Towards efficient probabilistic scheduling guarantees for real-time systems subject to random errors and random bursts of errors

Michael Short
Electronics & Control Group,
Teesside University,
Middlesbrough, UK.
e-mail: m.short@tees.ac.uk

Julián Proenza
Systems, Robotics & Vision Research Group,
Universitat de les Illes Balears,
Palma de Mallorca, Spain.
e-mail: julian.proenza@uib.es

Abstract—Real-time computing and communication systems are often required to operate with prespecified levels of reliability in harsh environments, which may lead to the exposure of the system to random errors and random bursts of errors. The classical fault-tolerant schedulability analysis in such cases assumes a pseudo-periodic arrival of errors, and does not effectively capture any underlying randomness or burst characteristics. More modern approaches employ much richer stochastic error models to capture these behaviors, but this is at the expense of greatly increased complexity. In this paper, we develop a quantile-based approach to probabilistic schedulability analysis in a bid to improve efficiency whilst still retaining a rich stochastic error model capturing random errors and random bursts of errors. Our principal contribution is the derivation of a simple closed-form expression that tightly bounds the number of errors that a system must be able to tolerate at any time subsequent to its critical instant in order to achieve a specified level of reliability. We apply this technique to develop an efficient ‘one-shot’ schedulability analysis for a simple fault-tolerant EDF scheduler. The paper concludes that the proposed method is capable of giving efficient probabilistic scheduling guarantees, and may easily be coupled with more representative higher-level job failure models, giving rise to efficient analysis procedures for safety-critical fault-tolerant real-time systems.

Keywords—Probabilistic Schedulability Analysis; Error models; Fault-Tolerance.

I. INTRODUCTION AND MOTIVATION

Real-time computing and communication systems are often required to operate with a pre-specified level of reliability in harsh environments. In many cases they may be subject to environmental hazards such as electromagnetic interference (EMI) and other forms of radiation, and also to excessive mechanical/electrical stresses. Exposure to hazards such as this can induce random errors into a system, which – if left uncorrected – may result in system failures [1][2]. In this paper we are principally concerned with schedulability analysis of real-time CPU tasks and messages which are scheduled by some priority-driven algorithm in the presence of transient and/or intermittent errors.

Although transient and intermittent errors are by definition only of short duration, their effects still need to be taken into account in the analysis of real-time systems. For example, an error may arise due to a nearby lightning strike or other EMI fault, leading to an erroneous state and the abortion of an executing job by a CPU or a transmitting message frame in a network. For crucial systems, the use of fault-tolerance (mainly in the form of temporal redundancy) is required such that the aborted job or message can be re-executed or re-transmitted [3][4]. Such a form of redundancy requires some temporal ‘slack capacity’ in the schedule; how much slack is required to be allocated depends upon many factors including the level of criticality in the service the system provides, the task/message parameters and scheduling algorithms and the nature of the error detection and correction mechanisms employed by the system. If insufficient slack is employed by a system to tolerate the effects of the errors it experiences, then aborted jobs or message frames will not be processed or delivered correctly before their deadlines and system failures may occur. Clearly then, a major factor that needs to be considered in the design of fault-tolerant real-time systems is the frequency and severity of the transient errors the system is likely to experience. In addition to sporadic error arrivals which are purely random and uncorrelated in nature, research has shown that errors are very likely to occur in short transient bursts (see [5-9] and the references therein). Therefore a representative error model must have the ability to include these types of bursty behavior within its domain of operation.

Although some approaches to fault-tolerant schedulability analysis employ models which capture these behaviors, to date this has been at the expense of greatly increased analysis complexity. In this paper, we develop a quantile-based approach to probabilistic schedulability analysis in a bid to reduce complexity whilst still retaining a rich stochastic error model capturing random errors and random bursts of errors. Our principal contribution is the derivation of a simple closed-form expression that tightly bounds the number of errors that a system must be able to tolerate at any time subsequent to the Synchronous Arrival Sequence (SAS) of its tasks in order to achieve a specified level of reliability. At the core of the technique is the use of a Markov-Modulated Poisson Binomial (MMPB) process; the model is defined by some simple parameters with clear physical interpretations. Although the exact solution of the MMPB process initially seems to be too complex for practical use (especially in on-line situations), we develop and prove the correctness of a tight upper bound on the worst-case number of errors that the system must be able to
tolerate in a given time interval to achieve a specified reliability $R$. By way of example, we apply this bounding technique to develop an efficient ‘one-shot’ schedulability analysis for a simple fault-tolerant EDF scheduler, and conclude that the proposed method is capable of giving efficient probabilistic scheduling guarantees.

The remainder of the paper is organized as follows. Section II presents relevant background material and reviews related work in the area. In Sections III and IV we present the probabilistic error model and the main results of the paper. Section V proves some important properties of the model. Sections VI and VII investigate schedulability analysis and present an illustrative example, and Section VIII concludes the paper.

II. RELATED WORK

In this Section, we aim to give a brief (but not exhaustive) review of some related work on error models and fault-tolerant schedulability analysis. Classical error models employed in schedulability analysis typically assume either (i) a pseudo-periodic arrival of errors (e.g. [12][13][14]), or (ii) that some fixed number $k$ of errors will arrive in a burst and that one burst will be experienced over the major cycle of system operation (e.g. [6][7][8]). Although approach (i) is relatively straightforward to incorporate into an existing schedulability analysis, in many cases it does not effectively capture either randomness (with exceptions, e.g. [13]) or bursty characteristics. Approach (ii), on the other hand, is well suited to bursty characteristics; however the assumption that not more than one burst of errors will arrive over the entire major cycle of a system’s timeline seems to be inappropriate if the error generating process is purely stochastic in nature. In addition, if $k$ errors arrive in some time interval of length proportional to the smallest relative deadline of any task, then it again seems unjustified to assume that only $k$ errors will arrive in some proportionally much larger time interval (e.g. the major cycle). We note, however, that the higher-level job failure models presented in both [7] and [8] are very appealing in that they allow the effects of many higher-level fault-tolerance mechanisms (such as execution of backup jobs in case of primary job failures) to be modeled in a systematic way and efficiently incorporated in a schedulability analysis.

