Performance Analysis of Buffers Shared by Independent Periodic Tasks

J. Legrand, F. Singhoff, L. Nana, L. Marcé
LISYC/EA 3883, University of Brest (France)
20, av Le Gorgeu
CS 93837, 29238 Brest Cedex 3
{jlegrand,singhoff,nana,marce}@univ-brest.fr

Abstract

This paper presents a performance analysis of buffers shared by real time tasks. We focus on systems composed of several uniprocessor systems connected by a network. Each of these systems owns buffers and runs periodic tasks. Messages arriving from the network are stored in buffers and are consumed by a periodic task scheduled according to a scheduler like RM (Rate Monotonic) or EDF (Earliest Deadline First). No assumption is done on the rate at which messages are delivered to the uniprocessor system. From queueing system models, we provide performance analysis of buffers. We propose a new service time distribution, the P distribution. This service time distribution models the scheduling of a set of independent periodic tasks. From these propositions, a worst case and an average case performance analysis is given.

1 Introduction

In this paper, we present a performance analysis of buffers shared by real time tasks. We study real time systems composed of uniprocessor systems connected by a network. Each processor hosts an application running a set of real time tasks accessing buffers. Two kinds of buffers are distinguished :

- Buffers receiving messages sent by remote applications.
- Buffers receiving messages sent by tasks running on the same uniprocessor system.

In any cases, **messages are consumed by a single periodic task running on the same uniprocessor system**. We assume consumer tasks are not activated on message arrivals. They are independent and read messages from buffers at their own rate : a consumer task can be awoken even if the buffer is empty. This model of buffer can be applied on time-triggered/polling based systems where data are periodically read [KB02].

Tasks are scheduled according to a preemptive scheduler such as Rate Monotonic or Earliest Deadline First [LL73]. Tasks are defined by a capacity C_i, a period P_i and a deadline D_i. The capacity C_i is a bound on the task execution time. The period P_i is the fixed delay between two activations of a task. Tasks have to meet temporal constraints : tasks execution must be ended before the deadline D_i. Message consumptions and productions by periodic tasks are assumed to be instantaneous events. Only the dates of these events are considered in the sequel.

This paper deals with our analysis work on such system. By analysis, we mean checking that task deadlines will be met and that sier of buffers will be sufficient to avoid overflow. The real time scheduling theory offers interesting feasibility tests, such as bounds on processor utilization factor [LL73] or task response times [JP86, ABRT93] to check deadlines of **independent tasks**. Unfortunately, few buffer performance analysis results exist for this kind of real time system (eg. message waiting time, number of messages in the buffer,...) [Kre00].

The queueing system theory makes it possible to study performance of a system composed of servers, customers and storage places [Kle75b] : people waiting in a room for a doctor, network switch routing data, ... If new customers arrive in the system when a server is busy, their requests are stored in a queue. By defining the average rate of customer arrivals and the average rate of requests that the server can handle, the queueing system theory allows the designer to predict the average number of customers, the average customer waiting time, and the probability of having a given number of customers in the queue.

Different customers inter-arrival time distributions and service time distributions exist. The most usuals are deterministic (D), markovian (M) and general (G). D means constant delay between two customers arrivals or between two customers service times. M describes a customer arrival rate or a service time where delays follow an exponential probability distribution. Finally, if no assumption on the probability distribution is done, G is used. G is defined by an average rate and its variance.

Following the Kendall notation, a queueing system is described by at least 3 parameters : $a|b|c$. The a parameter is the customer arrival rate. b describes the service time rate. Finally, c is the number of servers. For instance, a system with one server, with a constant service time and an exponential client arrival is an M/D/1 queueing system.

Even if buffers are common operating system functionnalities, it seems that few results exist about buffer performance analysis when periodic tasks are scheduled according to a real time scheduler [TZ99].

In queueing system theory, results for similar systems exist. In priority queueing, a priority can be given to customers [AVS02, Sta92]. The most common priority queue is the HOL[1] queue where priorities are fixed [Kle75a].

Chen proposes mean waiting time for real time traffic with deadline constraints [CD97]. This work is based on non preemptive M/G/1 and work-conserving queue. Each real time traffic is a customer with a priority given by the Earliest Deadline First policy.

The real time queueing theory aims at using priority queueing in order to check temporal constraints of tasks randomly activated under "heavy traffic" [Leh96].

None of these approaches suits to the systems we study in this paper. Indeed, these approaches assign priorities to customers. Futhermore, they can not handle the fact that task can be awoken even if no message is stored in buffers.

A periodic server can be found in the queueing system theory. Such a queueing system is composed of some queues cyclically served by a single server [SLF92]. Except the periodic behavior of the server, the service time distribution does not handle the fact that task reponse time can be variable due to the real time scheduler.

Several works on queueing system have been lead in the real time community. A lot of queue service disciplines have been studied in the communication field [ZF94]. These services generally aim at providing bandwidth, end-to-end determinist or statistic guarantee on delays. Unfortunately, to avoid buffer overflow, service policies usually proposed in this context have a behavior which depends on the number of messages in the buffer.

To study buffers shared by independent periodic tasks scheduled according to a scheduler like RM or EDF, we propose a new service time distribution : the P distribution. **The P distribution models the fact that periodic tasks are scheduled with a real time scheduler**. The P distribution assumes that task wake up times are not synchronized with message arrivals. **Task dependencies lead to more complex feasibility tests to check task deadlines [SSNB95, Bla76, CSB90].** With the P distribution, scheduling feasibility tests for independent tasks can be applied to check task deadlines (eg. task reponse times with a Rate Monotonic scheduler [JP86, ABRT93]).

