Worst case end-to-end response times of flows scheduled with FP/FIFO

Steven Martin
LRI, Univ. 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 are interested in real-time flows requiring quantitative and deterministic Quality of Service (QoS) guarantees. We focus more particularly on two QoS parameters: the worst case end-to-end response time and jitter. We consider a non-preemptive scheduling of flows, called FP/FIFO, based on fixed priorities. On each node, packets are scheduled according to their fixed priority, if several packets share the same one, they are scheduled according to their arrival time on the node considered. The fixed priority associated with a flow denotes the importance of the flow from the user point of view. The scheduling FP/FIFO is the most common implementation of FP. We show how to take into account the FIFO arbitration in the worst case analysis, based on the trajectory approach, allowing to establish a bound on the worst case end-to-end response time of any flow in the network. Finally, we present an example illustrating our results.

Keywords: Fixed priority scheduling, QoS, real-time scheduling, worst case end-to-end response time, trajectory approach, deterministic guarantee, FP/FIFO.

1 Context and motivations

In this paper, we are interested in real-time applications that require bounds on the worst case end-to-end response times and jitters to have a behavior compliant with their specifications (e.g. voice over IP and control-command applications). That is why we focus on deterministic guarantees of end-to-end response times and jitters in a packet network. We will show how to determine these times depending on the flow scheduling used in the network. With regard to flow scheduling, the assumption generally admitted is that packet transmission is not preemptive. Moreover, *Fixed Priority* (FP) scheduling has been extensively studied in the last years [1, 2]. It exhibits interesting properties. Indeed, the impact of a new flow is limited to flows having equal or lower fixed priorities, it is easy to implement and well adapted for service differentiation.

In a network, several packets can share the same fixed priority: for example, if the number of fixed priorities is less than the flow number, or if flows are processed by service class and the flow priority is this of its class. In this paper, we assume that such packets are scheduled according to their arrival time on the node considered. More precisely, we assume that packets are scheduled according to the non-preemptive FP/FIFO scheduling. 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. This corresponds to the most common implementation of FP. Unlike the state of the art, we take into account this FIFO arbitration of packets having the same fixed priority to compute the worst case end-to-end response times. In [3] we proved that in a uniprocessor context, the use of FP/FIFO instead of FP improves the schedulability of flows on each visited node. In this paper, we show how to extend this analysis to the distributed case, using the trajectory approach.

These results can be applied in a DiffServ architecture to determine the worst case end-to-end response time granted to flows belonging to the Expedited Forwarding (EF) class, assuming that routes of EF flows remain fixed, once assigned.

2 Problematic

We investigate the problem of providing a deterministic guarantee (i.e. an upper bound) on the end-to-end response time to any flow in a network. As we make no particular assumption concerning the arrival times of packets in the network, the feasibility of a set of flows is equivalent to meet the requirement, whatever the arrival times of the packets in the network. We assume that time is discrete. Reference [4] shows that results obtained with a discrete scheduling are as general as those obtained with a continuous scheduling when all flow parameters are multiples of the node clock tick. Moreover, we assume the following models.

Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)
0-7695-2552-0/06 $20.00 © 2006 IEEE

2.1 Models

Scheduling model All nodes in the network schedule packets according to the non-preemptive[1] FP/FIFO algorithm.

Network model We consider a network where links interconnecting nodes are supposed to be FIFO and the network delay between two nodes has known lower and upper bounds: L_{min} and L_{max}. Moreover, we consider neither network failures nor packet losses.

Traffic model We consider a set $\{\tau_1, ..., \tau_n\}$ of n sporadic flows. Each flow τ_i follows a path $\mathcal{P}_i$ that is an ordered sequence of nodes whose first node is the ingress node of the flow. Moreover, a sporadic flow τ_i is defined by:

- T_i, the minimum interarrival time between two successive packets of flow τ_i;
- C_i^h, the maximum processing time on node $h \in \mathcal{P}_i$ of a packet of flow τ_i. By convention, $C_i^h = 0$ if $h \notin \mathcal{P}_i$;
- J_i, the maximum release jitter of packets of flow τ_i at its ingress node. A packet is subject to a release jitter if there exists a non-null delay between its generation time and the time, called its release time, where it is taken into account by the scheduler;
- D_i, the end-to-end deadline of flow τ_i, that is its maximum end-to-end response time acceptable. A packet of flow τ_i generated at time t must be delivered at $t + D_i$;
- P_i, the fixed priority of flow τ_i.

