, the maximum release jitter of packets of* τ_i *arriving in the network considered. A packet is subject to a release jitter if there exists a non-null delay between its generation time and the time called its released time where it is taken into account by the scheduler;*
- D_i*, the end-to-end deadline required by* τ_i*;*

(i) For instance, MPLS [3] can be used to fix the route followed by a flow.

- *F_i, the fixed priority of τ_i;*
- *P_i, the probability required by τ_i to meet its deadline.*

For any flow τ_i, we then define the following three sets:

- $hp_i = \{j \in [1,n]\ ,\ F_j > F_i\}$, the set of flows having a fixed priority strictly higher than that of τ_i;
- $sp_i = \{j \in [1,n]\ ,\ j \neq i\ ,\ F_j = F_i\}$, the set of flows having a fixed priority equal to that of τ_i;
- $lp_i = \{j \in [1,n]\ ,\ F_j < F_i\}$, the set of flows having a fixed priority strictly lower than that of τ_i.

Moreover, a flow requires either a deterministic or a probabilistic guarantee. Then, we can define two disjoint sets:

- $\mathbb{D} = \{\tau_i, i \in [1,n], \text{such that } P_i = 1\}$, the set of flows requiring deterministic guarantees;
- $\mathbb{P} = \{\tau_i, i \in [1,n], \text{such that } P_i < 1\}$, the set of flows requiring probabilistic guarantees.

To provide probabilistic guarantees to flows belonging to $\mathbb{P}$, we use the following property.

Property 1 *The sporadic arrivals of any flow τ_i can be upper bounded by Poisson arrivals characterized by: $\lambda_i = 1/T_i$ and $\mu_i^h = 1/\overline{C}_i^h$, with $\overline{C}_i^h$ the average processing time of a packet of τ_i in node h.*

Proof: See [4]. □

3 Deterministic approach for computing the worst case end-to-end response times

3.1 Related work

Deterministic and quantitative guarantees can be provided by at least three approaches, which compute the worst case end-to-end response time of any flow:

The holistic approach [5, 6]. This approach, the first introduced in the literature, considers the worst case scenario on each node visited by a flow, taking into account the maximum possible jitter introduced by the previous visited nodes. The minimum and maximum response times on a node h induce a maximum jitter on the next visited node $h+1$, which leads to a worst case response time and then a maximum jitter on the following node, and so on. This approach can be pessimistic as it considers worst case scenarios on every node, possibly leading to impossible scenarios. Indeed, a worst case scenario for a flow τ_i on a node h does not generally result in a worst case scenario for τ_i on any node visited after h.

The network calculus approach [7]. Network Calculus is a powerful tool which has been recently developed to solve flow problems encountered in networking. Indeed, considering a network element characterized by a service curve and all the arrival curves of flows visiting this element, it is possible to compute the maximum delay of any flow, the maximum size of the waiting queue and the departure curves of flows. Results of such analysis are deterministic, provided that the arrival and service curves are deterministic. As bounds are generally used instead of the exact knowledge of the arrival and service curves, this approach can lead to an overestimation of the bounds on the end-to-end response times.

The trajectory approach. This approach considers the worst case scenario that can happen to a message along its trajectory, i.e., the sequence of nodes visited. This approach is described in this section.

3.2 Notations

We consider any flow τ_i, $i \in [1, n]$, following a path $\mathcal{H}_i$, and focus on the packet m of τ_i generated at time t. We adopt the following defintition and notations:

Definition 1 *Let m be the packet of flow τ_i generated at time t. Let m' be the packet of flow τ_j generated at time t'. On any node $h \in \mathcal{H}_i \cap \mathcal{H}_j$, priority of packet m is higher than or equal to this of packet m' if and only if: ($F_i > F_j$) or ($F_i = F_j$ and m arrives before m' on node h).*

- τ_i, a sporadic flow of the set $\{\tau_1, ..., \tau_n\}$;
- R_i, the worst case response time of flow τ_i;
- m, the packet of flow τ_i generated at time t;
- $W_{i,t}^h$, the latest starting time of packet m on node h;
- $first_i$, the first node visited by flow τ_i in the network;
- $last_i$, the last node visited by flow τ_i in the network;
- $\mathcal{H}_i = [first_i, ..., last_i]$, the path followed by flow τ_i;
- $|\mathcal{H}_i|$, the number of nodes visited by flow τ_i;
- $slow_i$, the slowest node visited by flow τ_i on path $\mathcal{H}_i$: $\forall\, h \in \mathcal{H}_i$, $C_i^{slow_i} \geq C_i^h$;
- $first_{j,i}$, the first node visited by flow τ_j on path $\mathcal{H}_i$;
- $last_{j,i}$, the last node visited by flow τ_j on path $\mathcal{H}_i$;
- $slow_{j,i}$, the slowest node visited by τ_j on path $\mathcal{H}_i$: $\forall\, h \in \mathcal{H}_i \cap \mathcal{H}_j$, $C_j^{slow_{j,i}} \geq C_j^h$;
- $Smin_i^h$, the minimum time taken by a packet of flow τ_i to go from its source node to node h;
- $Smax_i^h$, the maximum time taken by a packet of flow τ_i to go from its source node to node h;
- δ_i, the maximum delay incurred by a packet of flow τ_i directly due to non-preemption when visiting path $\mathcal{H}_i$;
- $pre_i(h)$, the node visited by τ_i just before node h;
- $\tau(g)$, the index of the flow which packet g belongs to;
- $\forall a \in \mathbb{R}$, $(1 + \lfloor a \rfloor)^+$ stands for $\max(0; 1 + \lfloor a \rfloor)$;
- $M_i^h = \sum_{h'=first_i}^{pre_i(h)} (\min_{\substack{j \in hp_i \cup sp_i \cup \{i\} \\ first_{j,i} = first_{i,j}}} \{C_j^{h'}\} + L_{min})$.

By convention, $Smin_i^h = Smax_i^h = 0$ if $h \notin \mathcal{H}_i$. Moreover, Figure 1 illustrates the notations of $first_{i,j}$, $first_{j,i}$, $last_{i,j}$ and $last_{j,i}$ when flows τ_i and τ_j are (1) in the same direction and (2) in reverse directions. Moreover, we assume, with regard to flow τ_i following path $\mathcal{H}_i$, that any flow τ_j, $j \in hp_i \cup sp_i$ following path $\mathcal{H}_j$ with $\mathcal{H}_j \neq \mathcal{H}_i$ and $\mathcal{H}_j \bigcap \mathcal{H}_i \neq \emptyset$ never visits a node of path $\mathcal{H}_i$ after having left this path.

Assumption 8 *For any flow τ_i following path $\mathcal{H}_i$, for any flow τ_j, $j \in hp_i \cup sp_i$, following path $\mathcal{H}_j$ such that $\mathcal{H}_j \cap \mathcal{H}_i \neq \emptyset$, we have either $[first_{j,i}, last_{j,i}] \subseteq \mathcal{H}_i$ or $[last_{j,i}, first_{j,i}] \subseteq \mathcal{H}_i$.*

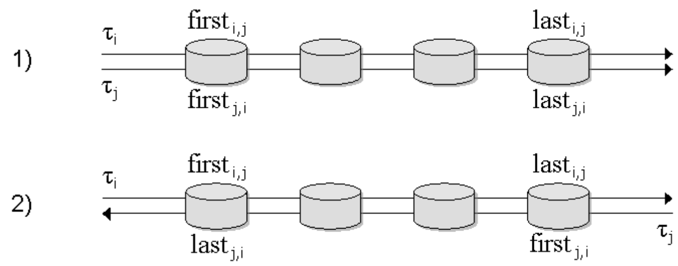

Fig. 1: $first_{i,j}, first_{j,i}, last_{i,j}$ and $last_{j,i}$

To achieve this, the idea is to consider a flow crossing path $\mathcal{H}_i$ after it has left $\mathcal{H}_i$ as a new flow. We proceed by iteration until meeting Assumption 8.