From the P distribution, two new queueing systems are defined : P/P/1 and M/P/1. An exact resolution of P/P/1 is given and we provide an approximation of the M/P/1. This approximation is based on a M/G/1 queueing model. **Applications sharing buffers can then be studied by both worst case and average case analysis. Worst case analysis can be performed if assumptions are made on message arrival rate. In this case, the system is checked with P/P/1 assuming that a smallest period of message arrivals rate exists. Otherwise, if no worst case assumption is made, we show that average analysis can be realized with M/P/1.**

This paper is organized as follows. In section 2, we describe the P service time distribution. From this distribution, we describe an average case analysis and a worst case analysis in sections 3 and 4. Some simulation performances which show the correctness of the model are provided in section 5. Finally, we conclude and give future works in section 6.

2 The P service time

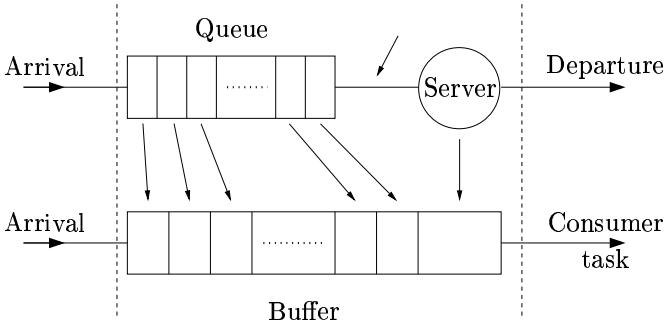

Figure 2: Buffer modelization

Buffers are modeled using queueing systems (see Figure 2). Customers are messages stored into buffers. The buffer state is modeled by the server and the queue state. Message departure dates in the queueing system are equal to consumption dates of consumer task. Thus, message waiting time in the buffer is equal to message waiting time in thequeue and in the server (this is also the case for the number of message in the buffer).

We propose a P distribution which describes the periodic behavior of a consumer task scheduled according to a real time scheduler.

In the sequel, a producer or a consumer task i will be defined by a period (denoted by P_i), a deadline (denoted by D_i) and a response time per activation.

[1] Head Of Line

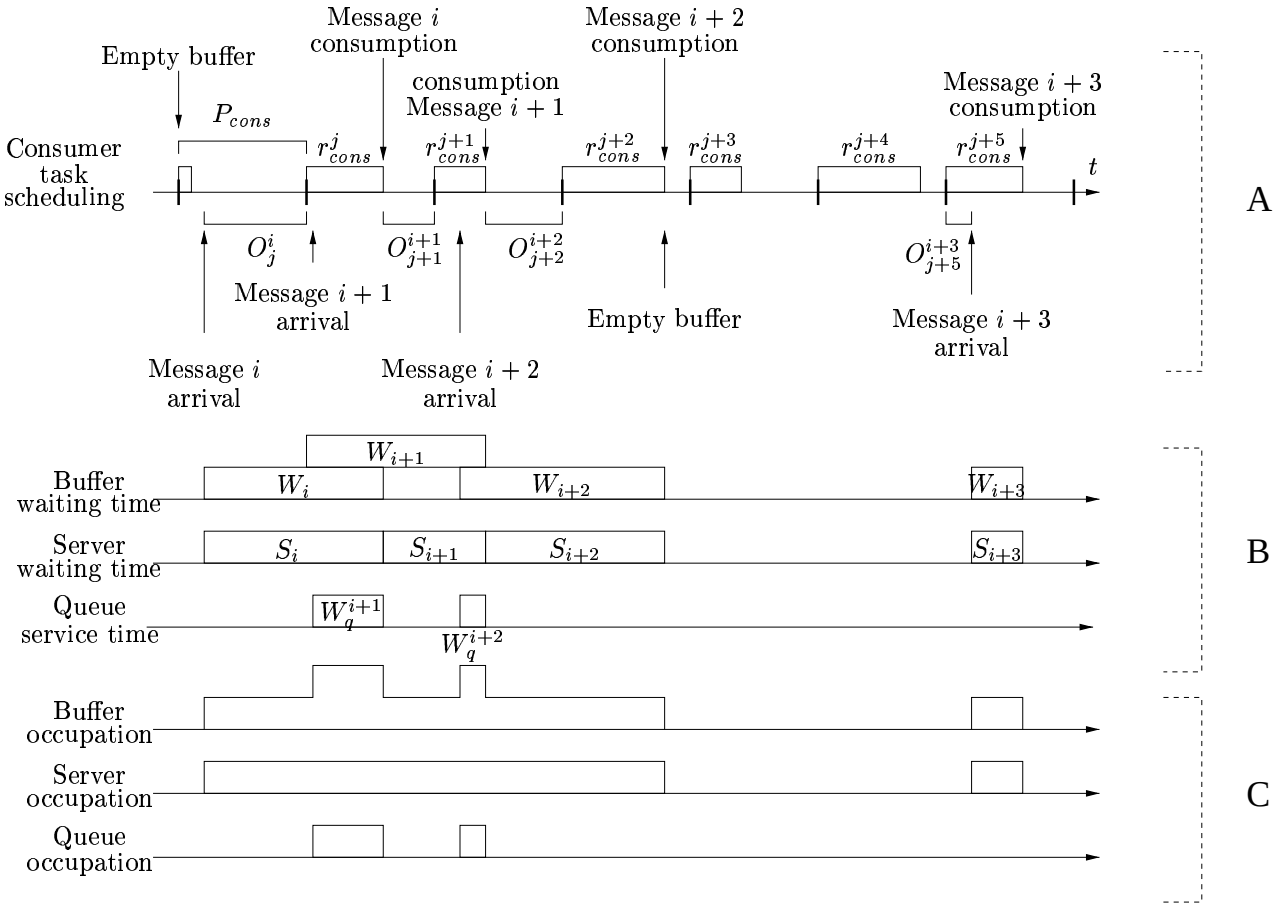

Figure 1: The P service time

In order to simplify explanation and results expression, we consider that one message is produced or consumed per task periodic activation. Furthermore, a message is consumed or produced r_i^j units of time. r_i^j is the response time of the j^{th} periodic activation of the task i.