2.2 Notations

We consider any flow τ_i, $i \in [1, n]$, following a path $\mathcal{P}_i$ and focus on the packet m of τ_i generated at time t. We then define the three following sets:

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

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 $h \in \mathcal{P}_i \cap \mathcal{P}_j$, priority of packet m is higher than or equal to this of packet m' if and only if: $(P_i > P_j)$ or $(P_i = P_j$ and m arrives before m' on node $h)$.*

We also adopt the following notations:

[1]The scheduler of the node considered waits for the completion of the current packet transmission (if any) before selecting the next packet.

- 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;
- $\mathcal{P}_i = [first_i, ..., last_i]$, the path followed by flow τ_i;
- $|\mathcal{P}_i|$, the number of nodes visited by flow τ_i;
- $slow_i$, the slowest node visited by flow τ_i on path $\mathcal{P}_i$: $\forall\, h \in \mathcal{P}_i, C_i^{slow_i} \geq C_i^h$;
- $first_{j,i}$, the first node visited by flow τ_j on path $\mathcal{P}_i$;
- $last_{j,i}$, the last node visited by flow τ_j on path $\mathcal{P}_i$;
- $slow_{j,i}$, the slowest node visited by τ_j on path $\mathcal{P}_i$: $\forall\, h \in \mathcal{P}_i \cap \mathcal{P}_j, C_j^{slow_{j,i}} \geq C_j^h$;
- $Smin_i^h$, 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 $\mathcal{P}_i$;
- $pre_i(h)$, 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 $h \notin \mathcal{P}_i$. 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.

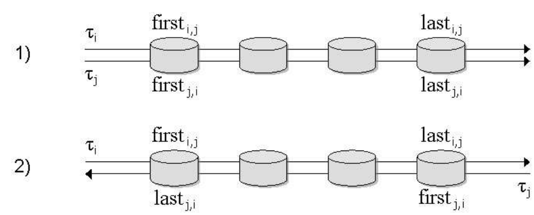

Figure 1. $first_{i,j}$, $first_{j,i}$, $last_{i,j}$ *and* $last_{j,i}$

Moreover, we assume, with regard to flow τ_i following path $\mathcal{P}_i$, that any flow τ_j, $j \in hp_i \cup sp_i$ following path $\mathcal{P}_j$ with $\mathcal{P}_j \neq \mathcal{P}_i$ and $\mathcal{P}_j \bigcap \mathcal{P}_i \neq \emptyset$ never visits a node of path $\mathcal{P}_i$ after having left this path.

Assumption 1 *For any flow τ_i following path $\mathcal{P}_i$, for any flow τ_j, $j \in hp_i \cup sp_i$, following path $\mathcal{P}_j$ such that $\mathcal{P}_j \cap \mathcal{P}_i \neq \emptyset$, we have either $[first_{j,i}, last_{j,i}] \subseteq \mathcal{P}_i$ or $[last_{j,i}, first_{j,i}] \subseteq \mathcal{P}_i$.*

To achieve that, the idea is to consider a flow crossing path $\mathcal{P}_i$ after it left $\mathcal{P}_i$ as a new flow. We proceed by iteration until meeting Assumption 1.

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: $R_i - (\sum_{h \in \mathcal{P}_i} C_i^h + (|\mathcal{P}_i| - 1) \cdot L_{min})$.*

3 Related work

Deterministic and quantitative guarantees can be provided by at least three approaches, that 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 $h+1$ that 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 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: the sequence of nodes visited. This approach is described in Section 4.

4 Worst case analysis: the trajectory approach

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. 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 $\mathcal{P}_i$ 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 that, 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.

4.1 Study of the trajectory of packet m

To compute the latest starting time of packet m, we proceed as follows. We first determine bp^q, that is the busy period[2] 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 this 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 level corresponding to the priority of packet $p(1)$ in which the packet $f(1)$ is processed (see Figure 2).