In the case of the error models mentioned above, and other similar works, to date there seems to have been relatively few attempts to relate the parameters of the low-level error models to the reliability of the resulting system. However, some notable extensions do exist. In [13], an upper bound for the probability of any two error arrivals violating a minimum inter arrival time requirement is developed that, when respected, makes the system schedulable. In [11] this result is extended to cases in which error bursts are possible. Although [11] principally concentrates upon providing probabilistic real time guarantees for the message scheduling in systems based on the Controller Area Network (CAN) protocol, it also includes some more general discussion and its results also apply to, for example, the execution of tasks in a uniprocessor (such as in [13]) and, very importantly, it addresses the occurrence of error bursts as well.

Indeed the analysis of distributed systems such as CAN has generally seen increased efforts to combine reliability and schedulability analysis, and the error models employed in these works are typically much richer than those employed for CPU schedulability analysis (although, as noted, the scope of these models is not restricted to the networked environment). For example, previous work has examined probabilistic (bursty/sporadic/intermittent) errors and also deterministic errors within combined reliability and schedulability analysis frameworks ([9][10][11]).

Among the most recently published related works, probably the one that is closest to ours is [11]. One complication of the methods employed in this paper, and of the use of probabilistic models in general, is the need to move away from the deterministic ‘yes/no’ one-shot analysis typical of CPU schedulability testing. Typically, an iterative procedure is needed to produce either a probability distribution of response times, or a breakdown reliability $R$ beyond which point one or more tasks or messages will miss a deadline. If $R$ is at an acceptable level (e.g. above that required for certification) then – from a schedulability perspective at least – the system can operate in the target environment. This complexity generally makes the use of probabilistic methods impractical where efficiency is required, e.g. for online admission controls in flexible automotive or wireless networks. In this paper, we aim to take a step towards the removal of some of this complexity by reducing tests for $R$-schedulability for an iterative procedure to a minimum overhead, one-shot ‘yes/no’ schedulability analysis.

III. PROBABILISTIC ERROR MODEL

A. Assumptions on the System Model

We assume a standard model for real-time computing and communication systems, in that the system to be implemented can be represented by a set $\Gamma$ of $n$ tasks (messages), denoted as $\Gamma=\{\tau_1, \tau_2, \ldots, \tau_n\}$. Each task is represented by a 3-tuple:

$$\tau_i = (T_i, C_i, D_i)$$  \hspace{1cm} (1)

in which $T_i$ is the task (message) period, $C_i$ is the worst-case execution (transmission) time of any instance of the task (message) and $D_i$ is the task (message) relative deadline. We assume that tight estimates of execution/transmission times are known. Each invocation of the task (message) is called a job (frame), and the $k^{th}$ invocation of task $i$ is denoted as $\tau_{i,k}$. When a job of task $i$ arrives (becomes ready) at some time $t$, its absolute deadline is set at time $t + D_i$ and the scheduling procedure must allocate $C_i$ units of CPU/network time to process the job in the interval $[t, t + D_i]$ else a deadline miss may occur. The utilization of an individual task is given by $u_i = C_i/T_i$ and represents the fraction of time the CPU (network) will be occupied processing the jobs (frames) generated from the task over its lifetime. Successive job arrivals from sporadic tasks are invoked by both internal and external events (typically hardware or software interrupts) and are always separated by at least $T_i$ units of time; job arrivals
from periodic tasks are always separated by exactly $T_i$ time units, and are invoked by a logical timer, which may be distributed over a number of network nodes in a distributed system. In the analysis that follows, we assume w.l.o.g. that time is discrete and is indexed by a non-negative integer variable $t$. Finally we assume (again w.l.o.g.) that the worst-case arrival pattern of the tasks and/or messages is the SAS, i.e. when they are all released simultaneously at $t = 0$ and thereafter arrive at their maximum allowed rates, see e.g. [3][4][15].

B. Preliminaries

We consider that at each discrete time step, only a single error may manifest with some specified probability. Although real-world error inter-arrivals are more likely to follow continuous distributions, if multiple errors arrive during the same logical system time step, then we consider only one of these errors will be ‘counted’ as such and cause the currently executing job (or transmitting frame) to deviate from its expected behavior. Errors in jobs or frames may or may not be detected immediately, but as this is principally defined by the higher-level (application-specific) error propagation model, taking into account such factors as the error detection and correction mechanisms employed by the system, we do not consider these elements to be an explicit part of the error model we present.

In our model, to incorporate the effects of multiple sources of uncorrelated and correlated (burst) errors we allow the probability at each time step(s) to take a distinct value, which can be a function of the probability of error occurrence at some previous time step(s). Therefore at a given time $t$, we have a series of possibly non-identical but independent Bernoulli indicator variables $x(i), 0 \leq i < t$, such that the probability of error occurrence (i.e. $p(x(i) = ‘1’)$) is given by a distinct probability $p(i), 0 \leq p(i) \leq 1$. Thus the summation of these $t$ non-identical Bernoulli indicator variables is itself a random variable counting the number of errors, and follows ‘Poisson’s Binomial’ distribution [16][17]. If each of the probabilities $p(i) = p = \text{constant}$, then the distribution becomes a regular Binomial. Given the application domain, we also assume that the required level of system reliability is given by a probability $R$ (note that usually $R \gg 0.9$). For the purposes of the error model, $R$ may be considered to be an adjustable level of confidence that should be placed upon the results of its predictions.

