FP/FIFO scheduling: deterministic versus probabilistic QoS guarantees and p-schedulability

Leila Azouz Saidane  
ENSI, Cristal  
2010, Tunis, Tunisia  
leila.saidane@ensi.rnu.tn

Skander Azzaz  
ENSI, Cristal  
2010, Tunis, Tunisia  
skander_azzaz@yahoo.fr

Steven Martin  
LRI, University of Paris-Sud  
91405 Orsay, France  
steven.martin@lri.fr

Pascale Minet  
INRIA Rocquencourt  
78153 Le Chesnay, France  
pascale.minet@inria.fr

Abstract—In this paper, we focus on applications with quantitative QoS (Quality of Service) requirements in their end-to-end response time. Two types of quantitative QoS guarantees can be delivered by a network: deterministic and probabilistic. 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 fixed priority, the packet that arrives first on the node considered is scheduled first. We compare deterministic and probabilistic QoS guarantees and introduce the concept of p-schedulability: the QoS requested by all flows is met with probability $p$.

I. INTRODUCTION

To achieve quantitative QoS (Quality of Service) guarantees to various applications, two approaches can be used:

• a deterministic approach, 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. This approach is based on a worst case analysis and can lead to a low resource utilization.

• a probabilistic approach, 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. However, providing only probabilistic guarantees with a probability less than 1 is not acceptable for applications requiring hard deadlines.

In this paper, we compare deterministic and probabilistic approaches and introduce the concept of p-schedulability based on the probabilistic approach.

We focus on the non-preemptive Fixed Priority (FP) scheduling [1], [2] that exhibits interesting properties. Indeed, it favors flows having the highest fixed priority (the fixed priority of a flow can be easily assigned to reflect the importance degree of this flow), the impact of a new flow is limited to flows having smaller fixed priorities and it is easy to implement.

However, in many cases, several flows may have to share the same priority, when, for example, the number of fixed priorities available on a processor is less than the flow number or when 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. 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. This scheduling enables to improve the worst case response times of flows. Indeed, a packet cannot be delayed by other packets of the same priority released after it.

The paper is organized as follows. We first present our assumptions and define our notations. In Section III, we show how to conduct a worst case analysis to provide deterministic end-to-end response times to the flows considered. In Section IV, we present a mathematical model to obtain, for any flow, the probability that its response time does not exceed a given value. The concept of p-schedulability is defined. Moreover, this model is validated by means of simulations performed with NS2. An example to illustrate these results is presented in Section V, where we compare the two types of QoS guarantees obtained by the deterministic and probabilistic approaches. Finally, we conclude the paper.

II. PROBLEMATIC

In this paper, we focus on a set $\{\tau_1, \tau_2, \ldots, \tau_n\}$ of $n$ flows coexisting in a network and show how to provide deterministic (see Section III) or probabilistic (see Section IV) end-to-end QoS guarantees to any flow $\tau_i$. For this, we adopt the following assumptions:

Assumption 1: Flows are scheduled according to the non-preemptive FP/FIFO: packets are 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 there is no relationship between the end-to-end deadline of a flow and its fixed priority.

Assumption 2: Links interconnecting nodes are FIFO and the network delay between two nodes has known lower and upper bounds: $L_{\text{min}}$ and $L_{\text{max}}$. Moreover, network is reliable: neither network failures nor packet losses are considered.

---

1 The transmission of a packet cannot be interrupted by another packet with a higher priority.
Assumption 3: Each flow \( \tau_i \) follows a fixed\(^2\) route \( \mathcal{H}_i \) that is an ordered sequence of nodes whose first node is the ingress node of the flow.

Moreover, for any flow \( \tau_i \), we adopt the following notations:

- \( D_i \), its end-to-end deadline;
- \( F_i \), its fixed priority;
- \( hp_i = \{ j \in [1, n] \mid F_j > F_i \} \), the set of flows having a strictly higher fixed priority;
- \( sp_i = \{ j \in [1, n] \mid j \neq i, F_j = F_i \} \), the set of flows having the same fixed priority;
- \( lp_i = \{ j \in [1, n] \mid F_j < F_i \} \), the set of flows having a strictly lower fixed priority;
- \( first_i \), its first node visited in the network;
- \( last_i \), its last node visited in the network;
- \( \mathcal{H}_i = \{ first_i, ..., last_i \} \), its path;
- \( |\mathcal{H}_i| \), the number of its visited nodes.