Definition 2 *The end-to-end jitter of any flow τ_i, $i \in [1,n]$, is the difference between the maximum and minimum end-to-end response times of τ_i packets, that is equal to:* $R_i - (\sum_{h\in\mathcal{H}_i} C_i^h + (|\mathcal{H}_i|-1)\cdot L_{min})$.

3.3 Study of the trajectory of packet m

Unlike the holistic approach, the trajectory approach is based on the analysis of the worst case scenario experienced by a packet on its trajectory and not on any node visited [8]. Then, only possible scenarios are examined. For instance, the fluid model is relevant to the trajectory approach. More precisely, we consider any flow τ_i, $i \in [1,n]$, following a path $\mathcal{H}_i$ consisting of q nodes numbered from 1 to q. We focus on the packet m of τ_i generated at time t.

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 m on its last node visited. For this, we adopt the trajectory approach, consisting 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.

To compute the latest starting time of packet m, we proceed as follows. We first determine bp^q, that is the busy period(ii) of level corresponding to the priority of m in which m is processed on node q. We define $f(q)$ as the first packet processed in bp^q with a priority higher than or equal to that of m. Due to the non-preemption, $f(q)$ can be delayed by at most one packet with a priority less than this of m.
As flows do not necessarily follow the same path in the network considered, it is possible that $f(q)$ does not come from node $q-1$. We then define $p(q-1)$ as the first packet processed between $f(q)$ and m such that $p(q-1)$ comes from node $q-1$. This packet has been processed on node $q-1$ in a busy period bp^{q-1} of level corresponding to the priority of $p(q-1)$. We then define $f(q-1)$ as the first packet processed in bp^{q-1} with a priority higher than or equal to this of $p(q-1)$. And so on until the busy period, on node 1, of the level corresponding to the priority of packet $p(1)$ in which the packet $f(1)$ is processed (see Fig. 2). For the sake of simplicity, on a node h, we number consecutively the packets processed after $f(h)$ and before $p(h)$ (with $p(q) = m$). Then, we denote $m' - 1$ (respectively $m' + 1$) the packet preceding

(ii) A busy period of level $\mathcal{L}$ is defined by an interval $[t, t')$ such that t and t' are both idle times of level $\mathcal{L}$ and there is no idle time of level $\mathcal{L}$ in (t, t'). An idle time t of level $\mathcal{L}$ is a time such that all packets with a priority greater than or equal to $\mathcal{L}$ generated before t have been processed at time t.

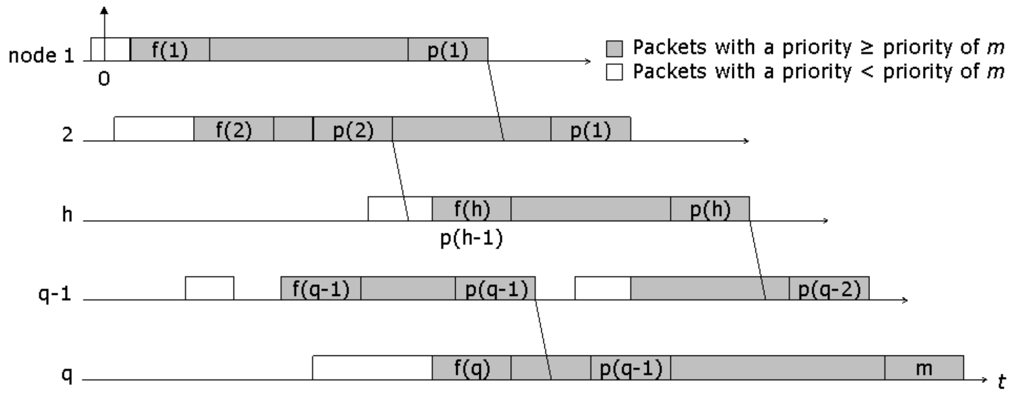

Fig. 2: Response time of packet m

(respectively succeeding to) m'. Moreover, we denote $a^h_{m'}$ the arrival time of m' on node h and consider that $a^1_{f(1)} = 0$. By adding parts of the busy periods considered, we can express the latest starting time of packet m in node q, that is:

the processing time on node 1 of packets $f(1)$ to $p(1)$ $+$ L_{max}

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

$+$ ...

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

In the worst case, $p(h) = f(h+1)$ on any node $h \in \mathcal{H}_i$. Moreover, in the worst case, on any node h visited by τ_i, the fixed priority of the packet $f(h)$ is that of packet m. Thus, we get:

$$W^q_{i,t} = \sum_{h=1}^{q}(\sum_{g=f(h)}^{f(h+1)} C^h_{\tau(g)}) \ - C^q_i \ + \delta_i \ + (q-1)\cdot L_{max}.$$

Then, the latest starting time of packet m, generated at time t, consists of three parts:

- $X_{i,t} = \sum_{h=1}^{q}(\sum_{g=f(h)}^{f(h+1)} C^h_{\tau(g)}) \ - C^q_i$, the delay due to packets with a priority $\geq$ priority of m;
- δ_i, the delay due to the non-preemptive effect;
- $(q-1)\cdot L_{max}$, the maximum network delay.

We evaluate $X_{i,t}$ and δ_i in the two following subsections.

3.4 Delay due to higher priority packets

We now evaluate the maximum delay incurred by m due to packets with a priority higher than or equal to this of m. This delay is equal to: $X_{i,t} = \sum_{h=1}^{q}(\sum_{g=f(h)}^{f(h+1)} C^h_{\tau(g)}) - C^q_i$. By definition, for any node $h \in [1,q)$, $f(h+1)$ is the first packet with a priority higher than or equal to this of m, processed in bp^{h+1} and coming from node h. Moreover, $f(h+1)$ is the last packet considered in bp^h. Let us show that in this sum, if we count packets processed in bp^h and bp^{h+1}, only $f(h+1)$ is counted twice.

Lemma 1 *For any flow τ_i, if there exists a node $h \in \mathcal{H}_i$ with a packet $m' \in (f(h), f(h+1))$, then for any node $h' \in \mathcal{H}_i - \{h\}$, $m' \notin (f(h'), f(h'+1))$.*

Proof: By induction. Let us consider any packet m' processed in $(f(1), f(2))$ on node 1. By definition, we have $F_{\tau(m')} \geq F_{\tau(f(2))} \geq F_i$. As m' leaves node h before $f(2)$ and links are FIFO, m' arrives on node 2 before $f(2)$. Consequently, on node 2, m' has a priority higher than $f(2)$. Arrived before $f(2)$, m' starts its transmission before $f(2)$ on node 2. As on this node, the busy period starts with $f(2)$, the processing of m' is completed at the latest at the arrival of $f(2)$. Hence $m' \notin (f(2), f(3))$. Similarly, we show that $m' \notin (f(h'), f(h'+1))$, for any $h' \in \mathcal{H}_i - \{1\}$. □

We now distinguish the nodes visited before $slow_i$, the node $slow_i$ itself and the nodes visited after $slow_i$. By definition, $\forall h \in [1, slow_i)$, $f(h+1)$ is the first packet with a priority higher than or equal to this of m, processed in bp^{h+1} and coming from node h. Moreover, $f(h+1)$ is the last packet considered in bp^h. Hence, if we count packets processed in bp^h and bp^{h+1}, only $f(h+1)$ is counted twice. In the same way, $\forall h \in (slow_i, q]$, $f(h)$ is the first packet with a priority higher than or equal to this of m, processed in bp^h and coming from node $h-1$. Moreover, $f(h)$ is the last packet considered in bp^{h-1}. Thus, $f(h)$ is the only packet counted twice when counting packets processed in bp^{h-1} and bp^h. Hence, $X_{i,t}$ is equal to:

$$\sum_{h=1}^{slow_i-1}\left(\sum_{g=f(h)}^{f(h+1)-1} C^h_{\tau(g)} + C^h_{\tau(f(h+1))}\right) + \sum_{g=f(slow_i)}^{f(slow_i+1)} C^{slow_i}_{\tau(g)} + \sum_{h=slow_i+1}^{q}\left(\sum_{g=f(h)+1}^{f(h+1)} C^h_{\tau(g)} + C^h_{\tau(f(h))}\right).$$