In terms of predictions, let the maximum number of errors we would expect to observe in any interval of time of length $t$ - for the given confidence probability $R$ - be returned by a function $\eta(t, R)$. The value obtained from $\eta(t, R)$ is the smallest non-negative integer $k$ such that the Cumulative Distribution Function (CDF) of Poisson’s Binomial distribution is greater than or equal to $R$, i.e.:

$$\eta(t, R) = \min\{k \in N: R \leq CDF(k, P)\} \tag{2}$$

where $P = \{p(0), p(1), \ldots, p(t-1)\}$ represents the vector of $t$ probabilities. In other words, we seek the $R$th quantile of the distribution. The authors are not aware of any methods to directly evaluate the quantile function, but since $k$ is integer and always lies in the range $[0, t]$ a binary search would require no more than $\log_2(t)$ evaluations of the CDF. The CDF for this distribution can be calculated using the formula [16][17][18]:

$$CDF(k, P) = \sum_{m=0}^{\lfloor t \rfloor} \sum_{A \subseteq \{1, 2, 3, \ldots, t\}} \prod_{i \in A} p(i) \prod_{i \notin A} (1 - p(i))$$

where $A_m$ is the set of all subsets of $m$ integers that can be selected from the set $\{1, 2, 3, \ldots, t\}$ and $A'$ is the complement of set $A$. The evaluation of the CDF itself - for a single value of $k$ - is non-trivial; direct evaluation of (3) is clearly not practical even for relatively small $t$. Several recursive methods with run-time $O(kt)$ for the evaluation of the CDF are known (several methods are described and investigated by Hong in [16]), and are practical for use with small to medium $t$. A method based upon a Discrete Fourier transform was recently developed [18], which exhibits complexity $O(t \log(t))$ – this is advantageous when $k \approx t$.

We say that a set of tasks is $R$-schedulable iff, for all $t > 0$, no deadlines will be missed by the system even when $\eta(t, R)$ errors have arrived by any given time $t$. For exact schedulability analysis of fault-tolerant real-time systems, many evaluations of $\eta(t, R)$ will clearly be required, sometimes for potentially very large values of $t$. This is due to the fact that for arbitrary-deadline systems scheduled using either static priority or dynamic priority scheduling techniques, the portion of the schedule that is required to be examined following the critical instant can be exponential (or at best pseudo-polynomial) in the task parameters [19][20]. In the case of on-line schedulability analysis, it is usual to perform a sufficient (c.f. exact) schedulability analysis in which some pessimism is allowed in order to reduce the implementation complexity [4][15]. In this latter case, although the number of checks to be performed is significantly reduced (typically to linear or log-linear complexity in $n$), $\eta(t, R)$ may still need to be evaluated for large values of $t$. Therefore exact evaluation of (2) does not seem well-suited to our purposes - for either on-line or off-line schedulability analysis - and we will instead seek to determine a provable upper bound for $\eta(t, R)$ that is tight enough to be of practical use yet efficient enough to be repeatedly computed in an efficient manner. Prior to presenting this bound, we will first discuss how the vector of probabilities $P$ is to be obtained for each of the individual $x(i)$ variables in our model.

C. Modeling Random Errors and Bursts of Errors

In order to evaluate $\eta(t, R)$ an efficient means to generate the probabilities $p(i)$ up to the time index $t$ is required. In order to be of practical use, an error model that is rich enough to capture random errors and bursts of random errors yet simple enough to lend itself to straightforward analysis is required. A common way to model bursty behavior is to use a simple two-state discrete Markov model of the Gilbert or Gilbert-Elliot types [5], such as is shown in Figure 1:
The model has two possible states $G$ and $B$, loosely representing ‘Good’ and ‘Bad’ (or ‘Burst’) states respectively. At any discrete time step $t$ the probability that the actual state $s(t)$ is ‘Bad’ or ‘Good’ is denoted as $p(s(t)=B)$ and $p(s(t)=G) = 1 - p(s(t)=B)$. Transitions between the two states $G$ and $B$ have associated with them the probabilities $p_{GB}$ and $p_{BG}$. The probability of remaining in a given state is then given by $p_{GG} = 1 - p_{GB}$ and $p_{BB} = 1 - p_{BG}$. Each state has also associated with it a probability of error arrival or ‘intensity’, denoted by $\lambda_G$ and $\lambda_B$. The model parameters $p_{GB}$ and $p_{BG}$ can be interpreted as follows: the reciprocal of $p_{GB}$ defines the expected (mean) gap between error bursts $\mu_{EG}$ and the reciprocal of $p_{BG}$ defines the expected (mean) duration of error bursts $\mu_{EB}$ both variables having a geometric distribution. The expected inter-arrival time of error bursts is then given as $(\mu_{EG} + \mu_{EB})$. The model parameters $\lambda_G$ and $\lambda_B$ can be interpreted as follows: when the system is in the ‘Good’ state, the reciprocal of $\lambda_G$ defines the expected (mean) inter-arrival time of errors, and when the system is in the ‘Bad’ state, the reciprocal of $\lambda_B$ defines the expected (mean) inter-arrival time of errors, both variables again having a geometric distribution. As the geometric distribution is the discrete equivalent to the continuous exponential distribution, this model would seem appropriate for use with both real-time communication and computing systems. The model parameters may be directly estimated from observed data, and statistical techniques can be applied to derive their values from larger empirical data sets [5]. If statistical techniques are employed, it would seem prudent to estimate the parameters in question to at least the same confidence level $R$ that is to be employed overall.

For ease of exposition, let the probabilistic state of the Markov model at time step $t$ be encoded as the variable $m(t) = p(s(t)=B)$, i.e. the probability that the link is in the error state. Applying the normal rules for Markov model state transitions, the probability that the link will be in the ‘Bad’ state at step $t+1$ depends only upon the probabilities $p_{BG}$ and $p_{GB}$ according to the recurrence relation:

$$m(t+1) = p_{BG} p(s(t)=B) + p_{GB} p(s(t)=G) = p_{BG} m(t) + p_{GB} (1-m(t))$$  \hspace{1cm} (4)

Thus, given a known starting state $x(0)$, the probability that the line is an error state at time step $t$ can be obtained by the recursive application of (4) $t$ times. In this paper, we assume henceforth that the state of the model at $t = 0$ is encoded as $m(0) = 1$, i.e. $p(s(0)=B) = 1$. As such, the worst-case for the model is that it is in the ‘Bad’ state with probability 1 at $t = 0$, such that it is aligned with the SAS of the tasks/messages to be scheduled by the system. This ensures that the worst-case arrival pattern of tasks occurs at the same time as the probabilities for error arrivals are maximized. Since the actual model can only ever be in one state or the other in a particular time step - and the probabilities of being in either ‘Bad’ or ‘Good’ at this step sum to one – the two states are exhaustive and mutually exclusive. As such the probability that an error will arrive at any step $i$ can be written:

$$p(i) = \lambda_B p(s(i)=B) + \lambda_G p(s(i)=G) = \lambda_B m(i) + \lambda_G (1-m(i))$$  \hspace{1cm} (5)

The Markov model may be used to modulate the probability of success for each of the $t$ Bernoulli variables as described in the previous Section, and thus the error model can be considered a ‘Markov-Modulated Poisson Binomial’ (MMPB) process. However if recursion on the Markov model is employed to directly obtain these probabilities, the complexity would be $O(t)$; this again is unsuitable for our purposes for the reasons described in the previous Section. In the following Section, we will show how this problem may be overcome such that tight bounds for $\eta(t, B)$ may be obtained in constant time per evaluation, for arbitrary $t$.