First of all, we give characteristics of our P service time distribution. Let's recall the service time definition of a queueing server :

Definition 1 (Service time) *Service time S_i is the time spent by a message i in the server [Kle75b]. Then, S_i is the time between the server activation and the end of a customer computation.*

If the system is empty, the server is activated when a new customer arrives.

If customers are waiting in the queue, the server is activated just after the end of the previous service.

One can apply the Definition 1 to our buffers :

Definition 2 (P service time) *P service time is equal to the delay between the message arrival date in the server and the message consumption date.*

From Definition 2, an equation of S_i can be obtained. On Figure 1, one can find three groups of chronograms :

- Chronograms A illustrate the consumer task *cons* scheduling. P_{cons} is the consumer task period and r^j_{cons} is the response time of the j^{th} consumer task activation. r^j_{cons} is also the delay between the j^{th} consumer task activation and the message consumption date. The O^i_j parameter represents the delay between the message i arrival date in the server and the j^{th} consumer task activation date.

- Chronograms B illustrate the buffer waiting time W_i, the queueing system waiting time W^i_q and the service time S_i of message i. Their values are based on consumer task scheduling. For instance, from Definition 2, S_i is equal to $O^i_j + r^j_{cons}$.

- Chronograms C give buffer, queue and server occupation rate.

From Figure 1, let's see now an example of message arrival (messages i and $i+1$) :

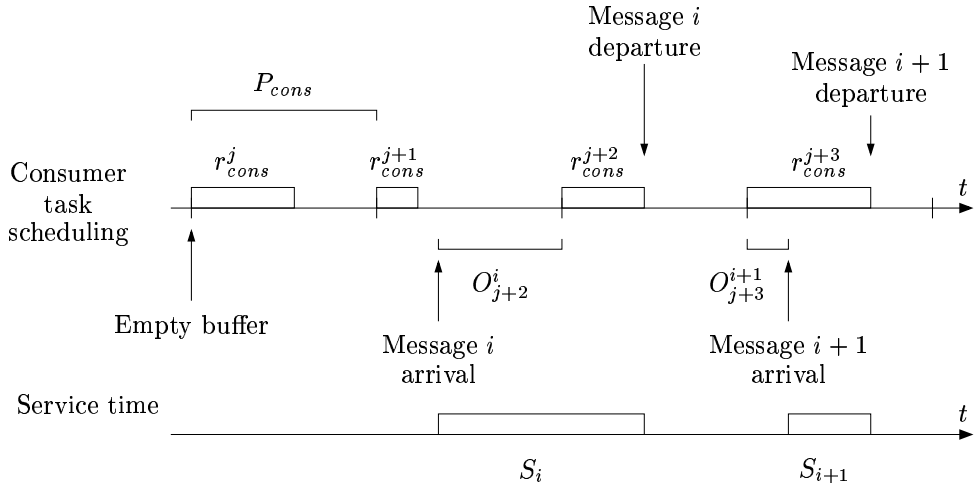

Figure 3: Service time with an idle server

1. Initially, the queueing system is empty. Message i is then immediately served. Message i waiting time in the queue W^i_q is equal to 0s. The queue is empty and the server occupation (current number of client) is equal to one.
2. Unlike the message i, when message $i+1$ arrives in the queueing system, the server is busy. Thus, message $i+1$ is stored in the queue during W^{i+1}_q units of time until the server is becoming idle again. There is 1 message in the server and 1 in the queue.
3. When message i is consumed, message $i+1$ starts to be served by the server. Occupation of the server is still equal to 1 and the queue becomes empty.
4. ...

From this example, one can have a delay between a message arrival and the periodic activation of the consumer. Then, on the one hand, a message is not necessarily consumed at each activation of the consumer task. On the other hand, a consumer can be activated when the buffer is empty. In the last case, the consumer just continues its execution. From Definition 2, the service time of the P distribution can be defined as follows :

Theorem 1 *When the server j is busy, its i^{th} service time is equal to :*

$$S_i^{'} = r^j_{cons} + P_{cons} - r^{j-1}_{cons}$$

In the opposite case, S_i is equal to :

$$S_i^{''} = r^j_{cons} + O^i_j$$

O^i_j belongs to the interval $[-r^j_{cons}; P_{cons} - r^{j-1}_{cons}]$.

Proof : Indeed, service time S_i is composed of two parts : O^i_j and r^j_{cons}.

When the buffer is empty, message i is immediately handled by the server. From Figure 3, one can see two cases : the message i arrives **before** or **after** the j^{th} consumer task activation.

When more than one message is waiting, the server handles the next message as soon as the precedent service is finished (see Definition 1). From Definition 2, the message entrance date in the server corresponds to the consumption date of the precedent message.

In the case of Figure 4, O^i_j is equal to $P_{cons} - r^{j-1}_{cons}$. Thus, service time is equal to $P_{cons} - r^{j-1}_{cons}$ plus r^j_{cons}. □

3 Buffer average performance analysis

Let's suppose now that messages arrive in the system at a random rate.

In the sequel, according to the Kendall notation, a buffer receiving random rate messages and shared by 1 consumer task, will be modeled with a M/P/1 queueing system [Kle75b, Rob90]. Messages are served in a FIFO manner : the earlier a message arrives, the earlier it is served.