Moreover, for any packet g visiting a node $h \in [1, q]$, $C^h_{\tau(g)} \leq C^{slow_{\tau(g),i}}_{\tau(g)}$. Then, as packets are numbered consecutively from $f(1)$ to $f(q+1) = m$, we get:

$$\sum_{h=1}^{slow_i-1}\left(\sum_{g=f(h)}^{f(h+1)-1} C^h_{\tau(g)}\right) + \sum_{g=f(slow_i)}^{f(slow_i+1)} C^{slow_i}_{\tau(g)} + \sum_{h=slow_i+1}^{q}\left(\sum_{g=f(h)+1}^{f(h+1)} C^h_{\tau(g)}\right) \leq \sum_{g=f(1)}^{m} C^{slow_i}_{\tau(g)}.$$

In addition, as in the worst case, $f(h+1)$ is a packet coming from node h, we have:

$$\sum_{h=1}^{slow_i-1} C^h_{\tau(f(h+1))} + \sum_{h=slow_i+1}^{q} C^h_{\tau(f(h))} \leq \sum_{\substack{h=1 \\ h \neq slow_i}}^{q} \max_{\substack{j \in hp_i \cup sp_i \cup \{i\} \\ first_{j,i} = first_{i,j}}} \{C^h_j\}.$$

Hence, $X_{i,t}$ is bounded by: $\sum_{g=f(1)}^{m} C^{slow_{\tau(g),i}}_{\tau(g)} - C^q_i + \sum^q_{\substack{h=1 \\ h \neq slow_i}} \max_{\substack{j \in hp_i \cup sp_i \cup \{i\} \\ first_{j,i} = first_{i,j}}} \{C^h_j\}$.

The term $X_{i,t}$ is maximized when the workload generated by such flows is the maximum. Then, we get:

Lemma 2 *Let m be the packet of flow τ_i generated at time t. When flows are scheduled FP/FIFO, the maximum delay incurred by m due to packets having a priority higher than or equal to this of m is bounded by:*

$$\sum_{j \in hp_i}\left(1 + \left\lfloor \frac{W^{last_{i,j}}_{i,t} - Smin^{last_{i,j}}_j + Smax^{first_{i,j}}_j - M^{first_{i,j}}_i + J_j}{T_j} \right\rfloor\right)^+ \cdot C^{slow_{j,i}}_j$$

$$+ \sum_{j \in sp_i \cup \{i\}} \left(1 + \left\lfloor \frac{t + Smax_i^{first_{j,i}} - Smin_j^{first_{j,i}} + Smax_j^{first_{i,j}} - M_i^{first_{i,j}} + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow_{j,i}}$$

$$+ \sum_{\substack{h \in \mathcal{H}_i \\ h \neq slow_i}} \max_{\substack{j \in hp_i \cup sp_i \cup \{i\} \\ first_{j,i} = first_{i,j}}} \{C_j^h\} - C_i^{last_i}.$$

Proof: Considering a packet m of τ_i generated at time t:

- Packets of flow τ_j, $j \in hp_i$, can delay m if they are generated at the earliest at time $a_{f(first_{i,j})}^{first_{i,j}} - Smax_j^{first_{i,j}} - J_j$ and at the latest at time $W_{i,t}^{last_{i,j}} - Smin_j^{last_{i,j}}$;
- Packets of flow τ_j, $j \in sp_i$, can delay m if they are generated at the earliest at time $a_{f(first_{i,j})}^{first_{i,j}} - Smax_j^{first_{i,j}} - J_j$ and at the latest at time $a_m^{first_{j,i}} - Smin_j^{first_{j,i}}$;
- Packets of τ_i can delay m if they are generated at the earliest at time $-J_i$ and at the latest at time t.

The maximum workload generated by any flow τ_j in the interval $[a, b]$ on node h is equal to $(1 + \lfloor (b - a)/T_j \rfloor)^+ \cdot C_j^h$. As $a_m^{first_{j,i}} \leq t + Smax_i^{first_{j,i}}$, $a_{f(first_{i,j})}^{first_{i,j}} \geq M_i^{first_{i,j}}$ and $Smin_i^{first_i} = Smax_i^{first_i} = 0$, we get the lemma. □

3.5 Delay due to non-preemption

We recall that packet scheduling is non-preemptive. Hence, despite the high priority of any packet m of any flow τ_i, a packet with a lower priority can delay m processing due to non-preemption. Indeed, if m arrives on node h while a packet m' belonging to lp_i is being processed, m has to wait until m' completion. By definition of FIFO scheduling, m cannot be delayed by a packet belonging to sp_i due to the non-preemption. It is important to notice that the non-preemptive effect is not limited to this waiting time. The delay incurred by packet m on node h directly due to m' may lead to consider packets belonging to hp_i, arriving after m on the node but before m starts its execution.

Property 2 *Let τ_i, $i \in [1, n]$, be a flow following path $\mathcal{H}_i = [first_i, ..., last_i]$. When flows are scheduled FP/FIFO, the maximum delay incurred by a packet of flow τ_i directly due to flows belonging to lp_i, denoted δ_i, is bounded by $\sum_{h \in \mathcal{H}_i} (\max_{j \in lp_i} \{C_j^h\} - 1)^+$, where $\max_{j \in lp_i} \{C_j^h\} = 0$ if $lp_i = \emptyset$.*

Proof: On each node h visited by τ_i, the delay incurred by m due to a packet m' of flow τ_j having a lower priority is maximum when (i) m' starts its processing on node h one time unit before the beginning of the busy period considered in the decomposition illustrated by Figure 2 and (ii) τ_j has the maximum processing time among flows belonging to lp_i. □

3.6 Latest starting time expression

From subsections 3.4 and 3.5, we can express the latest starting time of packet m on its last visited node.

Property 3 *Let m be the packet of flow τ_i generated at time t. When flows are scheduled FP/FIFO, the latest starting time of m on its last node visited, $W_{i,t}^{last_i}$, is bounded by:*

$$\sum_{j\in hp_i}\left(1+\left\lfloor\frac{W_{i,t}^{last_{i,j}}-Smin_j^{last_{i,j}}+Smax_j^{first_{i,j}}-M_i^{first_{i,j}}+J_j}{T_j}\right\rfloor\right)^+\cdot C_j^{slow_{j,i}}$$
$$+\sum_{j\in sp_i\cup\{i\}}\left(1+\left\lfloor\frac{t+Smax_i^{first_{j,i}}-Smin_j^{first_{j,i}}+Smax_j^{first_{i,j}}-M_i^{first_{i,j}}+J_j}{T_j}\right\rfloor\right)^+\cdot C_j^{slow_{j,i}}$$
$$+\sum_{\substack{h\in\mathcal{H}_i\\ h\neq slow_i}}\max_{\substack{j\in hp_i\cup sp_i\cup\{i\}\\ first_{j,i}=first_{i,j}}}\left\{C_j^h\right\}-C_i^{last_i}+\delta_i+(|\mathcal{H}_i|-1)\cdot Lmax.$$