### III. DETERMINISTIC APPROACH

#### A. 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 induce a maximum jitter on the next visited node, 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 \( \tau_i \) on a node \( h \) does not generally result in a worst case scenario for \( \tau_i \) 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. More precisely, it consists in moving backwards through the sequence of nodes visited by a flow packet, each time identifying preceding packets and busy periods that ultimately affect the delay of this packet. This approach is used in this section.

\(^2\)For instance, MPLS [3] can be used to fix the route followed by a flow.

#### B. Additional assumptions and notations

First, we introduce the following assumptions.

**Assumption 4:** Flows are characterized by sporadic arrivals. Hence, each flow \( \tau_i \) is defined by:

- \( T_i \), the minimum interarrival time (called period) between two successive packets of \( \tau_i \);
- \( C^h_i \), the maximum processing time on node \( h \) of a packet of \( \tau_i \). This parameter depends on the maximum packet size and the capacity of its output link;
- \( J_i \), the maximum release jitter of packets of \( \tau_i \) 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.

Let \( first_{j,i} \) (resp. \( last_{j,i} \)) be the first (resp. the last) node visited by flow \( \tau_j \) on path \( \mathcal{H}_i \). Figure 1 illustrates the notations of \( first_{i,j}, first_{j,i}, last_{i,j} \) and \( last_{j,i} \) when flows \( \tau_i \) and \( \tau_j \) are (1) in the same direction and (2) in reverse directions.

![Figure 1. first_{i,j}, first_{j,i}, last_{i,j} and last_{j,i}](image)

Assumption 5: For any flow \( \tau_i \), for any flow \( \tau_j \), \( j \in hp_i \cup sp_i \), 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 \([last_{j,i}, first_{j,i}] \subseteq \mathcal{H}_i \).

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 5.

**Assumption 6:** Time is discrete.

Indeed, 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 [8].

Finally, we consider the following notations:

- \( \forall a \in \mathbb{R}, (1 + [a])^+ \) stands for \( \max(0; 1 + [a]) \);
- \( S_{min}^h \), the minimum time taken by a packet of flow \( \tau_i \) to go from its source node to node \( h \);
- \( S_{max}^h \), the maximum time taken by a packet of flow \( \tau_i \) to go from its source node to node \( h \);
- \( M^h = \sum_{h' = first_{i,j}}^{pre((h'))} \min_{j \in hp_{i,j} \cup sp_{i,j}} \{ C^h_j \} + L_{min} \).

### C. Latest starting time

We focus on a packet \( m \) belonging to any flow \( \tau_i \), 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 \( W_{last}^i \), the latest starting time of \( m \) on its last node visited. As shown in [10], this time is equal to: \( X_{i,t} + \delta_i + (|\mathcal{H}_i| - 1) \cdot L_{max} \), where:
• $X_{i,t}$ is the delay due to packets having a priority higher than or equal to that of $m$. These packets are either having a fixed priority strictly higher than $F_i$ or those having the fixed priority $F_i$ but arrived before $m$ on a common node. Its maximum value is given in Lemma 1;  
• $\delta_i$ is the delay directly due to the non-preemptive effect. Its maximum value is given in Lemma 2;  
• $(|H_i|-1) \cdot L_{\text{max}}$ represents the maximum network delays.

**Lemma 1:** Let $m$ be the packet of flow $\tau_i$ generated at time $t$. When flows are scheduled $\text{FP/FIFO}$, the maximum delay incurred by $m$ due to packets with a priority higher than or equal to $m$ is bounded by:

$$
\sum_{j \in h_{p_i}} \left( 1 + \left[ \frac{W_{\text{last},j}^{l,t} - S_{\text{min},j}^{l,t} + A_{i,j}}{T_j} \right]^+ \cdot C_{j,\text{flow},j,i} \right)^+ + \sum_{j \in s_{p_i}\cup\{i\}} \left( 1 + \left[ \frac{t + S_{\text{max},j}^{\text{first},j,i} - S_{\text{min},j}^{\text{first},j,i} + A_{i,j}}{T_j} \right]^+ \cdot C_{j,\text{flow},j,i} \right)^+ + \sum_{h \in H_i, h \neq \text{slow}} \max_{j \in h_{p_i} \cup s_{p_i} \cup \{i\}} \left\{ C_{h,j}^{\text{slow},j,i} - C_{i,\text{last},i}^{\text{last},i,i} \right\} - C_{i,\text{last},i}^{\text{last},i,i} - \sum_{j \in s_{p_i}\cup\{i\}} \left\{ C_{h,j}^{\text{slow},j,i} \right\}
$$

with: $A_{i,j} = S_{\text{max},j}^{\text{first},j,i} - M_{i}^{\text{first},j,i} + J_j$.  