We propose an approximation of the M/P/1 queueing system. This M/P/1 approximation consists in evaluating its average service time W_s and its variance σ_s^2. With W_s and σ_s^2, a M/P/1 queueing system can be modeled with a M/G/1 queueing system. Then, M/P/1 message waiting time and number of messages in the buffer can be computed with the following M/G/1 equations [Kle75b] :

$$W = W_s + \frac{\lambda(W_s^2 + \sigma_s^2)}{2(1-\rho)}$$

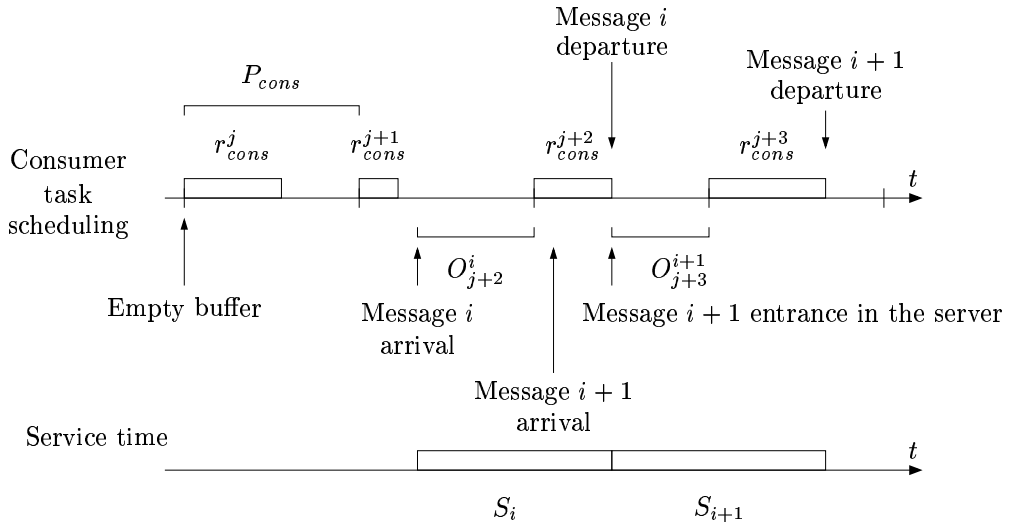

Figure 4: Service time with a busy server

$$L = \lambda W_s + \frac{\lambda^2 (W_s^2 + \sigma_s^2)}{2(1-\rho)}$$

where ρ is the queueing utilization factor and λ, the message arrival rate.

3.1 Proposition of a M/P/1 model

In this section, equations of average service time and variance service time are given. The proof is organized in three steps :

- First, M/P/1 queueing system service time is studied when ρ tends to 1.
- Second, M/P/1 queueing system service time is studied when ρ tends to 0.
- Finally, a linear regression is applied in order to compute average and variance service times which are valid for all ρ values.

To study W_s for the M/P/1 queueing system, we first have to define what an "actual" or "unactual" consumption is.

Due to the P service time distribution characteristics, a consumer can be activated whatever the number of messages stored in the buffer is. Figure 5 shows an actual and an unactual consumption. An "unactual consumption" occurs when a consumer is awoken when the buffer is empty. Otherwise, the consumption is said to be "actual".

Theorem 2 *The actual consumption rate U_c for a M/P/1 queueing system is equal to :*

$$U_c = 1 - P_0 = \rho$$

Where ρ is the utilization factor of the queue and P_0 is the probability of having no message in the buffer.

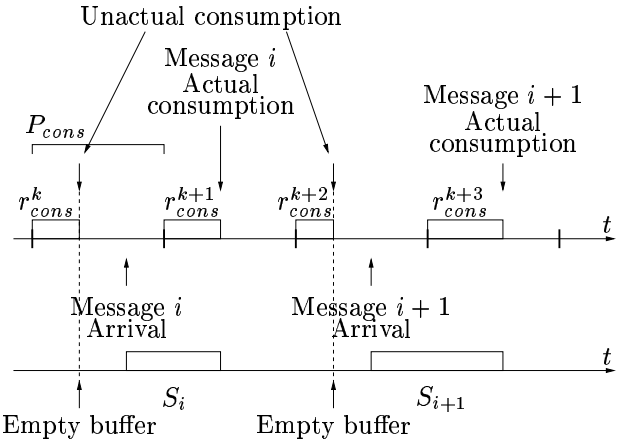

Figure 5: "Actual" versus "unactual" consumptions

Proof : Indeed, a consumption is actual when the buffer is not empty. The actual consumption rate U_c is equal to $1 - P_0$. Because for a G/G/1 queue, $\rho = 1 - P_0$ [Kle75b], U_c is equal to ρ. □

Since actual and unactual consumptions are defined, let's investigate on the mean service time. The usual mean service time equation is :

$$W_s = \frac{1}{n} \sum_{i=1}^{n} S_i$$

where n is the number of "actual" consumptions.

Unfortunately, for a given task activation i, it's difficult to know if the associated consumption will be actual or not. Then, it's difficult to evaluate if a given task activation will imply a request to the server.

We study two particular cases of the M/P/1 service time. First, we solve the case where the server is requested at each task activation (when ρ tends to 1, and then U_c tends to 1). Second, we propose an approximation when ρ tends to 0. In this case, only some task activations imply server requests. Finally, with a linear regression between these two cases, we give an approximation of M/P/1 mean service time and variance for all ρ values.

When ρ tends to 1, the behavior of the M/P/1 queueing system tends to be a deterministic one. We have the following mean service time :

Theorem 3 *When ρ tends to 1, the service time $S_i^{'}$ is equal to :*

$$S_i^{'} = r_{cons}^{i} + P_{cons} - r_{cons}^{i-1}$$

where cons is the consumer task of the M/P/1 queueing system. The average service time and its variance are then equal to :

$$W_s = P_{cons}$$

$$\sigma_s^2 = \frac{1}{n}\sum_{i=1}^{n} S_i^{'2} - W_s^2$$

Proof : Indeed, when ρ tends to 1, all consumptions are actual (see Theorem 2).

If we study the queueing system during an infinite time interval, we have an infinite sequence of S_i. Due to the real time scheduling of the consumer task, from one base period[2] to another one, the sequence of task response times is repeated.