Proof: By Lemma 2 and Property 2. □

The expression of $W_{i,t}^{last_i}$ is recursive. Let us consider the following series for any node $h\in\mathcal{H}_i$:

$$\begin{cases}
\mathcal{W}_{i,t}^{h\,(0)}=\sum_{j\in hp_i\cup sp_i}C_j^{slow_{j,i}^h}+\left(1+\left\lfloor\frac{t+J_i}{T_i}\right\rfloor\right)\cdot C_i^{slow_i^h}+\sum_{\substack{k\in\mathcal{H}_i^h\\ k\neq slow_i^h}}\max_{\substack{j\in hp_i\cup sp_i\cup\{i\}\\ first_{j,i}^h=first_{i,j}^h}}\left\{C_j^k\right\}-C_i^h+\delta_i^h+(|\mathcal{H}_i^h|-1)\cdot Lmax\\[2ex]
\mathcal{W}_{i,t}^{h\,(p+1)}=\sum_{\substack{j\in hp_i\\ last_{i,j}^h=h}}\left(1+\left\lfloor\frac{\mathcal{W}_{i,t}^{h\,(p)}-Smin_j^h+Smax_j^{first_{i,j}^h}-M_i^{first_{i,j}^h}+J_j}{T_j}\right\rfloor\right)^+\cdot C_j^{slow_{j,i}^h}\\
\quad+\sum_{\substack{j\in hp_i\\ last_{i,j}^h\neq h}}\left(1+\left\lfloor\frac{\mathcal{W}_{i,t}^{last_{i,j}^h}-Smin_j^{last_{i,j}^h}+Smax_j^{first_{i,j}^h}-M_i^{first_{i,j}^h}+J_j}{T_j}\right\rfloor\right)^+\cdot C_j^{slow_{j,i}^h}\\
\quad+\sum_{j\in sp_i\cup\{i\}}\left(1+\left\lfloor\frac{t+Smax_i^{first_{j,i}^h}-Smin_j^{first_{j,i}^h}+Smax_j^{first_{i,j}^h}-M_i^{first_{i,j}^h}+J_j}{T_j}\right\rfloor\right)^+\cdot C_j^{slow_{j,i}^h}\\
\quad+\sum_{\substack{k\in\mathcal{H}_i^h\\ k\neq slow_i^h}}\max_{\substack{j\in hp_i\cup sp_i\cup\{i\}\\ first_{j,i}^h=first_{i,j}^h}}\left\{C_j^k\right\}-C_i^h+\delta_i^h+(|\mathcal{H}_i^h|-1)\cdot Lmax
\end{cases}$$

with:

- $\mathcal{H}_i^h=[first_i,h]\subseteq\mathcal{H}_i$;
- $slow_i^h$, the slowest node visited by τ_i on $\mathcal{H}_i^h$;
- $first_{j,i}^h$, the first node visited by τ_j on $\mathcal{H}_i^h$;
- $last_{j,i}^h$, the last node visited by τ_j on $\mathcal{H}_i^h$;
- $slow_{j,i}^h$, the slowest node visited by τ_j on $\mathcal{H}_i^h$;
- δ_i^h, the maximum delay incurred by a packet of τ_i directly due to non-preemption when visiting $\mathcal{H}_i^h$.

When the series $\mathcal{W}_{i,t}^{last_i}$ converges, $W_{i,t}^{last_i}$ is its limit.

3.7 Worst case end-to-end response time

The worst case end-to-end response time of the packet of flow τ_i generated at time t is equal to: $W_{i,t}^{last_i} + C_i^{last_i} - t$. The worst case end-to-end response time of flow τ_i is then equal to: $R_i = \max_{t \geq -J_i}\{W_{i,t}^{last_i} + C_i^{last_i} - t\}$. In order not to test all times $t \geq -J_i$, we establish Lemma 3.

Lemma 3 *Let us consider a flow τ_i following a path $\mathcal{H}_i$. If flows are scheduled FP/FIFO, then for any time $t \geq -J_i$: $W_{i,t+\mathcal{B}_i^{slow}}^{last_i} \leq W_{i,t}^{last_i} + \mathcal{B}_i^{slow}$, with: $\mathcal{B}_i^{slow} = \sum_{j \in hp_i \cup sp_i \cup \{i\}} \lceil \mathcal{B}_i^{slow}/T_j \rceil \cdot C_j^{slow_{j,i}}$.*

Proof: In [9]. □

From the worst case analysis given in this section and the previous lemma, we get the following property.

Property 4 *When flows are scheduled FP/FIFO, the worst case end-to-end response time of any flow τ_i is bounded by $R_i = \max_{-J_i \leq t < -J_i + \mathcal{B}_i^{slow}}\{W_{i,t}^{last_i} + C_i^{last_i} - t\}$, with:*

$$W_{i,t}^{last_i} = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_{i,t}^{last_{i,j}} - Smin_j^{last_{i,j}} + Smax_j^{first_{i,j}} - M_i^{first_{i,j}} + J_j}{T_j} \right\rfloor \right)^+ \cdot C_j^{slow_{j,i}}$$

$$+ \sum_{j \in sp_i \cup \{i\}} \left(1 + \left\lfloor \frac{t + Smax_i^{first_{j,i}} - Smin_j^{first_{j,i}} + Smax_j^{first_{i,j}} - M_i^{first_{i,j}} + J_j}{T_j} \right\rfloor \right)^+ \cdot C_j^{slow_{j,i}}$$

$$+ \sum_{\substack{h \in \mathcal{H}_i \\ h \neq slow_i}} \max_{\substack{j \in hp_i \cup sp_i \cup \{i\} \\ first_{j,i} = first_{i,j}}} \{C_j^h\} - C_i^{last_i} + \sum_{h \in \mathcal{H}_i} \left(\max_{j \in lp_i}\{C_j^h\} - 1 \right)^+ + (|\mathcal{H}_i| - 1) \cdot Lmax,$$

and $\mathcal{B}_i^{slow} = \sum_{j \in hp_i \cup sp_i \cup \{i\}} \left\lceil \frac{\mathcal{B}_i^{slow}}{T_j} \right\rceil \cdot C_j^{slow_{j,i}}$.

Proof: By Property 3 and Lemma 3. □

3.8 Computation algorithm

To compute the worst case response times of a flow set, we proceed by decreasing fixed priority order. We first compute the response times of flows having the highest fixed priority. We then continue with flows having the highest priority among those whose response time is not yet computed and so on. Let F_i be the highest priority of flows whose response time has not yet been computed. Let τ_i, $i \in [1, n]$, be a flow of priority F_i. We compute the set $\mathcal{S}_i$ of flows crossing directly or indirectly τ_i and apply Property 4 to compute the worst case response time of τ_i. More formally, we determine $\mathcal{S}_i$ as follows:

- $\mathcal{S}_i = \{\tau_i\}$;
- $\mathcal{S}_i = \mathcal{S}_i \cup \{\tau_j, j \in hp_i \cup sp_i, \tau_j$ crosses directly $\tau_i\}$;
- $\mathcal{S}_i = \mathcal{S}_i \cup \{\tau_k, k \in hp_i \cup sp_i, \exists \tau_j \in \mathcal{S}_i$ such that τ_k crosses directly $\tau_j\}$.

Notice that if a flow exceeds its deadline, we stop the computation. We proceed in the same way for any flow having priority F_i.

4 Probabilistic approach for computing the probabilities of meeting the deadlines

The probabilistic approach will be able to guarantee to each flow τ_i belonging to the set $\mathbb{D}$ that the end-to-end response time of any of its packets does not exceed the deadline D_i with a probability higher than P_i. To obtain this probability, the end-to-end response time distribution must be computed.

4.1 Notations

We focus on the set $\{\tau_1, \tau_2, ..., \tau_n\}$ of n flows. We consider in this section that these flows are characterized by Poisson arrivals (see Property 1) and adopt (or recall) the following notations:

- τ_i, a sporadic flow of the set $\{\tau_1, ..., \tau_n\}$;
- λ_i, the average arrival rate of a packet of flow τ_i;
- μ_i^h, the average processing time of a packet of flow τ_i in node h;
- $P_{success}(D_i)$, the probability that flow τ_i will not miss its deadline;
- $\mathcal{H}_i = [first_i, ..., last_i]$, the path followed by flow τ_i;
- $|\mathcal{H}_i|$, the number of nodes visited by flow τ_i;
- $|F|$, the number of fixed priorities shared by the flows considered;
- $pre_i(h)$, the node visited by τ_i just before node h;
- $suc_i(h)$, the node visited by τ_i just after node h;
- ρ_i^{ab}, the utilization factor of the link server ab by the packets of flow τ_i, which is equal to $\lambda_i^{ab}/\mu_i^{ab}$;
- ρ^{ab}, the utilization factor of the link server ab, which is equal to $\sum_{j=1}^{n} \rho_j^{ab}$.

4.2 Node response time distribution

To compute the end-to-end response time distribution of any flow τ_i, we first focus on its 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 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 instantaneous, the node response time, for any packet of τ_i going through node a to node b, corresponds to the response time of the queuing system modelling the link ab [10]. To simplify this study, we introduce Assumption 9.