**Lemma 2:** Let $\tau_i$ be a flow following path $H_i$. When flows are scheduled $\text{FP/FIFO}$, the maximum delay incurred by a packet of flow $\tau_i$ directly due to flows belonging to $l_{p_i}$, denoted $\delta_i$, is bounded by $\sum_{h \in H_i} (\max_{j \in l_{p_i}} \{C_{h,j}\} - 1)^+$, where $\max_{j \in l_{p_i}} \{C_{h,j}\} = 0$ if there is no flow in $l_{p_i}$ visiting node $h$.

D. Worst case end-to-end response time

Let $R_{i,t}$ be the worst case end-to-end response time of the packet of flow $\tau_i$ generated at time $t$. We have:

$$R_{i,t} = X_{i,t} + C_{i,\text{last},i}^{\text{last},i,i} + \delta_i + (|H_i|-1) \cdot L_{\text{max}}.$$  
The worst case end-to-end response time of flow $\tau_i$, denoted $R_{i,t}$, is then equal to: $\max_{t \geq -J_i} \{R_{i,t}\}$. In order not to test all times $t \geq -J_i$, we use Lemma 3, proved in [10].

**Lemma 3:** Let us consider a flow $\tau_i$ following a path $H_i$. If flows are scheduled $\text{FP/FIFO}$, then for any time $t \geq -J_i$,

$$W_{\text{last},i}^{l,t} + B_{i,\text{flow}}^{\text{flow}} \leq W_{\text{last},i}^{l,t} + B_{i,\text{flow}}^{\text{flow}},$$

with:

$$B_{i,\text{flow}}^{\text{flow}} = \sum_{j \in h_{p_i} \cup s_{p_i} \cup \{i\}} \left[ B_{i,\text{flow}}^{\text{flow}} / T_j \right] \cdot C_{j,\text{flow},j,i}.$$

The worst case end-to-end response time of flow $\tau_i$ is given in the following property.

**Property 1:** When flows are scheduled $\text{FP/FIFO}$, the worst case end-to-end response time of any flow $\tau_i$ is bounded by $R_{i} = \max_{-J_i \leq t \leq -J_i} + B_{i,\text{flow}}^{\text{flow}} \{W_{\text{last},i}^{l,t} + C_{i,\text{last},i}^{\text{last},i,i} - t\}$, with:

$$W_{\text{last},i}^{l,t} = \sum_{j \in h_{p_i}} \left( 1 + \left[ \frac{W_{\text{last},j}^{l,t} - S_{\text{min},j}^{l,t} + A_{i,j}}{T_j} \right]^+ \cdot C_{j,\text{flow},j,i} \right)^+ + \sum_{j \in s_{p_i}\cup\{i\}} \left( 1 + \left[ \frac{t + S_{\text{max},j}^{\text{first},j,i} - S_{\text{min},j}^{\text{first},j,i} + A_{i,j}}{T_j} \right]^+ \cdot C_{j,\text{flow},j,i} \right)^+ + \sum_{h \in H_i, h \neq \text{slow}} \max_{j \in h_{p_i} \cup s_{p_i} \cup \{i\}} \left\{ C_{h,j}^{\text{flow},j,i} - C_{i,\text{last},i}^{\text{last},i,i} - \sum_{j \in s_{p_i}\cup\{i\}} \left\{ C_{h,j}^{\text{flow},j,i} \right\} \right\} + \delta_i + (|H_i|-1) \cdot L_{\text{max}}.$$

and: $B_{i,\text{flow}}^{\text{flow}} = \sum_{j \in h_{p_i} \cup s_{p_i} \cup \{i\}} \left[ B_{i,\text{flow}}^{\text{flow}} / T_j \right] \cdot C_{j,\text{flow},j,i}.$

**E. 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 $\tau_i, j \in [1, n]$, be a flow of priority $F_i$. We compute the set $S_i$ of flows crossing directly or indirectly $\tau_i$ and apply Property 1 to compute the worst case response time of $\tau_i$. More formally, we determine $S_i$ as follows:

1) $S_i = \{\tau_i\}$;  
2) $S_i = S_i \cup \{\tau_j, j \in h_{p_i} \cup s_{p_i}, \tau_j$ crosses directly $\tau_i\}$;  
3) $S_i = S_i \cup \{\tau_k, j \in h_{p_i} \cup s_{p_i}, \exists \tau_j \in S_i$ such that $\tau_k$ 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 $F_i$.