Thus, if we consider nc base periods of n consumer activations, we have :

- $nc * S_1^{'} = nc * (r_{cons}^{1} + P_{cons} - r_{cons}^{n})$
- $nc * S_2^{'} = nc * (r_{cons}^{2} + P_{cons} - r_{cons}^{1})$
- $nc * S_3^{'} = nc * (r_{cons}^{3} + P_{cons} - r_{cons}^{2})$
- ...
- $nc * S_{n-1}^{'} = nc * (r_{cons}^{n-1} + P_{cons} - r_{cons}^{n-2})$
- $nc * S_n^{'} = nc * (r_{cons}^{n} + P_{cons} - r_{cons}^{n-1})$

[2]The base period (or Hyperperiod) is a cycle such as the pattern of arrival of a periodic tasks set recurs similarly [LM80].

Finally, the average of all service times is equal to :

$$W_s = \frac{1}{nc.n} nc.n.P_{cons} = P_{cons}$$

The variance is computed on the same n service time samples. σ_s^2 is then equal to :

$$\sigma_s^2 = \frac{1}{n}\sum_{i=1}^{n} S_i^{'2} - W_s^2$$

□

Let's see now the case where ρ tends to 0.

Theorem 4 *When ρ tends to 0, the service time is equal to :*

$$S_i^{''} = \frac{S_i^{'}}{2}$$

The average service time and its variance are then equal to :

$$W_s = \frac{P_{cons}}{2}$$

$$\sigma_s^2 = \frac{P_{cons}^2}{12}$$

Proof : when ρ tends to 0, the actual consumptions are rare (see Theorem 2). It is difficult to compute an exact value of the service time. Therefore, we propose an approximation of it.

The service time $S_i^{''}$ is equal to $O_i + r_{cons}^{i}$ (see Theorem 1). A first difficulty is to find an exact value of O_i. A second one is that the number of server requests (or S_i) is smaller than the number of task activations.

To solve these problems, we study the probability to have an interval of time t between 2 arrivals [Rob90, Kle75b] :

$$f(\lambda, t) = 1 - e^{-\lambda t}$$

When ρ tends to 0 with a fixed value of average service time, λ tends to 0. In this case, the time t between 2 arrivals is infinite and for all values of t, the function $f(0,t)$ tends to 0.

The probability to have a particular time between 2 subsequent arrivals is the same whatever this time is. Then, the server request date is uniformely distributed in the interval $[0, S_i^{'}]$[3] :

$$S_i^{''} = \frac{S_i^{'} + 0}{2} = \frac{S_i^{'}}{2}$$

[3]$[0, S_i^{'}]$ is a time interval between two subsequent consumptions. The server will be necessarily requested during this time interval.

Moreover, if arrival dates are uniformely distributed on an infinite time interval, the number of activations nc is the same for all service times.
Thus, the average service time is equal to :

$$W_s = \frac{1}{n * nc} \sum_{i=1}^{n} \frac{nc * S_i^{'}}{2}$$

Or

$$W_s = \frac{1}{2n} \sum_{i=1}^{n} S_i^{'}$$

Finally, we have

$$W_s = \frac{P_{cons}}{2}$$

The variance is uniformely distributed on the interval $[0, P_{cons}]$, thus :

$$\sigma_s^2 = \frac{P_{cons}^2}{12}$$

□

Applying a linear regression to the cases studied above (when ρ tends to 0 and when ρ tends to 1), we propose a mean service time and its variance which are valid for all ρ values.

Theorem 5 *The M/P/1 service time is equal to :*

$$S_i = \rho S_i^{'} + (1-\rho).S_i^{''} = (1+\rho).S_i^{'}$$

The average service time is then equal to :

$$W_s = \frac{P_{cons}}{2}(1+\rho) = \frac{P_{cons}}{2(1-\lambda\frac{P_{cons}}{2})}$$

And the variance of the average service time is :

$$\sigma_s^2 = \rho.\left(\frac{1}{n}\sum_{i=1}^{n} S_i^{'} - W_s^2\right) + (1-\rho).\frac{P_{cons}^2}{12}$$

Where $\rho = \lambda W_s$ *and* $S_i^{'} = r_{cons}^{i} + P_{cons} - r_{cons}^{i-1}$.

Proof : This linear regression is based on the following fact : the probability for a message to arrive in the queue whereas the server is busy, resp. not busy, is ρ, resp. $1-\rho$ (see Theorem 2). Then, the weights associated to the service times $S_i^{'}$ and $S_i^{''}$ are respectively ρ and $1-\rho$. The service time S_i is then equal to :

$$S_i = \rho S_i^{'} + (1-\rho).S_i^{''}$$

Thus, the mean service time becomes :

$$W_s = \frac{1}{n}\sum_{i=1}^{n}\left((1-\rho)\frac{S_i^{'}}{2} + \rho S_i^{'}\right)$$

Or

$$W_s = \left(\frac{1-\rho}{2} + \rho\right)\frac{1}{n}\sum_{i=1}^{n} S_i^{'}$$

Since

$$\frac{1}{n}\sum_{i=1}^{n} S_i^{'} = P_{cons}$$

We have

$$W_s = \frac{P_{cons}}{2}(1+\rho)$$

If we change ρ to λW_s, we have :

$$W_s = \frac{P_{cons}}{2}(1+\lambda W_s)$$

And we deduce the final mean service time :

$$W_s = \frac{P_{cons}}{2(1-\lambda\frac{P_{cons}}{2})}$$

The same linear regression is applied to compute the variance. When ρ tends to 0, $\sigma_{s''}^2$ is equal to :

$$\sigma_{s''}^2 = \frac{P_{cons}^2}{12}$$

And when ρ tends to 1, $\sigma_{s'}^2$ is equal to :

$$\sigma_{s'}^2 = \frac{1}{n}\sum_{i=1}^{n} S_i^{'} - W_s^2$$

Thus,

$$\sigma_s^2 = \rho.\left(\frac{1}{n}\sum_{i=1}^{n} S_i^{'} - W_s^2\right) + (1-\rho).\frac{P_{cons}^2}{12}$$

□

4 Worst case buffer analysis

We now study a system where buffer productions and consumptions are assumed to be periodic.