Assumption 9 *Packet arrivals of any flow τ_i to a link ab is also a Poisson process with the parameter λ_i^{ab}, that is equal to: λ_i if ab belongs to the path of τ_i, 0 otherwise.*

According to the traffic description, Assumption 9 and the FP/FIFO scheduling, each link can be modelled by an M/G/1 station with n classes of customers, the non-preemptive Priority Queuing (with $|F|$ priorities) and the Head Of Line (HOL) discipline [11]. The average arrival rate of packets of class i (flow τ_i) on the link ab is λ_i^{ab} and the average service rate of packets of τ_i on the link ab is μ_i^{ab}. The node response time distribution for packets of flow τ_i at the link ab is obtained by inspecting its Laplace transform which will be denoted by $\left(S_i^{ab}\right)^*(s)$ and is given by: $\left(S_i^{ab}\right)^*(s) = \left(W_{F_i}^{ab}\right)^*(s) \cdot \left(B_i^{ab}\right)^*(s)$, where $\left(W_{F_i}^{ab}\right)^*(s)$ is the Laplace transform of the waiting time density of packets having the priority F_i at the link ab and $\left(B_i^{ab}\right)^*(s)$ is the Laplace transform of the service time probability density function of packets of flow τ_i at the link ab. Indeed, packets having the same priority have the same waiting time distribution. We first focus on the computation of $\left(W_{F_i}^{ab}\right)^*(s)$. A packet having the priority F_i must wait for [12]:

- packets with a priority $\geq F_i$ and found in the queue upon the arrival of our tagged packet;
- packets with a priority $> F_i$ and which arrive before the service beginning of our tagged packet;
- the packet found in service upon the arrival of our tagged packet.

As in [13], we define two categories of packets: those belonging to flows in $hp_i \cup sp_i \cup \{i\}$ (called priority packets) and those belonging to flows in lp_i (called ordinary packets). The Poisson arrival rates of these two packets categories are given by: $\lambda_{F_i}^+ = \sum_{j \in hp_i \cup sp_i \cup \{i\}} \lambda_j^{ab}$ and $\lambda_{F_i}^- = \sum_{j \in lp_i} \lambda_j^{ab}$.

The Laplace transforms of the service time densities of priority and ordinary packets are respectively: $\left(B_{F_i}^+\right)^*(s) = \sum_{j \in hp_i \cup sp_i \cup \{i\}} \frac{\lambda_j^{ab}}{\lambda_{F_i}^+} \cdot \left(B_j^{ab}\right)^*(s)$ and $\left(B_{F_i}^-\right)^*(s) = \sum_{j \in lp_i} \frac{\lambda_j^{ab}}{\lambda_{F_i}^-} \cdot \left(B_j^{ab}\right)^*(s)$.

Notice that the waiting time of a packet having priority F_i is invariant to the change in the order of service. This waiting time can be computed as follows [14]:

- the service time of the packet in service upon the arrival of our tagged packet and the packets having priority $\geq F_i$ and waiting in the system at this time. This time will be denoted by $W_{F_i}^+$;
- the service time of packets having priority $> F_i$ that arrive during $W_{F_i}^+$ and the duration of all busy periods generated by these packets.

Let $\left(W_{F_i}^+\right)^*(s)$ be the Laplace transform of the waiting time density of priority packets. $\left(W_{F_i}^+\right)^*(s)$ is given by [15]:

$$\left(W_{F_i}^+\right)^*(s) = \frac{\left(1-\rho^{ab}\right) \cdot s + \lambda_{F_i}^- \left(1 - \left(B_{F_i}^-\right)^*(s)\right)}{s - \lambda_{F_i}^+ + \lambda_{F_i}^+ \left(B_{F_i}^+\right)^*(s)}.$$

By coming back to the original system, the waiting time of packets having the priority F_i at the link ab corresponds to $W_{F_i}^+$ and the sum of the service times of packets having priorities higher than F_i that arrive during the busy period initiated by $W_{F_i}^+$. Hence, the Laplace transform of this waiting time density $\left(W_{F_i}^{ab}\right)^*(s)$ is given by:

$$\left(W_{F_i}^{ab}\right)^*(s) = \left(W_{F_i}^+\right)^* \left(s + \lambda_{F_i+1}^+ - \lambda_{F_i+1}^+ \left(\theta_{F_i+1}^+\right)^*(s)\right),$$

where $\left(\theta_{F_i+1}^+\right)^*(s)$ corresponds to the Laplace transform of a busy period duration density generated by packets having a priority strictly greater than F_i and is the solution to the equation:

$$\left(\theta^{+}_{F_i+1}\right)^{*}(s) = \left(B^{+}_{F_i+1}\right)^{*}\Big(s + \lambda^{+}_{F_i+1} - \lambda^{+}_{F_i+1}\left(\theta^{+}_{F_i+1}\right)^{*}(s)\Big).$$

$\left(\theta^{+}_{F_i+1}\right)^{*}(s)$ can be obtained numerically [15]:

$$\left(\theta^{+}_{F_i+1}\right)^{*}(s) = \int_0^s \sum_{n=1}^{\infty} e^{-\lambda^{+}_{F_i+1}x}\frac{\left(\lambda^{+}_{F_i+1}x\right)^{n-1}}{n!} b^{+}_{F_i+1(n)}(x)\,dx,$$

where $b^{+}_{F_i+1(n)}(x)$ is the n-fold convolution of $b^{+}_{F_i+1}(x)$ with itself and represents the probability density function for the sum of n independent random variables, where each corresponds to the service time of a packet belonging to hp_i. Let $\sigma_{F_i+1} = s + \lambda^{+}_{F_i+1} - \lambda^{+}_{F_i+1}\left(\theta^{+}_{F_i+1}\right)^{*}(s)$, then we get:

$$\begin{aligned}\left(W^{ab}_{F_i}\right)^{*}(s) &= \frac{\left(1-\rho^{ab}\right)\sigma_{F_i+1}+\lambda^{-}_{F_i}\left(1-\left(B^{-}_{F_i}\right)^{*}(\sigma_{F_i+1})\right)}{s+\lambda^{+}_{F_i+1}-\lambda^{+}_{F_i+1}\left(B^{+}_{F_i+1}\right)^{*}(\sigma_{F_i+1})-\lambda^{+}_{F_i}+\lambda^{+}_{F_i}\left(B^{+}_{F_i}\right)^{*}(\sigma_{F_i+1})}\\ &= \frac{\left(1-\rho^{ab}\right)\sigma_{F_i+1}+\sum_{j\in lp_i}\lambda^{ab}_j\left(1-\left(B^{ab}_j\right)^{*}(\sigma_{F_i+1})\right)}{s+\sum_{j\in hp_i}\lambda^{ab}_j+\sum_{j\in sp_i\cup\{i\}}\lambda^{ab}_j\left(B^{ab}_j\right)^{*}(\sigma_{F_i+1})}.\end{aligned}$$

The Laplace transform of the node response time distribution for packets of flow τ_i at the link ab, $\left(S^{ab}_i\right)^{*}(s)$, is then equal to: $\left(B^{ab}_i\right)^{*}(s)\cdot\frac{\left(1-\rho^{ab}\right)\sigma_{F_i+1}+\sum_{j\in lp_i}\lambda^{ab}_j\left(1-\left(B^{ab}_j\right)^{*}(\sigma_{F_i+1})\right)}{s+\sum_{j\in hp_i}\lambda^{ab}_j+\sum_{j\in sp_i\cup\{i\}}\lambda^{ab}_j\left(B^{ab}_j\right)^{*}(\sigma_{F_i+1})}$.