IV. PROBABILISTIC APPROACH

The probabilistic approach enables to guarantee to each flow $\tau_i$ that the end-to-end response time of any of its packets does not exceed the deadline $D_i$ with a given probability. To obtain this probability, the end-to-end response time distribution must be computed. We consider in this section that flows are characterized by Poisson arrivals.

**Assumption 7:** At network entries, the flows arrive according to a Poisson process. Each flow $\tau_i$ is a Poisson process with parameter $\lambda_i$.

Indeed, the arrival process of the Internet traffic, at a packet scale, does not correspond to a Poisson process (e.g., on-off sources). However, it can be overestimated by a Poisson process [4].

A. Node response time distribution

To compute the end-to-end response time distribution of any flow $\tau_i$, 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 $\tau_i$ 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 Assumption 8.

**Assumption 8:** Packet arrivals of any flow $\tau_i$ to a link $ab$ is also a Poisson process with the parameter $\lambda_i^{ab}$, that is equal to $\lambda_i$ if $ab$ belongs to the path of $\tau_i$, 0 otherwise.
According to the traffic description, Assumption 8 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. The average arrival rate of packets of class i (flow \( \tau_i \)) on the link \( ab \) is \( \lambda_i^{ab} \) and the average service rate of packets of \( \tau_i \) on the link \( ab \) is \( \mu_i^{ab} \). The node response time distribution for packets of flow \( \tau_i \) at the link \( ab \) is obtained by inspecting its Laplace transform which will be denoted by \( (S_i^{ab})^* (s) \) and is given by:

\[
(S_i^{ab})^* (s) = (W_i^{ab})^* (s) \cdot (B_i^{ab})^* (s),
\]

where \( (W_i^{ab})^* (s) \) is the Laplace transform of the waiting time density of packets having the priority \( F_i \) at the link \( ab \) and \( (B_i^{ab})^* (s) \) is the Laplace transform of the service time probability density function of packets of flow \( \tau_i \) at the link \( ab \). Indeed, packets having the same priority have the same waiting time distribution. We first focus on the computation of \( (W_i^{ab})^* (s) \). A packet having the priority \( F_i \) must wait for:

- packets having priority greater or equal to \( F_i \) and found in the queue upon the arrival of our tagged packet;
- packets having priority greater than \( F_i \) 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 [11], we define two categories of packets: those belonging to flows in \( h_{pi} \cup sp_i \cup \{ \} \) (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: \( \lambda_i^{ab} = \sum_{j \in h_{pi} \cup sp_i \cup \{ \}} \lambda_j^{ab} \) and \( \lambda_i^{ab} = \sum_{j \in lp_i} \lambda_j^{ab} \).

The Laplace transforms of the service time densities of priority and ordinary packets are respectively:

\[
(B_i^{ab})^* (s) = \sum_{j \notin h_{pi} \cup sp_i \cup \{ \}} (\lambda_j^{ab} / \lambda_i^{ab}) \cdot (B_j^{ab})^* (s) \quad \text{and} \quad (B_i^{ab})^* (s) = \sum_{j \in lp_i} (\lambda_j^{ab} / \lambda_i^{ab}) \cdot (B_j^{ab})^* (s).
\]

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 [12]:

- the service time of the packet in service upon the arrival of our tagged packet and the packets having priority greater than or equal to \( F_i \) and waiting in the system at this time. This time will be denoted by \( W_i^{ab} \);
- the service time of packets having priority greater than \( F_i \) that arrive during \( W_i^{ab} \) and the duration of all busy periods generated by these packets.

Let \( (W_i^{ab})^* (s) \) the Laplace transform of the waiting time density of priority packets. \( (W_i^{ab})^* (s) \) is given by [12]:

\[
(W_i^{ab})^* (s) = \frac{(1 - \rho_i^{ab})^{s + \lambda_i^{ab}} (1 - (B_i^{ab})^* (s))}{s - \lambda_i^{ab} + \lambda_i^{ab} (B_i^{ab})^* (s)},
\]

where \( \rho_i^{ab} \) is the utilization factor of the link server \( ab \).