IV. AN EFFICIENT BOUND ON THE NUMBER OF ERRORS

A. Preliminary Lemma

Prior to presenting the main result of the paper, we will introduce a ‘large deviation’ probability inequality that will be required for the proof of our main Theorem.

Lemma 1: Given a random variable $k$ that is the sum of $n$ (possibly non-identical) Bernoulli variables $x(i), 0 \leq i < n$, with probability of success $p(x(i)=1) = p(i)$, expected value $E[k] = p(0) + p(1) + \ldots + p(n-1)$ and standard deviation $\sigma = \sqrt{\text{Var}[k]}$, then given some real $q \geq 0$ an upper bound on the residual probability of the upper tail of the distribution is:

$$P(k > E[k] + q\sigma) < \exp\left(-\frac{q^2}{2q^2 + \frac{1}{3\sigma^2}}\right)$$  \hspace{1cm} (6)

Proof: Bennett (1962) [21].

The expression above is an enhancement of an inequality originally attributable to S.N. Bernstein in the 1920s, the latter of which has appeared in several forms in the literature;

---

1 In many cases the assumption of mutual exclusion between error arrivals due to being in a ‘Bad’ or ‘Good’ state will not hold in practice if the ‘Good’ state represents a background residual error arrival probability, as the error sources will exhibit statistical independence. In this case, a trivial adjustment of $\lambda_B$ to $\lambda_B' = \lambda_B + \lambda_G - (\lambda_B \lambda_G)$ can be performed to reflect this situation.
it was subsequently derived as a special case of more general bounds presented by Chernoff and Hoeffding (e.g. [21][22][23]). The inequality originally presented by Bennett is slightly more general than the version presented above (it allows for non-Bernoulli variables and also assumes zero mean), however the adaption to expression (6) is trivial and we will henceforth refer to the inequality (6) as the Bennett / Bernstein bound. Several inequalities with similar form to that of (6) can be found in literature; we selected the expression above as it may be solved algebraically for \( q \) (whereas many variants of the classical Chernoff bound cannot), yet it performed almost as well as the Chernoff bound in an experimental comparison for this particular application. We will use the inequality stated above in the following way; given a desired failure rate \( \lambda_f \) (the compliment of the confidence probability \( R \)), this can be substituted into the l.h.s. of (6) and the relationship assumed to be an equality. Assuming that \( E[k] \) and \( Var[k] \) are known, the resulting equation may be solved for the unknown \( q \) thus placing a bound from above upon the \( R^k \) quantile. Now, we can state the main result of the current paper.

B. Statement and Proof of Main Result

Theorem 1: Given a Markov model as described in Section 2.3 described by the parameters \( \{p_{bb} \in [0, 1], p_{gb} \in (0, 1], \lambda_b \in [0, 1], \lambda_g \in [0, 1] \} \) and a confidence probability \( R \in [0.5, 1) \). Defining the following four quantities: \( C(R) = -\ln((1-R)) \), \( \alpha = (p_{bb} - p_{gb}) \), \( \beta = (\lambda_b - \lambda_g) \), \( m_i = p_{gb} / (1 - \alpha) \) and assuming the system is in the burst state with probability ‘1’ at \( t = 0 \), we have that for any \( t \geq 0 \):

\[
\eta(t, R) \leq \eta_s(t, R) = \left( e(t) + \sqrt{2C(R) \sigma(t) + \frac{C(R)}{3}} \right) \tag{7a}
\]

with the expected value \( e(t) \) at time \( t \) calculated from:

\[
e(t) = (\lambda_g + \beta m_t) t + \beta(1 - m_t) \left( \frac{1 - \alpha'}{1 - \alpha} \right) \tag{7b}
\]

and an upper bound \( \sigma(t) \) on the standard deviation at time \( t \) calculated using:

\[
\sigma(t) \leq \sqrt{\frac{e(t) - e(t)}{t}} \tag{7c}
\]

Proof: As discussed above, at time \( t \) we have \( t \) non-identical Bernoulli variables \( x(i) \), indexed from \( i = 0 \) through \( t-1 \), with an associated probability of error \( p(i) \) obtained via modulation from the Markov model. Denote the random variable that is the sum of these \( t \) Bernoulli variables as \( k \), with an expected value at time \( t \) given by \( e(t) = p(0) + p(1) + \ldots + p(t-1) \) and variance \( \text{Var}(t) = \text{Var}(x(0)) + \text{Var}(x(1)) + \ldots + \text{Var}(x(t-1)) \). Firstly, we will establish that a bound on the maximum value that \( k \) will obtain for a confidence probability \( R \) can be obtained through equation (7a). Since \( R \geq 0.5 \) we are in the upper tail of the distribution, hence the Bennett / Bernstein bound of expression (6) can be applied. Substituting \( (1-R) \) into the l.h.s. of the bound, setting as an equality and taking natural logs:

\[
\ln(1 - R) = -\frac{q^2}{1 + \frac{q}{3\sigma(t)} + \sqrt{1 + \frac{2q}{3\sigma(t)}}} \tag{8}
\]

substituting in the definition \( C(R) = -\ln(1-R) \) into the expression and re-arranging:

\[
C(R) = \frac{q^2}{1 + \frac{q}{3\sigma(t)} + \sqrt{1 + \frac{2q}{3\sigma(t)}}} \tag{9}
\]

which after some further manipulation and simplification can be put into a standard quadratic form:

\[
q^2 - \frac{2C(R)}{3\sigma(t)} q + \frac{C(R)^2}{3\sigma(t)} = 0 \tag{10}
\]

and taking the principal root of the quadratic (to ensure \( q \geq 0 \)) then simplifying the result gives:

\[
q = \sqrt{2C(R)} + \frac{C(R)}{3\sigma(t)} \tag{11}
\]