4.3 End-to-end response time distribution

Let $\widetilde{s_i}$ be the random variable representing the end-to-end response time for a packet of flow τ_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 to the egress node. The random variable $\widetilde{s_i}$ corresponds to the sum of the response times on the nodes crossed by the packet while going through the network and the sum of the transmission delays between the different nodes: $\widetilde{s_i} = \sum_{h=first_i}^{pre_i(last_i)}(\widetilde{s}_i^{(h,suc_i(h))} + \widetilde{d})$, where $\widetilde{d}$ is the random variable corresponding to the transmission delays between two nodes. The random variables corresponding to these different durations being independent, we obtain:

$$S^{*}_i(s) = \left(L^{*}(s)\right)^{(|\mathcal{H}_i|-1)}\prod_{h=first_i}^{pre_i(last_i)}\left(\left(\widehat{S}_i^{(h,suc_i(h))}\right)^{*}(s)\right),$$

where $\left(\widehat{S}^{ab}_i\right)^{*}(s) = \begin{cases} 1 & \text{if link } ab \text{ belongs to } \mathcal{H}_i \\ \left(S^{ab}_{rk}\right)^{*}(s) & \text{otherwise}\end{cases}$

and $L^{*}(s)$ is the Laplace transform of $\widetilde{d}$ probability density function. According to Assumption 4, we have: $L^{*}(s) = \frac{e^{-sL_{\max}}-e^{-sL_{\max}}}{s(L_{\max}-L_{\min})}$. The end-to-end response time distribution is obtained by inspecting its Laplace transform.

4.4 Probabilistic QoS guarantee

The end-to-end response time distribution enables us to determine, for a given configuration, the probability that a flow packet does not stay in the network beyond a given duration.

Property 5 *A packet belonging to the flow τ_i with a relative deadline D_i meets its deadline with the probability: $P_{success}(D_i) = P[\widetilde{s_i} < D_i] = \int_0^{D_i} s_i(t)dt$, where $s_i(t)$ is the end-to-end response time distribution obtained by inspecting its Laplace transform.*

The study developed here allows to provide probabilistic QoS guarantees to flows having quantitative constraints on their end-to-end response times.

5 Admission control

Now we will show how to manage the coexistence of deterministic and probabilistic QoS guarantees in a network. This is done by an admission control, derived from the results established in the previous sections. We will see a numerical example in the next section.

The admission control is in charge of deciding whether a new flow τ_k can be accepted in the network. This decision is based on the following conditions:

1. the acceptance of τ_k should not compromise the guarantees granted to the already accepted flows. This condition requires that both flows belonging to $\mathbb{D}$ and flows belonging to $\mathbb{P}$ meet their deadlines with the requested probability (this probability is 1 for flows in $\mathbb{D}$). For this purpose, the admission control proceeds as follows:

   - for each flow τ_j belonging to $\mathbb{D}$, we recompute its end-to-end response time R_j and check that $R_j \leq D_j$, by applying Property 4;
   - for each flow τ_j belonging to $\mathbb{P}$, we recompute its end-to-end response time distribution and check that $P_{success}(D_j) \geq P_j$, by applying Property 5. We recall that for this computation, packet arrivals of all flows in $\mathbb{D} \cup \mathbb{P}$ are upper bounded by Poisson arrivals.

2. the guarantee requested by τ_k can be met taking into account the available resources. Depending on the type of the QoS guarantee required by τ_k, we apply Property 4 or Property 5 to check either that $R_k \leq D_k$ or $P_{success}(D_k) \geq P_k$.

Remark: The impact of a new flow τ_k on flows belonging to $\mathbb{D}$ with a priority strictly higher than this of τ_k is due to the non-preemptive effect. In order to avoid the computation of the worst case end-to-end response time of any flow $\tau_j \in \mathbb{D}$ such that $F_j > F_k$, we will maximize the processing time of any potential flow that could be accepted with a priority lower than that of τ_j by C_{max,F_j}. With no particular knowledge on the paths followed by the potential flow, we can assume that in the worst case, τ_j can be delayed by $C_{max,F_j} - 1$ on each node visited, because of a potential flow with a fixed priority striclty lower than F_j. Then, Property 2 becomes Property 6.

Property 6 *Let τ_i, $i \in [1, n]$, be a flow following path $\mathcal{H}_i = [first_i, ..., last_i]$. When flows are scheduled FP/FIFO, the maximum delay incurred by a packet of flow τ_i directly due to flows belonging to lp_i, denoted δ_i, is bounded by: $\sum_{h=first_i}^{last_i}(C_{max,F_i} - 1)$, where C_{max,F_i} is the maximum processing time of any possible flow with a priority lower than F_i and $C_{max,F_i} - 1 = 0$ if F_i is the smallest possible fixed priority.*

Thanks to this property, the acceptance of a new flow τ_k does not impact flows having a priority higher than that of τ_k.

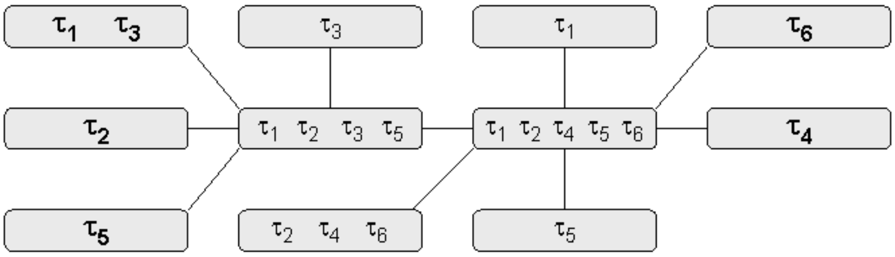

Fig. 3: Paths followed by the flows considered

6 Model validation

In this section, we focus on the validation of our analytical model and the mathematical study presented in the previous sections. More precisely, we show, in an example, that the theoretical results are (i) very close to those obtained by simulation with NS2 for the probabilistic model and (ii) reached in a given worst case scenario for the deterministic study. In the next section, we consider a more general example.

6.1 Example

Let us consider six flows: two control and command flows (τ_1 and τ_2), two video flows (τ_3 and τ_4), and two flows corresponding to file transfer (τ_5 and τ_6). The characteristics of the flows considered are given in Table 1 and their paths are illustrated in Figure 3, where the node identifiers have been omitted for clarity reason. Moreover, for each flow, we have emphasized its name in its input node.

Tab. 1: Characteristics of flows considered.

flow	C_i (μs)	T_i (μs)	$\overline{C_i}$ (μs)	D_i (μs)	F_i	P_i (%)
τ_1	100	8000	51.2	7000	3	100
τ_2	100	8000	51.2	7000	3	100
τ_3	2240	4870	1168	10000	2	99
τ_4	100	340	51.2	1000	2	90
τ_5	560	4610	460	7000	1	95
τ_6	560	4610	460	5000	1	95

Notice that only flows τ_1 and τ_2 require deterministic QoS guarantees. Flow τ_3, for instance, requires its end-to-end response time to be lower than 10 milliseconds with a probability higher than or equal to 99%. Moreover, links are 10 Mbit/s.

Table 2 presents the results (the worst case end-to-end response time R_i and the deadline success probability $P_{success}(Di)$) obtained by the analytical study and simulation.

First, we can see for the probabilistic guarantees that the deadline success probabilities obtained by simulation are very close to the results obtained with the mathematical study. As a consequence, the mathe-

Tab. 2: Analytical study versus simulation

			analytical study		*simulation*	
$flow$	D_i (μs)	P_i (%)	R_i (μs)	$P_{success}(Di)$	R_i (μs)	$P_{success}(Di)$
τ_1	7000	100	5537		2255	
τ_2	7000	100	3758		2221	
τ_3	10000	99		0.9933		0.9967 ± 0.005
τ_4	1000	90		0.9854		0.9934 ± 0.003
τ_5	7000	95		0.9999		0.9999 ± 0.00001
τ_6	5000	95		0.9999		0.9998 ± 0.00003

matical model is validated in this example.