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

\[
(W_i^{ab})^* (s) = \left( \frac{1 - \rho_i^{ab}}{s - \lambda_i^{ab} + \lambda_i^{ab} (B_i^{ab})^* (s)} \right) ^{s + \lambda_i^{ab}} \left( 1 - (B_i^{ab})^* (s) \right),
\]

where \( (\theta_i^{ab})^* (s) \) corresponds to the Laplace transform of a busy period delay density generated by packets having a priority strictly greater than \( F_i \) and is the solution of the equation:

\[
(\theta_i^{ab})^* (s) = \left( B_i^{ab} \right)^* (s) \cdot \lambda_i^{ab} = \left( \theta_i^{ab} \right)^* (s).
\]

The Laplace transform of the node response time distribution for packets of flow \( \tau_i \) at the link \( ab \), denoted \( (S_i^{ab})^* (s) \), is then equal to:

\[
(B_i^{ab})^* (s) = \frac{(1 - \rho_i^{ab}) \sigma_i + \sum_{j \in j_{\pi_i} \cup \{ \}} \lambda_j^{ab} \lambda_j^{ab} \left( (B_j^{ab})^* (s) \right)}{s + \sum_{j \in h_{pi}} \lambda_j^{ab} \sum_{j \in \pi_j \cup \{ \}} \lambda_j^{ab} \lambda_j^{ab} + \sum_{j \in \pi_j \cup \{ \}} \lambda_j^{ab} \lambda_j^{ab} \left( B_j^{ab} \right)^* (s) \right) \cdot \left( (S_i^{ab})^* (s) \right) + \left( (S_i^{ab})^* (s) \right).
\]

B. End-to-end response time distribution

Let \( \tau_i \) be the random variable representing the end-to-end response time for a packet of flow \( \tau_i \) and \( S_i\tau_i^*(s) \) 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 \( \tau_i \) 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: \( \tau_i = \sum_{h=first_i}^{h_{pre;i}(\tau_i)} (\tau_{i_{\text{trans};h}}(\tau_i) + d) \), with if the random variable corresponding to the transmission delays between two nodes and \( \tau_i \) the node visited by \( \tau_i \) just before (respectively after) node \( h \). The random variables corresponding to these different durations being independent, we obtain:

\[
S_i\tau_i^*(s) = \left( L^*(s) \right)^{|H_i| - 1} \prod_{h=first_i}^{h_{pre;i}(\tau_i)} \left( (\tilde{S}_i^*(s)) \right),
\]

where \( \left( \tilde{S}_i^*(s) \right) \) is the Laplace transform of \( \tilde{d} \) belonging to \( H_i \), and \( L^*(s) \) is the Laplace transform of \( \tilde{d} \) belonging to \( H_i \).

C. Probabilistic QoS guarantee and p-schedulability

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

**Property 2:** A packet belonging to the flow \( \tau_i \) with a relative deadline \( D_i \) meets its deadline with the probability:

\[
P_{\text{success}}(D_i) = P[\tau_i < D_i] = \int_0^{D_i} s_i(t)dt,
\]

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. Moreover, it enables to generalize the concept of flow schedulability to the probabilistic world, in introducing the p-schedulability.

**Definition 1:** The schedulability of a flow set \( \{\tau_1, \tau_2, \ldots, \tau_n\} \) can be measured by the probability:

\[
P_{\text{success}} = \prod_{i=1}^{n} P_{\text{success}}(D_i),
\]

where \( P_{\text{success}}(D_i) \) is obtained from Property 2.

Hence, the end-to-end QoS requirements of all flows are met with the probability \( P_{\text{success}} \).

**Definition 2:** A flow set \( \{\tau_1, \tau_2, \ldots, \tau_n\} \) is p-schedulable if and only if \( p \leq P_{\text{success}} \).

**D. Model validation**

In this section, we validate our analytical model by means of simulations with NS2. So, we consider a network composed of 8 nodes crossed by 4 flows \( \{\tau_1, \tau_2, \tau_3, \tau_4\} \) characterized in Table I. The end-to-end miss probability of flows \( \tau_1 \) and \( \tau_4 \) are represented in Figure 2. It shows that our analytical results, obtained as described in Section IV, are very close to the simulation results. Hence, our analytical model is validated on this example.