Noting that the CDF of the distribution is discrete, its value can only change at integer values of \( q \sigma(t) \); thus to ensure that the Bennett / Bernstein bound (6) is respected for integer \( q \sigma(t) \), the ceiling operation is appropriate and hence:

\[
k \leq \left[ e(t) + q \sigma(t) \right] = \left[ e(t) + \sqrt{2C(R) + \frac{C(R)^2}{3\sigma(t)}} \right] \tag{12}
\]

which is as stated in (7a). Next, we will show that the expectation \( e(t) = p(0) + p(1) + \ldots + p(t-1) \) can be computed exactly for the Markov model at time \( t \) with initial state \( m(0) = 1 \) using the expression given in (7b). Firstly, expanding the r.h.s of (4):

\[
m(i + 1) = p_{bb} m(i) + p_{gb} (1 - m(i)) = p_{bb} m(i) + p_{gb} m(i) = (p_{bb} - p_{gb}) m(i) + p_{gb} = \alpha m(i) + p_{gb} \tag{13}
\]
Which is a non-homogeneous recurrence having \( \alpha = (p_{GB} - p_{CB}) \) as the co-efficient of recurrence. Since \( p_{GB} > 0 \), we have that \(-1 \leq \alpha < 1\) and the recurrence is stable in the sense that \( |m(i)| \leq |m(0)| \) for all \( i \geq 0 \). Consider first the case in which \( |\alpha| < 1 \), such that each successive value of \( m(i) \) exponentially approaches the fixed-point of the recurrence (4), with alternating signs for negative \( \alpha \). Let us denote the fixed-point of the recurrence (the 'steady-state' of the Markov model) as the value \( m_s \). The value of \( m_s \) is easily obtained by substituting \( m(i+1) = m(i) = m_s \) into (13) and solving:

\[
m_s = \alpha m_s + p_{GB} \rightarrow m_s = \frac{p_{GB}}{1 - \alpha}
\]

and given the assumption that the initial state \( m(0) = 1 \) and using (13) and (14), a simple closed form to compute \( m(i) \) for all \( i \geq 0 \) can be written:

\[
m(i) = m_s + (m(0) - m_s) \alpha^i = m_s + (1 - m_s) \alpha^i
\]

Considering now the limit case \( \alpha = -1 \) (which can only occur when \( p_{GB} = 0 \) and \( p_{CB} = 1 \)), we can see from (11) that each successive value of \( m(i) \) oscillates between 0 and 1, giving a sequence starting at \( m(0) = 1 \) and successively taking on the values 0, 1, 0, 1, ... for increasing \( i \). In this case, (14) gives the steady-state \( m_s = 0.5 \), which is the center of the oscillation; it is easy to verify that successive values in the sequence are still given by (15). Hence (14) and (15) remain valid over the specified range of model transition probabilities. Now observing that since \( e(t) = p(0) + p(1) + ... + p(t-1) \), substituting into this the expression for \( p(i) \) given by equation (5):

\[
e(t) = \sum_{i=0}^{t-1} p(i) = \sum_{i=0}^{t-1} (\lambda_G m(i) + \lambda_G (1-m(i)))
\]

simplifying and substituting in the definition of \( \beta \) in the statement of the Theorem:

\[
\sum_{i=0}^{t-1} (\lambda_G m(i) + \lambda_G (1-m(i))) = \sum_{i=0}^{t-1} (\lambda_G m(i) + \lambda_G - \lambda_G m(i))
\]

\[
= \sum_{i=0}^{t-1} (\lambda_G + \beta m(i))
\]

\[
t \lambda_G + \sum_{i=0}^{t-1} \beta m(i)
\]

substituting the expression for \( m(i) \) given by (15) in to (17) and simplifying further:

\[
t \lambda_G + \sum_{i=0}^{t-1} \beta m(i) = t \lambda_G + \sum_{i=0}^{t-1} \beta (m_s + (1 - m_s) \alpha^i)
\]

\[
= t \lambda_G + \beta \sum_{i=0}^{t-1} m_s + (1 - m_s) \sum_{i=0}^{t-1} \alpha^i
\]

\[
= t \lambda_G + \beta m_s + (1 - m_s) \sum_{i=0}^{t-1} \alpha^i
\]

Next, to solve the rightmost summation term of (19) we apply the well-known identity for the sum of the first \( n \) terms of a geometric series:

\[
\sum_{i=0}^{n-1} \alpha^i = \left(1 - \alpha^n \right) / (1 - \alpha) \quad -1 \leq \alpha < 1
\]

and substituting (20) into the last line of (19) gives:

\[
e(t) = \left(\lambda_G + \beta m_s + \beta (1 - m_s) \left(1 - \alpha^t \right) \right) / (1 - \alpha)
\]

which is the expression stated in (7b). In order to complete the proof, it is needed to show that the true variance of the variable \( k \) at time \( t \) (denoted by \( v(t) = Var(x(t)) \) + \( ... + Var(x(t-1)) \)) is upper bounded by the square of the r.h.s. in expression (7c). The variance of a Bernoulli variable is directly related to its probability of occurrence according to \( Var(x(t)) = p(t)(1-p(t)) \). Thus the total variance in \( k \) at time \( t \) can be written as:

\[
v(t) = \sum_{i=0}^{t-1} p(i) = \sum_{i=0}^{t-1} (p(i)(1-p(i)))
\]

\[
= \sum_{i=0}^{t-1} (p(i) - p(i)^2) = \sum_{i=0}^{t-1} p(i) - \sum_{i=0}^{t-1} p(i)^2
\]

\[
= e(t) - \sum_{i=0}^{t-1} p(i)^2
\]

and defining the mean expectation as \( \mu = e(t)/t \), the sum of squared probabilities in (22) can be re-written as:

\[
\sum_{i=0}^{t-1} p(i)^2 = \sum_{i=0}^{t-1} \mu^2 + \sum_{i=0}^{t-1} (\mu - p(i))^2 = t \mu^2 + \sum_{i=0}^{t-1} (\mu - p(i))^2
\]