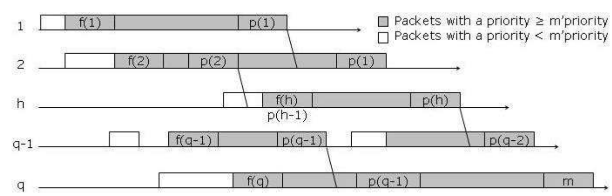

Figure 2. Response time of packet m

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 $m'-1$ (resp. $m'+1$) the packet preceding (resp. succeeding to) m'. Moreover, we denote $a_{m'}^h$ the arrival time of m' on node h and consider that $a_{f(1)}^1 = 0$. 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} $-$ $(a_{p(1)}^2 - a_{f(2)}^2)$ $+$ ... $+$ the processing time on node q of packets $f(q)$ to $(m-1)$ $-$ $(a_{p(q-1)}^q - a_{f(q)}^q)$ $+$ δ_i.

[2] 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.

We can notice that on any node $h \in \mathcal{P}_i$, if there exists no flow τ_j such that $h = first_{j,i}$, then $p(h-1) = f(h)$ and so $a^h_{p(h-1)} - a^h_{f(h)} = 0$. In other words, if $p(h-1) \neq f(h)$, there exists a flow τ_j such that $h = first_{j,i}$. In such a case, by definition of $p(h)$, all the packets in $[f(h), p(h-1))$ cross path $\mathcal{P}_i$ for the first time at node h. We can then act on their arrival times. Postponing the arrivals of these packets in the busy period where $p(h-1)$ is processed would increase the departure time of m from node q. Hence, in the worst case, $p(h) = f(h+1)$ on any node $h \in \mathcal{P}_i$. Moreover, in the worst case, on any node h visited by τ_i, the fixed priority of the packet $f(h)$ is this 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 having a priority higher than or equal to m;
- δ_i, the delay due to the non-preemptive effect;
- $(q-1) \cdot L_{max}$, the maximum network delay.

In the two following subsections, we evaluate $X_{i,t}$ and δ_i.

4.2 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 $h \in \mathcal{P}_i$ with a packet $m' \in (f(h), f(h+1))$, then for any node $h' \in \mathcal{P}_i - \{h\}$, $m' \notin (f(h'), f(h'+1))$.*

Proof: By induction. Let us consider any packet m' processed in $(f(1), f(2))$ on node 1. By definition, we have $P_{\tau(m')} \geq P_{\tau(f(2))} \geq P_i$. 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 $h' \in \mathcal{P}_i - \{1\}$. ■

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:

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

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:

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

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 maximum. Then, we get Lemma 2.

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:*

$$\sum_{j\in hp_i}\left(1 + \left\lfloor \frac{W^{last_{i,j}}_{i,t} - Smin_j^{last_{i,j}} + A_{i,j}}{T_j} \right\rfloor\right)^+ \cdot C^{slow_{j,i}}_j$$
$$+ \sum_{j\in sp_i\cup\{i\}}\left(1 + \left\lfloor \frac{t + Smax_i^{first_{j,i}} - Smin_j^{first_{j,i}} + A_{i,j}}{T_j} \right\rfloor\right)^+ \cdot C^{slow_{j,i}}_j$$
$$+ \sum_{\substack{h\in\mathcal{P}_i\\ h\neq slow_i}} \max_{\substack{j\in hp_i\cup sp_i\cup\{i\}\\ first_{j,i}=first_{i,j}}} \{C^h_j\} - C^{last_i}_i,$$

with: $A_{i,j} = Smax_j^{first_{i,j}} - M_i^{first_{i,j}} + J_j$.

Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)
0-7695-2552-0/06 $20.00 © 2006 IEEE

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 flow 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. ■

4.3 Delay due to non-preemption

We recall that packet scheduling is non-preemptive. Hence, despite the high priority of any packet m, generated at time t, a packet with a lower priority can delay m processing due to non-preemption. Indeed, if a packet m of any flow τ_i 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, arrived after m on the node but before m starts its execution. Then, we denote δ_i the maximum delay incurred by packet m while following its path directly due to the non-preemptive effect.