**TABLE I**

<table>
<thead>
<tr>
<th>Flow</th>
<th>( \tau_1 )</th>
<th>( \tau_2 )</th>
<th>( \tau_3 )</th>
<th>( \tau_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed priority</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Deadline (ms)</td>
<td>0.1</td>
<td>0.2</td>
<td>0.12</td>
<td>0.6</td>
</tr>
<tr>
<td>Average service time (( \mu s ))</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
</tbody>
</table>

![Fig. 2. The end-to-end miss probability](image)

**V. COMPARATIVE EVALUATION**

In this section, we compare the two types of QoS guarantees obtained by the deterministic and probabilistic approaches. We first present the following property enabling to overestimate sporadic arrivals by Poisson ones.

**Property 3:** The sporadic arrivals of any flow \( \tau_i \) can be upper bounded by Poisson arrivals characterized by:

\[
\lambda_i = \frac{1}{T_i} \quad \text{and} \quad \mu_i = \frac{1}{C_i}, \quad \text{with} \quad C_i \quad \text{the average processing time of a packet of} \tau_i \text{in node} \ h.
\]

Proof: See [4].

We now present an example of six flows coexisting in a network, whose characteristics are given in Table II.

**TABLE II**

<table>
<thead>
<tr>
<th>Flow</th>
<th>( \tau_1 )</th>
<th>( \tau_2 )</th>
<th>( \tau_3 )</th>
<th>( \tau_4 )</th>
<th>( \tau_5 )</th>
<th>( \tau_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_i (\mu s) )</td>
<td>100</td>
<td>100</td>
<td>2240</td>
<td>100</td>
<td>560</td>
<td>560</td>
</tr>
<tr>
<td>( T_i (\mu s) )</td>
<td>8000</td>
<td>8000</td>
<td>4870</td>
<td>340</td>
<td>4010</td>
<td>4010</td>
</tr>
<tr>
<td>( C_i (\mu s) )</td>
<td>51.2</td>
<td>51.2</td>
<td>1168</td>
<td>51.2</td>
<td>460</td>
<td>460</td>
</tr>
</tbody>
</table>

Links are 10 Mbit/s and paths of flows considered are the following:

- \( \tau_1: 1 \rightarrow 2 \rightarrow 4 \rightarrow 10 \)
- \( \tau_2: 8 \rightarrow 2 \rightarrow 4 \rightarrow 11 \)
- \( \tau_3: 1 \rightarrow 2 \rightarrow 6 \)
- \( \tau_4: 3 \rightarrow 4 \rightarrow 11 \)
- \( \tau_5: 7 \rightarrow 2 \rightarrow 4 \rightarrow 5 \)
- \( \tau_6: 9 \rightarrow 4 \rightarrow 11. \)

With the deterministic approach, the worst case response times are obtained by applying Property 1 and are given in Table III.

**TABLE III**

<table>
<thead>
<tr>
<th>Flow</th>
<th>( \tau_1 )</th>
<th>( \tau_2 )</th>
<th>( \tau_3 )</th>
<th>( \tau_4 )</th>
<th>( \tau_5 )</th>
<th>( \tau_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_i (\mu s) )</td>
<td>5438</td>
<td>3758</td>
<td>7479</td>
<td>1519</td>
<td>15820</td>
<td>5240</td>
</tr>
</tbody>
</table>

![Fig. 3. Worst case scenario of \( \tau_5 \)](image)

Notice that bounds obtained with the deterministic approach are close or equal to the exact worst case end-to-end response times. For example, for flow \( \tau_3 \), its bound is reached in the following worst case scenario illustrated in Figure 3:

- On node 1, flows \( \tau_1 \) and \( \tau_3 \) generate a packet at time 0. As the fixed priority of \( \tau_1 \) is higher than that of \( \tau_3 \), the packet of \( \tau_3 \) is delayed by the packet of \( \tau_1 \).
- On node 2, as flow \( \tau_5 \) follows an independent path until this node, its packets can arrive at any time. Then, we assume that a packet of \( \tau_5 \) arrives at time 2339. Therefore, this packet starts its execution one time unit before the arrival of the packet of \( \tau_3 \). The non-preemptive effect is then maximized for that packet. Moreover, if a packet of \( \tau_2 \) arrives during the processing of the packet of \( \tau_5 \), it is processed before the packet of \( \tau_3 \).
- On node 6, the packet of flow \( \tau_3 \) arrives at time 5239. Hence, as \( \tau_3 \) is the only flow visiting this node, its worst case end-to-end response time is equal to 5239 plus 2240 (its maximum processing time on this node), that is 7479.
This example shows that we can provide to each flow a probability of success close to one for a deadline much lower than its worst case end-to-end response time. Notice that the probability \( P_{\text{success}} \) cannot in any case equal 1, as a Poisson arrival process and a service time exponentially distributed have been considered. Moreover, the flow set considered is schedulable with a probability depending of the required end-to-end deadlines. For instance, the last column of Table IV shows that the flow set is schedulable with a probability equal to 97.86% for deadlines smaller than the worst case end-to-end response times given in Table III. Indeed, the deterministic approach may lead to a bound that can be reached infrequently.