On the other hand, for the deterministic guarantees, the worst case end-to-end response times obtained by simulation for the flows τ_1 and τ_2 are smaller than those obtained with the mathematical study. This can be explained by the fact that no packet of τ_1 nor τ_2 has gone through the worst case scenario in the simulation duration. However, we can show that the worst case end-to-end response time can be reached. For example, this of τ_1 given in Table 2 is reached in the worst case scenario, illustrated in Figure 4. Indeed, if we number nodes visited by τ_1 from 1 to 4, the following scenario leads to a worst case end-to-end response time equal to 5537:

- On node 1, flow τ_3 generates a packet at time 0 and flow τ_1 generates a packet at time 1. Hence, the packet of τ_1 is delayed by the packet of τ_3.
- On node 2, packets of τ_3 and τ_1 arrive respectively at times 2240 and 2340. As flow τ_5 follows an independent path until node 2, a packet of this flow can arrive at any time. Then, we assume that a packet of τ_5 arrives at time 1779. Therefore, the packet of τ_3 starts its execution at time 2339, that is one time unit before the arrival of the packet of τ_1. The non-preemptive effect is then maximized for this packet. Moreover, if a packet of τ_2 arrives during the processing of the packet of τ_3, it is processed before the packet of τ_1.
- On node 3, packets of flow τ_2 and τ_3 arrive respectively at times 4679 and 4779. If a packet of τ_6 arrives at time 4678 (this is possible for the same reasons as τ_5 on node 2), then packets of flows τ_2 and τ_1 experiment a maximum non-preemptive effect equal to $560 - 1$ time units.
- On node 4, the packet of flow τ_1 arrives at time 5438. Hence, as flow τ_1 is the only flow visiting this node, its worst case end-to-end response time is equal to 5438 plus 100 (its maximum processing time on this node) minus 1 (its generation time), that is 5537.

We can therefore conclude that both deterministic and probabilistic studies are validated in this example.

6.2 Coexistence benefits

To illustrate the interest of the coexistence of deterministic and probabilistic QoS guarantees, we first show, by computing the worst case end-to-end response times, that this network fails to provide determin-

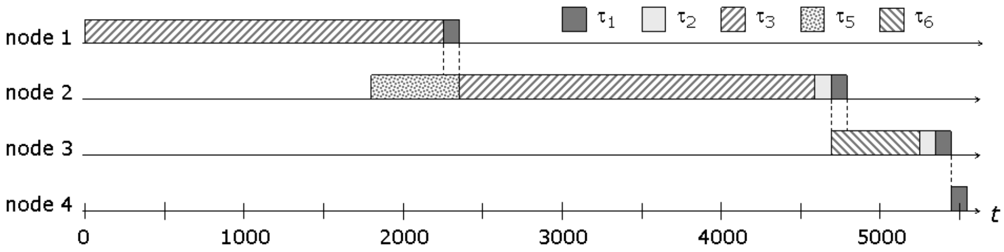

Fig. 4: Worst case scenario for τ_1: the upper bound is reached

istic QoS guarantees to all flows. These response times are obtained by applying Property 4 and are given in Figure 5. We notice that flows τ_4 and τ_5 have missed their deadlines. Hence, with a pure deterministic approach, only four flows (τ_1, τ_2, τ_3 and τ_6) would be accepted.

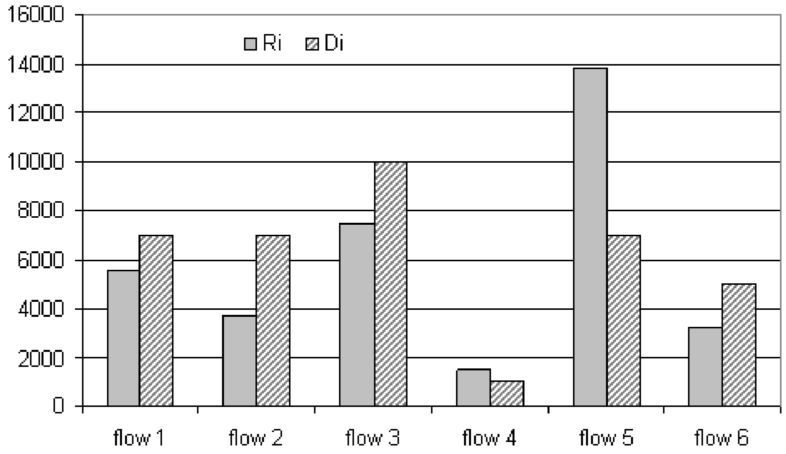

Fig. 5: Pure deterministic approach

Nevertheless, flows τ_1 and τ_2 do not tolerate any deadline violation. The probabilistic approach fails to provide such a guarantee. Indeed, the probability $P_{success}$ cannot in any case equal 1, as a Poisson arrival process and a service time exponentially distributed have been considered. To accept all the flows considered and meet their QoS requirements, we provide deterministic QoS guarantees for flows τ_1 and τ_2 and probabilistic for the others. The success probabilities given in Table 3 are computed by applying Property 5.

Providing both deterministic and probabilistic guarantees enables a better resource utilization rate. Indeed, if τ_4, for instance, required a deterministic QoS guarantee, it would be rejected as the computed bound on its end-to-end response time is equal to $R_4 = 1519\ \mu s > D_3 = 1000\ \mu s$. By asking a probabilistic QoS guarantee, τ_3 is accepted in the network with the probability of 98.54% to meet its deadline.

Tab. 3: Coexistence of deterministic and probabilistic QoS garantees

flow	D_i (μs)	P_i $(\%)$	R_i (μs)	$P_{success}(D_i)$ $(\%)$
τ_1	7000	100	5537	
τ_2	7000	100	3758	
τ_3	10000	99		99.336
τ_4	1000	90		98.540
τ_5	7000	95		99.999
τ_6	5000	95		99.999

Consequently, this example shows that the coexistence of deterministic and probabilistic QoS guarantees enables us to accept more flows than a pure deterministic approach does.

7 Extended example

In this section, we present an example illustrating the benefits brought by the coexistence of deterministic and probabilistic QoS guarantees in a network consisting of 48 nodes. Let us consider 24 flows, presented in Table 4. We suppose that links are 10 Mbits/s and the paths of flows considered are illustrated in Figure 6. As in the previous section, the node identifiers have been omitted for clarity reason. Moreover, for each flow, we have emphasized its name in its input node.

Tab. 4: Characteristics of flows

$flow(s)$	C_i (μs)	T_i (μs)	$\overline{C_i}$ (μs)	D_i (μs)	F_i	P_i $(\%)$
$\tau_{(1,3,7,9,17,19,23)}$	100	8000	51.2	13000	3	100
$\tau_{(12,24)}$	100	8000	51.2	7000	3	97
$\tau_{(2,8,10,22)}$	2240	4870	1168	10000	2	85
$\tau_{(4,14,18)}$	100	340	51.2	1000	2	90
$\tau_{(5,13,16,21)}$	560	4610	460	7000	1	95
$\tau_{(6,11,15,20)}$	560	4610	460	5000	1	80

We present in Figure 7 the worst end-to-end response times of flows requiring deterministic QoS guarantees (i.e., τ_1, τ_3, τ_7, τ_9, τ_{17}, τ_{19} and τ_{23}). Notice that the deterministic bounds are computed according to Property 4. As we can see, each flow meets its end-to-end deadline. Moreover, flows τ_{12} and τ_{24} have the same characteristics, except the deadline and the type of QoS guarantee. If we compute the worst case end-to-end response times of these flows, we obtain $R_{12} = 9235$ and $R_{24} = 10215$. Thus, these flows do not meet their end-to-end deadlines. However, we see in Table 6 that the considered deadlines are met with a probability higher than 97%.
The deadline success probabilities of flows requiring probabilistic QoS guarantees are given in Table 5.