Since the rightmost term in (23) is always \( \geq 0 \) due to the squaring of the deviation from the mean it can be written:

\[
v(t) = e(t) - t \mu^2 - \sum_{i=0}^{t-1} (\mu - p(i))^2 \leq e(t) - t \mu^2 = e(t) - \frac{e(t)^2}{t}
\]

and since \( \sigma(t) = \sqrt{v(t)} \) the final part of the proof is finished.
C. Remarks

We think Theorem 1 warrants some further comments; firstly, note that the expressions of (7) may be evaluated in constant time and with time and space complexity $O(1)$. For a given set of MMPB model parameters and desired probability $R$, many of the coefficients for (7a), (7b) and (7c) need only be computed once, and this may even be done offline in the case of admission controls. In this situation an evaluation of the bound for any single $t$ may then be carried out with $\approx 10$ FLOPS, plus those needed for one exponentiation and one square root. This makes them suitable even for on-line use in fault-tolerant task admission control. The price that is paid for this efficiency is a slight over-estimation of the number of errors; in the next Section, we show that the bound is tight, and that the relative error becomes vanishingly small for large $t$. Additionally, we avoid the need to directly calculate the variance; we observe that the exact computation of $\nu(t)$ is also possible in constant time, however we have found this does not lead to any significant improvement. This is principally due to the fact that the overestimation in (7c) is generally small (and is also square-rooted); in addition, as $r$ becomes large each of the probabilities converges upon the steady-state $m_r$, and the distribution becomes asymptotically Binomial. In this condition the sum of squared deviations from the mean in (23) becomes vanishingly small with respect to the quantity $e(t) - tp$.

The specified range of state transition probabilities allowed in the MMPB model ($p_{BB} \in [0,1]$ and $p_{GB} \in (0,1]$) corresponds to the logical case where $\mu_g \geq 1$ and $\mu_b \geq 1$. For situations in which the system does not experience burst errors but has a single static probability of error $p$ at each time step, then setting $p_{BB} = 1$ and $\lambda_g = p$ will achieve the desired effect. However, the computation of the expectation $e(t)$ can now be simplified considerably, which we state without proof in the following trivial Corollary.

Corollary 1: Given a random variable $k$ that is the sum of $t$ identical Bernoulli variables with probability of success $p$, then the expressions (7a) and (7c) hold with the expected value of $k$ given by $e(t) = tp$.

Note also that the calculation of $\nu(t)$ with (7c) now also becomes exact as the sum of squared deviations from the mean in (23) exactly equals zero. One further observation that can be made is related to the actual arrival times of the errors. The model does not make any assumptions related to the (temporal) distribution of the errors in the interval $[0, t)$, except that only one error can arrive at any individual time step. As such there is no concept of a minimal inter-arrival time between any two successive errors; they may all be clustered close to 0 (or alternatively clustered close to $t$), or evenly distributed across the entire interval. This is an appropriate generalization as it leaves open many possibilities for tailoring to the specifics of the higher-level system and scheduling algorithm details, and any worst-case assumptions about error arrival patterns that may be needed for the higher-level error propagation models. Perhaps one drawback of this generalization is that care must be taken regarding the interpretation of the results of a schedulability analysis using the equations. If, following the SAS, a set of real-time tasks $\Gamma$ scheduled with scheduling algorithm $S$ has enough slack to cope with $\eta(t, R)$ errors for all $t \geq 0$, then the system will function without overload and hence without any deadline misses with probability $R$.

The use of the term without overload should be noted here; the complement of this statement is with overload, and should an overload occur (which may happen with probability $1-R$), then some deadline misses may occur depending on the choice of $S$. This is due to the well-documented domino-effect of failures in overload conditions for certain kinds of scheduling algorithms (e.g. EDF – see [4], Ch. 5). This point may be especially important when quantifying the dangerous failure rate of a system; it is not necessarily $(1-R)$ if a single deadline miss is classified as a dangerous failure. In such situations some form of run-time overload management scheme will also be required to prevent starvation of critical tasks and enforce a fair scheduling policy. Save from directing the interested reader to some general texts [3][4][15], we do not consider such schemes further in this paper. In Section VI we do, however, provide a basic illustrative example of the use of the error model in fault-tolerant schedulability analysis.

V. PROPERTIES OF THE BOUND

In this Section we first present some empirical results to illustrate the tightness of the bound on $\eta(t, R)$ obtained with equations (7). In the first example, we consider a system experiencing severe bursts of errors with mean inter-arrival 1000 and length 100. The probability of error occurrence during a burst is given by 0.3 (i.e. $\lambda_b = 0.3$), and the background probability of error is 0.001 (i.e. $\lambda_g = 0.001$). Figures 2 and 3 illustrate the exact number of errors the system should be designed to tolerate between $t = 0$ and $t = 2000$, and also the bound on the errors obtained from (7) for confidence probabilities of $R = 0.99999$ and $R = 0.9999999999$. In order to obtain the exact quantile (computed via. equation (2)), we employed a modified version of the recursive algorithm of Barlow & Heidtmann [24] for computing the CDF (the DFT-based algorithm of Fernandez & Williams [18] was found to be numerically unstable for $t > 300$).

As can be seen from these two figures, the higher the required probability of operation without overload, the more errors the system needs to be capable of tolerating in any time interval starting from the critical instant. It is also interesting to observe that, as may intuitively be expected, the error density is maximized at low $t$ but steadily reduces into a line of (almost) constant slope as the Markov probabilities converge upon the steady-state. Error arrival density is a nonlinear function of time, and has elements of both a static $'k'$-burst type of error model and also a pseudo-periodic error model.
From the figures, the level of overestimation in the number of errors seems acceptable in both cases. We will now strengthen this observation by showing that the bounds are asymptotically tight in the sense that the relative error vanishes as \( \sigma(t) \to \infty \). Let us first define the relative error in the bound at time \( t \) as the positive real variable \( \epsilon(t) \):

\[
\epsilon(t) = \frac{\eta(t) \tau(t) - \eta(t) \xi(t)}{\eta(t) \xi(t)}
\]  

(25)

Theorem 2: In the limit as \( \sigma(t) \to \infty \), \( \epsilon(t) \to 0 \) and the convergence is \( O(\sigma^{-1}) \).