According to the Kendall notation, a buffer shared by n periodic producer tasks and 1 periodic consumer task scheduled with a real time scheduler can be modeled with a P/P/1 queueing system. Messages are served in a FIFO manner.

Some similarities exist between this system and voice transmission service provided by the AAL1 layer of ATM networks [GK96]. In order to solve our P/P/1 queueing system, we apply results from this ATM layer.

In AAL1/ATM, a producer sends audio packets at a fixed rate $\frac{1}{d}$. This troughtput is expressed in cells per second, the protocol data unit of ATM networks. A bounded variable delay is required by each cell to go from the sender to the receiver. In an AAL1 communication service, the consumer should receive the cells at the same rate the producer sends them. Each received cell is then buffered during a sufficient amount of time to hide this variation transmission delay. In [GK96], it has been shown that the size of the buffer used to hide variable transmission delay is bounded by L_{max} as follows :

Theorem 6 *The maximum size of ATM AAL1 layer buffers is :*

$$L_{max} = \left\lceil \frac{W_{max}}{d} \right\rceil$$

Where W_{max} is the maximum delay a cell stays in the buffer. We call this delay the maximum memorization delay. This delay is also the maximum delay between two successive consumptions.

The systems we study are similar to the one described above and we can apply Theorem 6 to find bounds on buffers shared by real time scheduled periodic tasks. Let's now suppose that the messages arrival rate is bounded by a period, the smallest period between two successive arrivals. For a buffer shared by n periodic producers and one periodic consumer, the buffer bound is given by [LSN+03] :

Definition 3 *The P/P/1 maximum buffer size L_{max} is :*

$$L_{max} = \max_{\forall y \geq 0} \left(\sum_{prod \in PROD} \left\lceil \frac{W_{max} + O_{prod}}{P_{prod}} \right\rceil - y \right)$$

where $PROD$ is the set of producers, P_{prod} the period of the producer prod and O_{prod} the maximum delay between the wake up time of the consumer and the wake up time of the producer prod.

Definition 3 is based on the maximum memorization delay W_{max} which is defined as follows :

Definition 4 *The maximum waiting time of a message i is :*

$$W_{max} = (y+1).P_{cons} + D_{cons}$$

Where y is the number of messages present in the buffer before the message i is inserted.

From Definition 3 and for all possible values of y, one can be prove that for a buffer shared by one periodic consumer and n periodic producers, the buffer bound is :

Theorem 7 *For a P/P/1 buffer shared by an harmonic tasks set [4] and $\forall i : D_i \leq P_i$, the maximum buffer size and the maximum memorization delay are respectively :*

$$L_{max} = 2.n$$
$$and$$
$$W_{max} = 2.n.P_{cons}$$

For non harmonic tasks set, the maximum buffer size and the maximum memorization delay are respectively :

$$L_{max} = 2.n + 1$$
$$and$$
$$W_{max} = (2.n+1).P_{cons}$$

A proof of Theorem 7 is given in [LSN+03].

5 Simulations of M/P/1 and P/P/1 models

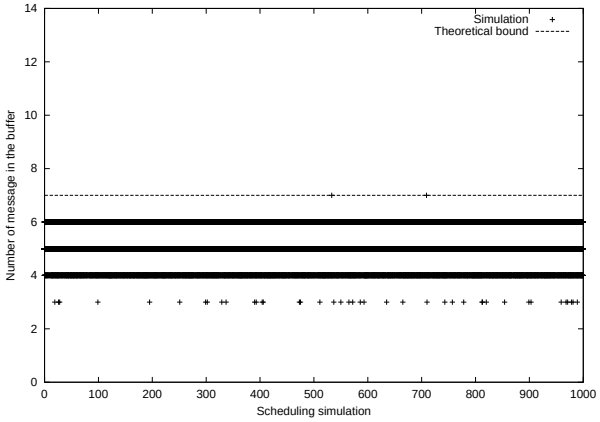

Figure 6: Worst case analysis : P/P/1 buffer bounds

Let's see now few simulations result which show the efficiency of the proposed performance models.

Simulations are performed with randomly generated systems composed of n producers and one consumer.

For the worst case analysis (P/P/1 queueing system), 1000 tasks sets with three periodic producers and one periodic consumer were generated. 1000 scheduling were randomly generated for each tasks set over 20 times the consumer period. No assumption is done concerning the real time scheduler. In Figure 6, the maximum buffer bound proposed in Theorem 7 is compared to simulation

[4] A tasks set is said to be harmonic if and only if each task period is a positive integer multiple of all smaller task periods.