Property 1 *Let τ_i, $i \in [1, n]$, be a flow following path $\mathcal{P}_i = [first_i, ..., last_i]$. 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:*

$$\Big(\max_{\substack{j \in lp_i \\ first_{j,i} = first_i}} \{C_j^{first_i}\} - 1 \Big)^+$$

$$+ \sum_{\substack{h \in \mathcal{P}_i \\ h \neq first_i}} \Big(\max_{\substack{j \in lp_i \\ first_{j,i} = h}} \{C_j^h\} - 1; \max_{\substack{j \in lp_i \\ h \in (first_{j,i}, last_{j,i}] \\ first_{j,i} \neq first_{i,j}}} \{C_j^h\} - 1;$$

$$1_\alpha \cdot (\max_{\substack{j \in lp_i \\ h \in (first_{j,i}, last_{j,i}] \\ first_{j,i} = first_{i,j}}} \{C_j^h\} - C_i^{pre_i(h)} + Lmax - Lmin) \Big)^+,$$

where $\max_{j \in lp_i} \{C_j^h\} = 0$ *if* $lp_i = \emptyset$ *and* $1_\alpha = 1$ *if* $lp_i \neq \emptyset$ *and* 0 *otherwise.*

Proof: By recurrence on the number of nodes visited. On the first node visited, Property 1 is true. Assuming that Property 1 is true at rank h. We prove it at rank $h+1$. Let us consider packet m of flow τ_i generated at time t. Due to the non-preemption, on any node $h \in (first_i, last_i]$, a packet m' belonging to a flow τ_j, $j \in lp_i$, can delay the execution of m if m arrives on node h while m' is being processed. Then, we have to distinguish three cases:

- Node h is the first node of $\mathcal{P}_i$ visited by flow τ_j ($first_{j,i} = h$). Hence, the maximum delay incurred by m directly due to flow τ_j meets: $C_j^h - 1$;
- Node h is not the first node of $\mathcal{P}_i$ visited by flow τ_j ($h \in (first_{j,i}, last_{j,i}]$) and $first_{j,i} \neq first_{i,j}$. Hence, the maximum delay incurred by m directly due to flow τ_j meets: $C_j^h - 1$;
- Node h is not the first node of $\mathcal{P}_i$ visited by flow τ_j ($h \in (first_{j,i}, last_{j,i}]$) and $first_{j,i} = first_{i,j}$. Packet m' leaves node $pre_i(h)$ at the latest at time $W_{i,t}^{pre_i(h)}$. Then, m' ends its processing on node h at the latest at time $W_{i,t}^{pre_i(h)} + Lmax + C_j^h$. As packet m arrives on node h at the earliest at time $W_{i,t}^{pre_i(h)} + C_i^{pre_i(h)} + Lmin$, the maximum delay incurred by m directly due to flow τ_j meets: $\max\left(0\,; C_j^h - C_i^{pre_i(h)} + Lmax - Lmin\right)$.

Moreover, $C_j^h \leq \max_{j \in lp_i} \{C_j^h\}$. Hence the property. ■

4.4 Latest starting time expression

From the previous two subsections, we can express the latest starting time of packet m on its last visited node.

Property 2 *Let m be the packet of flow τ_i generated at time t. When flows are scheduled FP/FIFO, the latest starting time of packet m on its last node visited, denoted $W_{i,t}^{last_i}$, is bounded by:*

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

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

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

with: $A_{i,j} = Smax_j^{first_{i,j}} - M_i^{first_{i,j}} + J_j$.