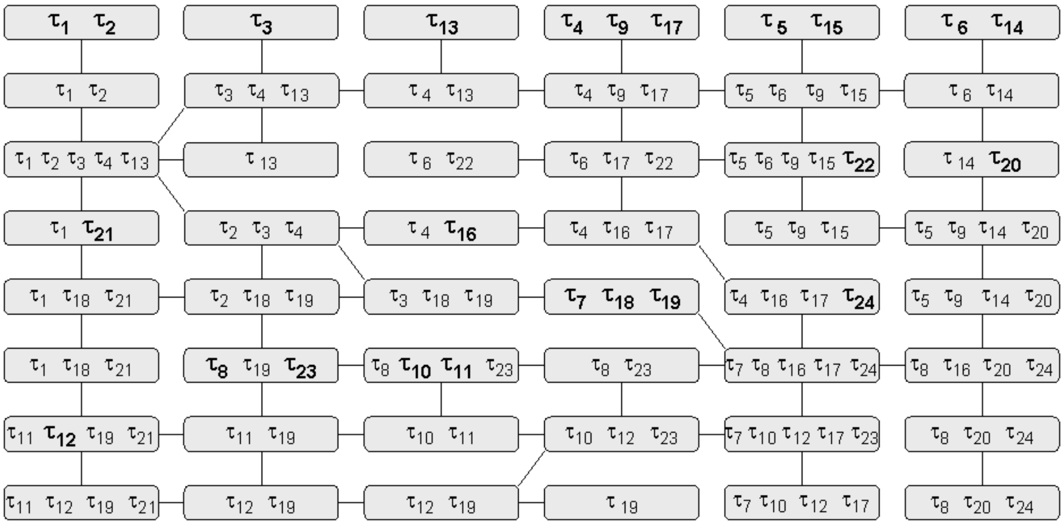

Fig. 6: Paths followed by the flows considered

These results have been obtained by simulation with NS2.
As in Section 6, these results highlight the benefits brought by our solution. Indeed, providing only deterministic bounds on the end-to-end response times would lead either to accept a small number of flows or to have a low resource utilization. Indeed, the worst case scenario occurs infrequently. Nevertheless, probabilistic guarantees are not satisfying when specific applications, such as control/commande applications, have strict end-to-end response time requirements. Notice that our solution makes no restriction about the importance degree of a flow and its type of guarantee. As a consequence, each flow receives the guarantee requested and globally the network achieves a higher resource utilization.

8 Perspectives

In this paper, we have established new results to provide quantitative QoS guarantees when flows are scheduled according to non-preemptive FP/FIFO. The coexistence of deterministic and probabilistic QoS guarantees allows a higher resource utilization. Consequently, more flows can be accepted in the network.

To further improve the resource utilization, we can investigate two directions. First, we can focus on techniques dropping packets as soon as it can be proved that these packets are unable to meet their deadlines. These techniques allow us to spare resources. However, they should be selected carefully because very aggressive techniques will discard a packet as soon as its local deadline is missed, even if this excessive delay can be compensated on other nodes such that the end-to-end deadline is finally met. The second direction can be the study of other scheduling strategies combining the flow's degree of importance and its end-to-end deadline. For instance, scheduling flows having the same fixed priority according to EDF (Earliest Deadline First) would lead to a better schedulability of a flow set by taking into account the end-to-end deadlines.

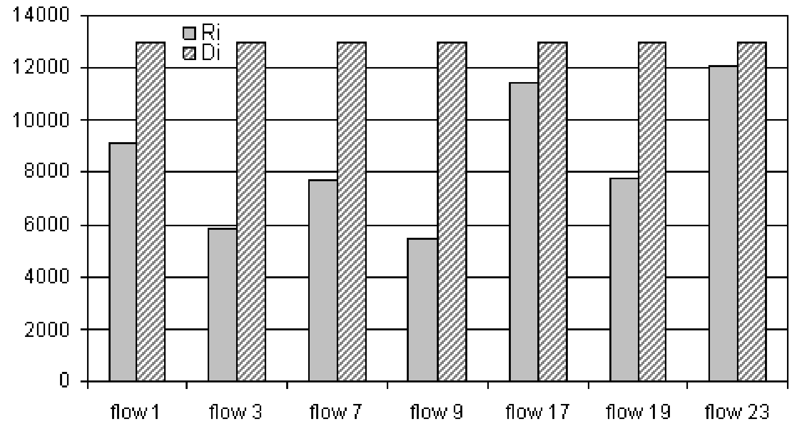

Fig. 7: Pure deterministic approach

Moreover, in a future work, we will see how to apply our results to networks using shaping techniques. Different cases will be considered: (i) shaping done only in the ingress nodes, (ii) shaping done in every node and (iii) shaping done in specific nodes.

Finally, flow aggregation techniques could be interesting to study.

9 Conclusion

FP scheduling is used when flows have different degrees of importance. FP/FIFO is the most commonly used implementation of FP: packets having the same fixed priority are scheduled according to their arrival order on the node considered. In this paper, we have shown how to provide quantitative QoS guarantees to flows having constraints in their end-to-end response times. We have proposed a solution to achieve this goal while preserving a high resource utilization rate. This solution allows the coexistence of two types of quantitative QoS guarantees: deterministic and probabilistic. No restriction is given concerning the relationship between the fixed priority value and the type of QoS guarantee that can be granted to a flow.

On the one hand, deterministic guarantees are obtained from a worst case end-to-end response time analysis, based on the trajectory approach. This ensures that the worst case end-to-end response time of the flow considered does not exceed the required deadline. On the other hand, probabilistic guarantees are obtained from a mathematical model, based on Poisson arrivals. The distribution of the end-to-end response time of the flow considered is computed. This ensures that the end-to-end response time of this flow does not exceed the given deadline with a probability higher than what is required.

Finally, we have shown how to derive an admission control from our results. This admission control, in charge of deciding the acceptance of a new flow, allows us to accept more flows, leading to a better resource utilization than a pure deterministic approach. Moreover, each flow receives the quantitative QoS guarantee in accordance with its requirements.

Tab. 5: Simulation results for flows with probabilistic QoS guarantees

$flow$	D_i (μs)	P_i $(\%)$	$P_{success}(Di)$ (μs)
τ_2	1000	85	0.9857 ± 0.0018
τ_4	1000	90	0.9135 ± 0.0034
τ_5	7000	95	0.9219 ± 0.0031
τ_6	5000	80	0.8372 ± 0.005
τ_8	10000	85	0.9443 ± 0.0032
τ_{10}	10000	85	0.9863 ± 0.0017
τ_{11}	5000	80	0.9644 ± 0.0019
τ_{12}	7000	97	0.9799 ± 0.0021
τ_{13}	7000	95	0.9999 ± 0.0001
τ_{14}	1000	90	0.9899 ± 0.0002
τ_{15}	5000	80	0.9946 ± 0.0008
τ_{16}	7000	95	0.9517 ± 0.0031
τ_{18}	1000	90	0.9697 ± 0.0010
τ_{20}	5000	95	0.8521 ± 0.0046
τ_{21}	7000	95	0.9990 ± 0.0004
τ_{22}	10000	85	0.9654 ± 0.0011
τ_{24}	7000	97	0.9737 ± 0.0026

References

[1] K. Tindell, A. Burns, A. J. Wellings, *Analysis of hard real-time communications*, Real-Time Systems, Vol. 9, pp. 147-171, 1995.

[2] J. Liu, *Analysis of hard real-time communications*, Real-Time Systems, Prentice Hall, 2000.

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

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

[5] K. Tindell, J. Clark, *Holistic schedulability analysis for distributed hard real-time systems*, Microprocessors and Microprogramming, Euromicro Jal, Vol. 40, 1994.

[6] M. Spuri, *Holistic analysis for deadline scheduled real-time distributed systems*, INRIA Research Report No 2873, April 1996.

[7] J. Y. le Boudec, P. Thiran, Network calculus: A theory of deterministic queuing systems for the Internet, *LNCS 2050, Springer-Verlag*, September 2003.

[8] S. Martin, *Maîtrise de la dimension temporelle de la qualité de service dans les réseaux*, Ph.D. Thesis, University of Paris 12, July 2004.

[9] S. Martin, P. Minet, *Worst case end-to-end response times of flows scheduled with FP/FIFO*, International Conference on Networking (ICN'06), Mauritius, April 2006.

[10] Y. Jiang, C. K. Tham and C. C. Ko, *A Probabilistic Priority Scheduling Discipline for High Speed Networks*, Proc. IEEE Workshop on High Performance Switching and Routing (HPSR 2001), May 2001.

[11] J. Walraevens, B. Steyaert, M. Moeneclaey and H. Bruneel, *Delay Analysis of a HOL Priority Queue*, Telecommunication Systems, pp 81-98, December 2005.

[12] C. Tham, Q. Yao and Y. Jiang, *Achieving differentiated services through multi-class probabilistic priority scheduling*, * Computer Networks, pp 577-593, 2002.

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

[14] I. Korbi, S. Azzaz and L. A.Saidane, *Probabilistic QoS Guarantees: Fixed and Dynamic Priority Scheduling*, IJCSNS, Vol.6 No.9B, September 2006.

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