Proof: As a direct consequence of the Central Limit Theorem, as \( \sigma \to \infty \) the distribution of a Poisson-Binomial random variable becomes asymptotically normal \([17][22]\), and hence its quantile satisfies the relationship:

\[
\lim_{\sigma(t) \to \infty} \eta_{\tau}(t) \tau(t) = \epsilon(t) + \zeta(R) \sigma(t)
\]  

(26)

where \( \zeta(R) \) is the inverse of the standard normal CDF \( \Phi(\cdot) \), i.e. \( \Phi(\zeta(R)) = R \). From \([22]\) we have that \( 0 \leq \zeta(R) < \sqrt{2\text{C}(R)} \) when \( R \geq 0.5 \), and using (25) as a lower bound on \( \eta(t, R) \) the relative error may be bounded as follows:

\[
\lim_{\sigma(t) \to \infty} \epsilon(t) \leq \frac{\eta_{\tau}(t, R) - \eta(t, R)}{\eta(t, R)}
\]

\[
= \frac{(\epsilon(t) + \sqrt{2\text{C}(R)} \sigma(t) + \text{C}(R)/\text{3})}{\epsilon(t) + \zeta(R) \sigma(t)} - 1
\]

\[
\leq \frac{\sqrt{2\text{C}(R) - \zeta(R)} + \text{C}(R)/\text{3}}{\epsilon(t) + \zeta(R) \sigma(t)}
\]

(27)

where the final inequality above follows from the trivial lower bound \( \epsilon(t) \geq \sigma(t)^2 \). For any fixed \( R \), the quantities \( \zeta(R) \) and \( \sqrt{2\text{C}(R)} \) are both finite and independent of \( \sigma(t) \), and hence the error \( \epsilon(t) \to 0 \) as \( \sigma(t) \to \infty \). Thus \( \epsilon(t) \) can be asymptotically bounded by some constant \( K \) (depending upon \( R \)) such that \( \epsilon(t) \leq K \sigma(t)^3 \). Finally, observe that as \( t \) becomes large the overestimation error in the variance (7c) becomes vanishingly small with respect to the quantity \( \tau(t) - \tau_+^{\text{t}} \), and the stated convergence to zero in the limit holds.

Before proceeding to discuss how the error model may be incorporated into a schedulability analysis, there is one more result that needs to be established. Given that the number of errors in a given time interval is a non-linear function of the length of the interval itself and is also non-decreasing, from a real-time perspective it needs to be established that the error density (i.e. the rate of increase in the number of errors relative to the rate of increase in the length of a considered interval) is indeed finite and always decreasing.

Theorem 3: Assume that \( \lambda_{r} \geq \lambda_{G} > 0 \) and \( \mu_{EG} \geq \mu_{EB} > 0 \). Then given any two intervals \([0, t) \) and \([0, 2t), t > 0 \), we have that \( \eta(2t, R) \leq 2 \eta(t, R) \).

Proof: The expectation \( \epsilon(t) \) is formed from the sum of \( t \) non-negative probabilities which are in a geometric progression to the steady-state \( m_{s} \), and hence \( \epsilon(2t) > \epsilon(t) \). As we have that \( \lambda_{s} \geq \lambda_{G} > 0 \) and \( \mu_{EG} \geq \mu_{EB} > 0 \), we have that \( p(i) \geq p(i+1) \geq m_{s} \) for any \( i \geq 0 \) since \( \alpha = (p_{AB} - p_{EB}) > 0 \). Hence \( \epsilon(t) < \epsilon(2t) \leq 2\epsilon(t) \) and also \( \epsilon(t) < \epsilon(2t) \leq 2\epsilon(t) \), and the result follows from the definition of \( \eta \) given by (7a) since \( C(R) > 0 \).

The result established by Theorem 3 is important in the following sense. Suppose, following the SAS, that a set of real-time tasks \( \Gamma \) scheduled with scheduling algorithm \( S \) has enough slack to tolerate every error given by \( \eta(t, R) \) and still meet every deadline, up to some specific time \( t' \). Then, if the worst-case manifestation pattern of the tasks in \( \Gamma \) is also contained within the interval \([0, t') \), then the probability that the system will function without overload in any interval \([0, 2t'), [0, 3t'), [0, 4t') \)… and so on will be at least \( R \). From this, it is evident that relatively simple modifications to existing
schedulability analysis techniques may be applied with (7) to
give a one-shot schedulability analysis capable of verifying
that a system can operate with reliability $R$ without deadline
failure. In the next section, we give an example of its
application to dynamic priority preemptive scheduling, by
deriving sufficient schedulability conditions for a task set
scheduled by a simple fault-tolerant EDF algorithm.

VI. APPLICATION TO FAULT-TOLERANT EDF
SCHEDULING

As mentioned above, although only relatively simple
modifications to existing schedulability analysis techniques
will normally be required, the required modifications are
heavily influenced by the higher-level assumptions that are
made regarding the error detection and fault-tolerance
capabilities of the computing/communication system and
scheduling algorithm. These assumptions and behaviors must
be captured when considering the worst-case manifestation
patterns of errors with respect to CPU demands and
overloads in a given interval of time.

To keep the discussions relatively simple, assume that a
standard task set $\Gamma$ is to be CPU scheduled by a Fault
Tolerant Earliest Deadline First (FT-EDF) scheduling
algorithm, in which the modification is simply to re-insert a
task that fails due to an error into the run queue with its
original deadline. Assume also that (i) all errors will manifest
as task failures, (ii) all task failures are detected and (iii) if $k$
task failures occur in the time interval $[0, t]$, then each of
these $k$ failures will manifest itself in a worst-case pattern
such that the demand for CPU time in the interval $[0, t]$ is
maximized.