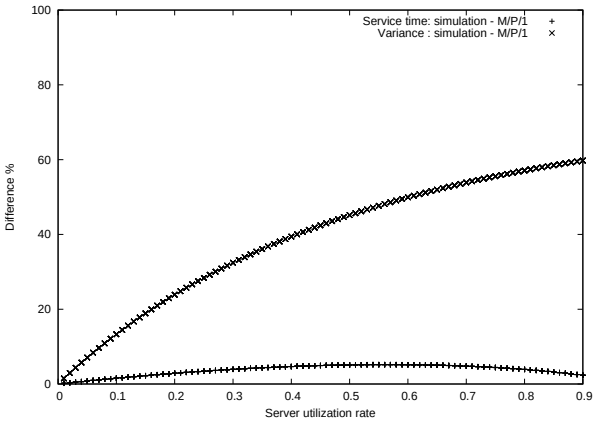

Figure 7: M/P/1 mean service time and variance with high priority tasks

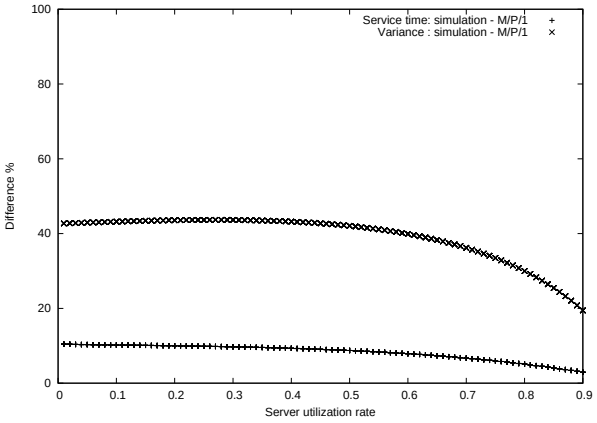

Figure 8: M/P/1 mean service time and variance with low priority tasks

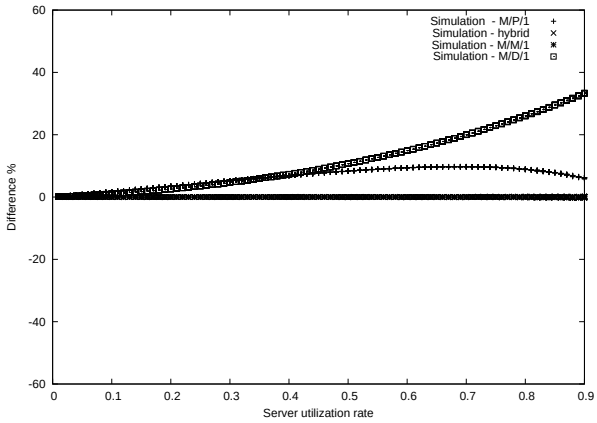

Figure 9: M/P/1 : number of messages in the buffer with high priority tasks

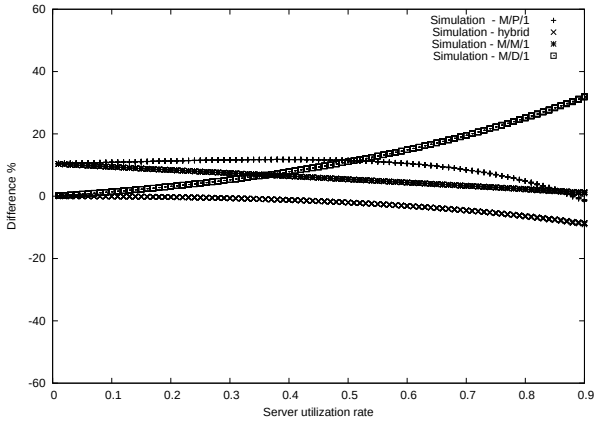

Figure 10: M/P/1 : number of messages in the buffer with low priority tasks

results. This Figure shows that the buffer bound is met but never overflowed.

For the average case analysis (M/P/1 queueing system), 990 systems composed of n message arrival flows and one periodic consumer task were generated. Simulations were done during 20000000000 units of time.

The main objective of those simulations is to test the mean service time, the variance and their impact on the average message waiting time and buffer utilization factor. A fixed priority scheduler is chosen for the simulations. The variance of the service time depends on the response time of the consumer task. Consequently, modifying the priority of the consumer tasks will modify the variance of the service time. We present graphs for both high and low fixed priority consumer tasks. Low priority consumer task response times were uniformely generated in $[0, P_{cons}]$. We assume that high priority tasks have fixed response time.

Simulations were done for different queueing utilization factors ($\rho \in [0.01, 0.99]$). Figures 7, 8, 9 and 10 show simulation and theoretical results of M/P/1, M/M/1 and M/D/1 queueing systems in order to compare their efficiency. On the horizontal axis, we have the queueing utilization factor. On the vertical axis, we have the gap between simulation results and theoretical results.

In order to calibrate the simulator correctness, we show the gap between M/P/1 simulation results and M/G/1 theoretical results computed with mean service time and variance resulting from simulations (see "hybrid results" graphs in Figures).

Figures 7 and 8 show the comparisons between M/P/1 theoretical and simulated mean service time and variance. The gap between theoretical and simulated mean service time is less than 8 percents. Unfortunately, it seems that theoretical mean service time variance does not match simulated results.

Figures 9 and 10 show the comparisons between M/P/1 theoretical and simulated number of messages in

W / L	High priority	Low priority
M/P/1 (M/G/1 with W_s from Theorem 5)	[0%,10%]	[0%,10%]
M/M/1 with $W_s = P_{cons}$	100%	[75%,100%]
M/M/1 with W_s from simulation	[0%,19%]	[0%,19%]
M/M/1 with W_s from Theorem 5	0%	[0%,10%]
M/D/1 with $W_s = P_{cons}$	[−100%,5%]	[−75%,5%]
M/D/1 with W_s from simulation	[0%,37%]	[0%,37%]
M/D/1 with W_s from Theorem 5	[0%,37%]	[0%,37%]

Table 1: Simulation results summary

the buffer. Despite of the variance results, the theoretical M/P/1 number of messages in the buffer match the simulated results (less than 10 percents). From Figure 9 and Figure 10 and the Little's result, one can deduce average message waiting time in buffers [Kle75b].

Table 1 summarizes the difference between proposed theoretical results and simulation results obtained during simulation.

6 Conclusion

This paper presents performance analysis of buffers shared by independent periodic tasks. Tasks are scheduled according to a real time scheduler like RM or EDF [LL73].