Proof: By Lemma 2 and Property 1. ■

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

$$\begin{cases} \mathcal{W}_{i,t}^{h\,(0)} = \sum\limits_{j \in hp_i \cup sp_i} C_j^{slow_{j,i}^h} + \left(1 + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow_i^h} \\ + \sum\limits_{\substack{k \in \mathcal{P}_i^h \\ k \neq slow_i^h}} \max\limits_{\substack{j \in hp_i \cup sp_i \cup \{i\} \\ first_{j,i}^h = first_{i,j}^h}} \{C_j^k\} - C_i^h + \delta_i^h + (|\mathcal{P}_i^h| - 1) \cdot Lmax \\ \\ \mathcal{W}_{i,t}^{h\,(p+1)} = \sum\limits_{\substack{j \in hp_i \\ last_{i,j}^h = h}} \left(1 + \left\lfloor \frac{\mathcal{W}_{i,t}^{h\,(p)} - Smin_j^h + A_{i,j}^h}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow_{j,i}^h} \\ + \sum\limits_{\substack{j \in hp_i \\ last_{i,j}^h \neq h}} \left(1 + \left\lfloor \frac{\mathcal{W}_{i,t}^{last_{i,j}^h} - Smin_j^{last_{i,j}^h} + A_{i,j}^h}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow_{j,i}^h} \\ + \sum\limits_{j \in sp_i \cup \{i\}} \left(1 + \left\lfloor \frac{t + Smax_i^{first_{j,i}^h} - Smin_j^{first_{j,i}^h} + A_{i,j}^h}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow_{j,i}^h} \\ + \sum\limits_{\substack{k \in \mathcal{P}_i^h \\ k \neq slow_i^h}} \max\limits_{\substack{j \in hp_i \cup sp_i \cup \{i\} \\ first_{j,i}^h = first_{i,j}^h}} \{C_j^k\} - C_i^h + \delta_i^h + (|\mathcal{P}_i^h| - 1) \cdot Lmax \end{cases}$$

with:

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

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

4.5 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 $\mathcal{P}_i$. When flows are scheduled FP/FIFO, we have 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: We consider the series $\mathcal{W}_{i,t}^{last_i}$ and prove this lemma by induction.

Step 1 The lemma is proved on node $first_i$. As $\forall (a,b) \in {\mathbb{R}^+}^2$, $\lfloor a+b \rfloor \leq \lfloor a \rfloor + \lceil b \rceil$, we have, for any time $t \geq -J_i$, $\mathcal{W}_{i,t+\mathcal{B}_i^{slow}}^{first_i\,(0)}$ equal to:

$$\sum_{j \in hp_i \cup sp_i} C_j^{first_i}$$

$$+ \left(1 + \left\lfloor \frac{t + \mathcal{B}_i^{slow} + J_i}{T_i} \right\rfloor\right) \cdot C_i^{first_i} - C_i^{first_i} + \delta_i^{first_i}$$

$$\leq \sum_{j \in hp_i \cup sp_i} C_j^{first_i} + \left(1 + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor\right) \cdot C_i^{first_i}$$

$$+ \left\lceil \frac{\mathcal{B}_i^{slow}}{T_i} \right\rceil \cdot C_i^{first_i} - C_i^{first_i} + \delta_i^{first_i}$$

$$\leq \mathcal{W}_{i,t}^{first_i\,(0)} + \mathcal{B}_i^{slow}.$$

We now show that if the property is true at rank p, then it is true at rank $p+1$. Indeed, for any time $t \geq -J_i$, we have $\mathcal{W}_{i,t+\mathcal{B}_i^{slow}}^{first_i\,(p+1)}$ equal to:

$$\sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_{i,t+\mathcal{B}_i^{slow}}^{first_i\,(p)} - Smin_j^{first_i} + A_{i,j}^{first_i}}{T_j} \right\rfloor\right)^+ \cdot C_j^{first_i}$$

$$+ \sum_{j \in sp_i \cup \{i\}} \left(1 + \left\lfloor \frac{t + \mathcal{B}_i^{slow} - Smin_j^{first_i} + A_{i,j}^{first_i}}{T_j} \right\rfloor\right)^+ \cdot C_j^{first_i}$$

$$- C_i^{first_i} + \delta_i^{first_i}$$

$$\leq \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_{i,t}^{first_i\,(p)} - Smin_j^{first_i} + A_{i,j}^{first_i}}{T_j} \right\rfloor\right)^+ \cdot C_j^{first_i}$$

$$+ \sum_{j \in sp_i \cup \{i\}} \left(1 + \left\lfloor \frac{t - Smin_j^{first_i} + A_{i,j}^{first_i}}{T_j} \right\rfloor\right)^+ \cdot C_j^{first_i}$$