The network dimensioning based on the deterministic bounds can be expensive in terms of resources. For instance, the worst case end-to-end response time of flow \( \tau_5 \) is 13820. However, according to the probabilistic approach, we can guarantee that its end-to-end response time does not exceed 3000 with a probability of 99.98%.

We now assume that the six flows have the following deadlines: \( D_1 = 3000 \), \( D_2 = 1000 \), \( D_3 = 6000 \), \( D_4 = 500 \), \( D_5 = 3000 \) and \( D_6 = 1000 \). If each flow is satisfied with a probability of success equal to 98%, then the probabilistic approach allows to accept the six flows, as shown in Table IV, unlike the deterministic approach, as shown in Table III.

Offering deterministic guarantees is necessary for hard real-time applications, that do not tolerate any deadline violation. However, it leads to a low resource utilization and many applications do not require such guarantees. Indeed, soft real-time applications do not require such guarantees. This example shows that we can provide to each flow a probability of success close to one for a deadline much lower than its worst case end-to-end response time. Notice that the probability \( P_{\text{success}} \) cannot in any case equal 1, as a Poisson arrival process and a service time exponentially distributed have been considered. Moreover, the flow set considered is schedulable with a probability depending of the required end-to-end deadlines. For instance, the last column of Table IV shows that the flow set is schedulable with a probability equal to 97.86% for deadlines smaller than the worst case end-to-end response times given in Table III. Indeed, the deterministic approach may lead to a bound that can be reached infrequently.

The network dimensioning based on the deterministic bounds can be expensive in terms of resources. For instance, the worst case end-to-end response time of flow \( \tau_5 \) is 13820. However, according to the probabilistic approach, we can guarantee that its end-to-end response time does not exceed 3000 with a probability of 99.98%.

We now assume that the six flows have the following deadlines: \( D_1 = 3000 \), \( D_2 = 1000 \), \( D_3 = 6000 \), \( D_4 = 500 \), \( D_5 = 3000 \) and \( D_6 = 1000 \). If each flow is satisfied with a probability of success equal to 98%, then the probabilistic approach allows to accept the six flows, as shown in Table IV, unlike the deterministic approach, as shown in Table III.

Offering deterministic guarantees is necessary for hard real-time applications, that do not tolerate any deadline violation. However, it leads to a low resource utilization and many applications do not require such guarantees. Indeed, soft real-time applications are satisfied with probabilistic QoS guarantees. Providing such guarantees leads to a better resource utilization rate and allows us to accept a higher number of flows in the network. That is the reason why deterministic QoS guarantees should be restricted to applications with hard QoS constraints.

### VI. Conclusion and Perspectives

FP scheduling is used when flows have different importance degrees. 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 on their end-to-end response times. We have compared on an example the two types of quantitative QoS guarantees: deterministic and probabilistic.

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 and validated by simulations. 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 the required one. As shown in the example, bounds obtained with the deterministic approach are close or equal to the exact worst case end-to-end response times. In addition, the probabilistic approach allows us to guarantee end-to-end deadlines with probabilities sufficient for many applications.

As a further work, we will study how to derive an admission control from our results. The admission control has to be done on line, its complexity should remain limited. Then, approximations can be used, in order to reduce the complexity. However, approximations will lead to a less accurate view of the network state and so to an inaccurate decision. A tradeoff should be found to obtain a pertinent on line decision.

This study shows that offering probabilistic QoS guarantees leads to a better network resource utilization. Moreover, such guarantees are sufficient for a lot of multimedia applications where a low loss rate can be tolerated. Deterministic QoS guarantees should be reserved to hard real-time applications, that do not tolerate a deadline violation. Offering both types of QoS guarantees in a network would constitute a very promising tradeoff. This is our further research direction.

### REFERENCES