The analysis focuses on task deadlines and buffer performance such as the messages waiting times and number of messages in the buffer. The real time scheduling theory offers interesting feasibility tests to check deadlines of **independent tasks** [JP86, ABRT93]. Unfortunately, few results exist for buffer performance analysis [Kre00].

We propose a new service time distribution : the P distribution. The P distribution models the fact that periodic tasks are scheduled with a real time scheduler. It also assumes that task wake up times are not synchronized with message arrivals. Scheduling feasibility tests **designed for independent tasks** can then be applied to check task deadlines [JP86, ABRT93]. Our model of buffer can be applied on time-triggered/polling based systems where data are periodically read [KB02].

From the P distribution, two new queueing systems are presented : M/P/1 and P/P/1. Applications sharing buffers can then be studied by both worst and average cases analysis. for the worst case analysis, the system is checked with P/P/1 assuming that a smallest period of message arrivals rate exists. When no worst case assumption is done, we show that average analysis can be performed with M/P/1.

An exact resolution of P/P/1 is given and we provide an M/P/1 approximation. The M/P/1 queueing system approximation is based on the fact that, due to the real time scheduling, the system is mainly deterministic when the utilization factors are high and that, due to the message arrival rate, the system is mainly Markovian for low utilization factors.

The P/P/1 and M/P/1 proposed models have been simulated and tested. Simulations show that M/P/1 theoretical average service time is close by the one observed at simulation time. It is not the case for the theoretical variance on the mean service time. Despite of the variance results, the theoretical M/P/1 message waiting time and the theoretical M/P/1 number of messages in the buffer are better than the M/M/1 and M/D/1 ones for the simulated systems.

Future works will consider randomly activated tasks. Today, few results exist for checking temporal constraints of such tasks. We aim at providing feasibility tests and buffer analysis tools for such tasks running on the studied systems.

7 Acknowledgments

We would like to thank Gégardo Rubino (IRISA/INRIA Rennes) for his help on the work presented in this paper.

References

[ABRT93] A. N. Audsley, A. Burns, M. Richardson, and K. Tindell. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal*, pages 284–292, 1993.

[AVS02] K. E. Avrachenkov, N. O. Vilchensky, and G. L. Shevlyaokov. Priority queueing with finite buffer size and randomized push-out mechanism. Technical report, INRIA technical Report number 4434, March 2002.

[Bla76] J. Blazewicz. Scheduling Dependant Tasks with Different Arrival Times to Meet Deadlines. In. Gelende. H. Beilner (eds), Model-

ing and Performance Evaluation of Computer Systems, Amsterdam, Noth-Holland, 1976.

[CD97] K. Chen and L. Decreusefond. An Approximate Analysis of Waiting Time in Multiclasses M/G/1/./EDF Queues. 24(1), May 1997.

[CSB90] H. Chetto, M. Silly, and T. Bouchentouf. Dynamic Scheduling of Real-time Tasks Under Precedence Constraints . *Real Time Systems, The International Journal of Time-Critical Computing Systems*, 2(3):181–194, September 1990.

[GK96] M. Gagnaire and D. Kofman. *Réseaux Haut Débit : réseaux ATM, réseaux locaux, réseaux tout-optiques.* Masson-Inter Editions, Collection IIA, 1996.

[JP86] M. Joseph and P. Pandya. Finding Response Time in a Real-Time System. *Computer Journal*, 29(5):390–395, 1986.

[KB02] H. Kopetz and G. Bauer. The time-triggered architecture. *Proceedings of the IEEE Special Issue on Modeling and Design of Embedded Software*, October 2002.

[Kle75a] L. Kleinrock. *Queueing Systems : Computer Application.* Wiley-interscience, 1975.

[Kle75b] L. Kleinrock. *Queueing Systems : theory.* Wiley-interscience, 1975.

[Kre00] J. Kreimer. Real Time System with homogeneous servers and noidentical channels in steady-state. *Computers and Operations Research*, (29):1465–1473, December 2000.

[Leh96] J. P. Lehocsky. Real Time Queueing Theory. pages 186–194. Proceedings of the 17th IEEE Real-Time Systems Symposium (RTSS '96), Washington, DC, USA, December 1996.

[LL73] C. L. Liu and J. W. Layland. Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environnment. *Journal of the Association for Computing Machinery*, 20(1):46–61, January 1973.

[LM80] J.Y.T Leung and M.L. Merril. A note on preemptive scheduling of periodic real time tasks. *Information processing Letters*, 3(11):115–118, 1980.

[LSN+03] J. Legrand, F. Singhoff, L. Nana, L. Marcé, F. Dupont, and H. Hafidi. About Bounds of Buffers Shared by Periodic Tasks : the IRMA project. In the 15th Euromicro International Conference of Real Time Systems (WIP Session), Porto, July 2003.

[Rob90] T. G. Robertazzi. *Computer Networks and Systems : queueing theory and performance evaluation.* Springer-Verlag, 1990.

[SLF92] M. Sidi, H. Levy, and S. Fuhrmann. A Queueing Network with a Single Cyclically Roving Server. *Queueing systems, Theory and Applications*, 11:121–144, 1992.

[SSNB95] J. Stankovic, M. Spuri, M. Di Natale, and G. Buttazzo. Implications of Classical Scheduling Results For Real-Time Systems. *IEEE Computer*, 28(6):16–25, June 1995.

[Sta92] I. Stavrakakis. A Considerate Priority Queueing System with Guaranteed Policy Fairness. pages 2151–2159. In the proceedings of IEEE Infocom'92 Conference, Florence, May 1992.

[TZ99] P. Tsigas and Y. Zhang. Non-blocking data sharing in multiprocessor real-time systems. *RTCSA99*, 1999.

[ZF94] H. Zhang and D. Ferrari. Rate-Controlled Service Disciplines. *In journal of High Speed Netwokrs*, 4(3), 1994.