$$+ \sum_{j \in hp_i \cup sp_i \cup \{i\}} \left\lceil \frac{\mathcal{B}_i^{slow}}{T_i} \right\rceil \cdot C_i^{first_i} - C_i^{first_i} + \delta_i^{first_i}$$

$$\leq \mathcal{W}_{i,t}^{first_i\,(p)} + \mathcal{B}_i^{slow}.$$

Hence, $\mathcal{W}_{i,t+\mathcal{B}_i^{slow}}^{first_i} \leq \mathcal{W}_{i,t}^{first_i} + \mathcal{B}_i^{slow}$.

Step 2 By assuming that property is true on any node visited between nodes $first_i$ and $pre_i(h)$, with $h \in \mathcal{P}_i$, we can prove in the same way as Step 1 that:
$\mathcal{W}_{i,t+\mathcal{B}_i^{slow}}^h \leq \mathcal{W}_{i,t}^h + \mathcal{B}_i^{slow}$. ■

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

Property 3 *When flows are scheduled FP/FIFO, the worst case end-to-end response time of any flow τ_i is bounded by:*

$R_i = \max_{-J_i \le t < -J_i + \mathcal{B}_i^{slow}} \{W_{i,t}^{last_i} + C_i^{last_i} - t\}$, *with:*

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

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

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

$A_{i,j} = Smax_j^{first_{i,j}} - M_i^{first_{i,j}} + J_j$ *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 2 and Lemma 3. ∎

4.6 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 P_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 P_i. We compute the set $\mathcal{S}_i$ of flows crossing directly or indirectly τ_i and apply Property 3 to compute the worst case response time of τ_i. More formally, we proceed as follows to determine $\mathcal{S}_i$:

- $\mathcal{S}_i = \{\tau_i\}$;
- $\mathcal{S}_i = \mathcal{S}_i \cup \{\tau_j, j \in hp_i \cup sp_i, \tau_j$ 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 P_i.

5 Example

In this section, we give an example of bounds on the end-to-end response times of sporadic flows, when these flows are scheduled according to FP/FIFO. We assume that the network meets: $Lmax = Lmin = 1$. Moreover, we consider the set $\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$. All these flows have a period equal to 36 and enter the network without jitter. The maximum processing time of any packet of flow τ_i on node $h \in \mathcal{P}_i$ is assumed to be equal to 4. Table 1 gives the fixed priority and the end-to-end deadline of each flow.

Table 1. Priorities and end-to-end deadlines

	τ_1	τ_2	τ_3	τ_4	τ_5
P_i	10	10	11	11	12
D_i	36	36	54	54	45

The path taken by each flow is defined as follows:

- $\mathcal{P}_1 = \{1, 3, 4, 5\}$
- $\mathcal{P}_2 = \{9, 10, 7, 6\}$
- $\mathcal{P}_3 = \{2, 3, 4, 7, 10, 11\}$
- $\mathcal{P}_4 = \{2, 3, 4, 7, 10, 11\}$
- $\mathcal{P}_5 = \{2, 3, 4, 7, 8\}$.

Applying Property 3, we obtain Table 2 giving the worst case end-to-end response time of any flow τ_i, $i \in [1, 5]$. We notice that each flow meets its end-to-end deadline.

Table 2. End-to-end response times of sporadic flows

	τ_1	τ_2	τ_3	τ_4	τ_5
R_i	31	31	46	46	33

6 Conclusion

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 compute worst case response time of flows scheduled with non-preemptive FP/FIFO. A possible application of these results lies in the deterministic quantitative guarantee provided to the EF class in a DiffServ architecture, assuming that the routes assigned to EF flows remain fixed, once assigned.

References

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

[2] J. Liu, *Real-time systems*, Prentice Hall, New Jersey, 2000.

[3] S. Martin, P. Minet, L. George, *Improving Fixed Priority schedulability with Dynamic Priority as secondary criterion*, Journal of Embedded Computing, to appear in 2005.

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

[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 2873, 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.

0-7695-2552-0/06 $20.00 © 2006 IEEE