Following the worst-case arrival pattern of tasks under
EDF, when considering some job with a deadline at $t = d$,
then the worst-case behavior as per point (iii) above is
induced when one of the mandatory jobs – the one with the
largest execution - time fails repeatedly due to error. Under
the EDF scheduling policy, the mandatory jobs out of those
eligible for execution in the interval $[0, d]$ following the SAS
are those which are generated by tasks satisfying the
relationship $D_i \leq d$. As such the worst-case CPU demand
due to failed jobs in the interval $[0, t]$, denoted by the function
$f(t)$, can be bounded as follows:

$$f(t) \leq \max_{D_i \leq t} \{C_i \} \cdot \eta_i(t, R) \quad (28)$$

which leads to the following sufficient condition to verify the
$R$-schedulability of the tasks in $\Gamma$:

Theorem 4: Given a set $\Gamma$ of tasks defined as per (1), and
an MMPB error model defined as in Section 3 satisfying the
conditions of Theorem 3, if the tasks are sorted in order of
non-decreasing relative deadline the following conditions are
sufficient to verify that all deadlines will be met with
probability $R$ for FT-EDF scheduling a set of tasks or
messages:

$$\forall j = 1, \ldots, n :$$

$$\sum_{i=1} u_i + \sum_{j=1}^{k} \frac{T_j - \min\{T_i, D_j\}}{T_i D_j} C_i + \frac{f(D_j)}{D_j} \leq 1.0 \quad (29)$$

Proof: To achieve a reliability of operation $R$, we have from
Theorem 4 that not more than $\eta_D(D)$ errors need to be
considered when examining any interval of length $D$, and thus
from (28) a bound on the worst-case utilization penalty
due to the re-execution of failed jobs in the interval $D$ is
given by the quantity $f(D)/D$. The Theorem then follows
directly from Theorem 5 in Devi [25].

The original EDF schedulability test proposed by Devi
(from which (29) was derived) has a linear complexity if the
tasks are ordered by non-decreasing relative deadline, thus
the inclusion of the initial sort leads to an overall time and
complexity $O(n \log n)$. Since $\eta_D(t, R)$ may be calculated in
constant time for any $t$, the overall complexity of evaluating
(29) is also $O(n \log n)$ time, with the only a small measurable
overhead increase as discussed in Section VI. We note that
the various extensions proposed by Devi in [25] (e.g. to
model effects of limited/non-preemption) may also be
incorporated into (29) with relative ease.

VII. ILLUSTRATIVE EXAMPLE

In this Section we discuss an example to illustrate the
main concepts proposed in the paper. Consider a system
experiencing bursts of errors with mean inter-arrival 100,000
and mean length 1000, with a mean error inter-arrival during
a burst of 100 and a mean background inter-arrival of
1,000,000. It is required to schedule a set of $n = 4$ real-time
tasks $\Gamma = \{\tau_1, \tau_2, \tau_3, \tau_4\}$ with the following parameters:

<table>
<thead>
<tr>
<th>Task</th>
<th>T</th>
<th>D</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>100</td>
<td>100</td>
<td>6</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>1000</td>
<td>1000</td>
<td>35</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>2000</td>
<td>2000</td>
<td>45</td>
</tr>
<tr>
<td>$\tau_4$</td>
<td>5000</td>
<td>5000</td>
<td>150</td>
</tr>
</tbody>
</table>

For this task set the total (non-faulted) CPU utilization is
14.75%, and the tasks are trivially schedulable in error-free
conditions. The question we wish to address here is: are the
tasks schedulable in an application requiring a reliability $R =
0.999999$ when scheduled using the FT-EDF algorithm
described in Section VI? Firstly, we form the constant
$C(R) = -\ln(0.00001) = 11.513$, thus giving the expression for $\eta_D(t, R)$ as:

$$\eta_D(t, R) = \left[ e(t) + 4.7985 \sigma(t) + 3.8376 \right] \quad (30)$$

Next, we can apply (29) to determine schedulability;
firstly we compute, for each $D_n$, the fault load $f(t)$. For $t =
100$, from (7b) and (7c) we have that $e(t) = 0.952099$ and
$\sigma(t) \leq 0.971099$ (the true SD is equal to 0.971095), which
gives $\eta_D(100, R) = 8$ and hence $f(100) = 48$. Repeating for
each \( D \) we obtain \( \eta(1,000, R) = 20 \) and \( f(1,000) = 700, \eta(2,000, R) = 25 \) and \( f(2,000) = 1125, \) and finally \( \eta(5,000, R) = 28 \) with \( f(5,000) = 4,200. \) Next, we can evaluate the schedulability of the tasks by forming the load \( L(t) \) (i.h.s. of (29)) at each deadline: this gives \( L(100) = 0.54, L(1,000) = 0.80, L(2,000) = 0.68 \) and \( L(5,000) = 0.99, \) each of which is \( \leq 1 \) thus verifying \( R \)-schedulability. It is interesting to observe that although the CPU utilization would be low in an error-free environment, a large amount of the spare CPU capacity is required to tolerate the worst case conditions for this simple FT-EDF scheduler (admittedly the environmental conditions are somewhat harsh in this example). However, it does also illustrate that naïve FT mechanisms such as that used in this EDF scheduler (unbounded job re-execution or message re-transmission in the case of error-induced job failure) are not overly effective in real-time settings, as has been previously observed (e.g. in [4]-[26]).

VIII. CONCLUSION

In this paper we have considered a quantile-based approach to probabilistic schedulability analysis in a bid to reduce complexity whilst still retaining a rich stochastic error model. Our principal contribution has been the derivation of a simple closed-form expression that tightly bounds the error model into a simple FT-EDF schedulability analysis. Extensions of the error model to cases where more complex failure conditions are somewhat harsh in this example). However, it does also illustrate that naïve FT mechanisms such as that used in this EDF scheduler (unbounded job re-execution or message re-transmission in the case of error-induced job failure) are not overly effective in real-time settings, as has been previously observed (e.g. in [4]-[26]).

ACKNOWLEDGMENTS

Part of the work presented in this paper was carried out whilst Julían Proenza was visiting the Electronics & Control Group, Teesside University in June/July 2012. Financial support for this visit from Teesside University is gratefully acknowledged. This work was also partially supported by the Spanish Economy and Competitiveness Ministry with grant DPI2011-22992 and by FEDER funding.

REFERENCES