How Synchronisation Strategy Approximation in PEPA Implementations affects Passage Time Performance Results

Jeremy T. Bradley[1]   Stephen T. Gilmore[2]   Nigel Thomas[3]

[1] Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2BZ, United Kingdom.
jb@doc.ic.ac.uk
[2] Laboratory for Foundations of Computer Science
The University of Edinburgh,
Edinburgh EH9 3JZ, United Kingdom.
Stephen.Gilmore@ed.ac.uk
[3] School of Computing Science,
University of Newcastle-upon-Tyne,
Newcastle-upon-Tyne NE1 7RU, United Kingdom.
Nigel.Thomas@ncl.ac.uk

Abstract. Passage time densities are useful performance measurements in stochastic systems. With them the modeller can extract probabilistic quality-of-service guarantees such as: the probability that the time taken for a network header packet to travel across a heterogeneous network is less than 10ms must be at least 0.95. In this paper, we show how new tools can extract passage time densities and distributions from stochastic models defined in PEPA, a stochastic process algebra. In stochastic process algebras, the synchronisation policy is important for defining how different system components interact. We also show how these passage time results can vary according to which synchronisation strategy is used. We compare results from two popular strategies.

1 Introduction

Probabilistic quality-of-service guarantees underpin most commercial SLAs (service level agreements): e.g. the probability that a 10-node cluster should be able to process 3000 database transactions in less than 6 seconds should be greater than 0.915; or a train service should not run more than 10 minutes late more than 20% of the time. Whether these commercial guarantees are met or broken depends on the aggregate time behaviour across a whole system of complex interactions.

It is frequently useful to model these systems with a process model and still further convenient to let individual process actions have random delay: this random delay might represent either incomplete or uncertain knowledge on the part of the modeller, or a good approximation to underlying aggregate complex but deterministic dynamics or genuine random behaviour.

All these factors combined point to the conclusion that a stochastic process algebra (SPA) such as PEPA [1], EMPA [2] or IMC [3] is an appropriate modelling tool for many commercial and industrial problems. In this paper we use Hillston's Performance Evaluation Process Algebra (PEPA).

Traditionally, SPAs like PEPA have been analysed for mean statistics or steady-state values [4–8] with later extensions to transient (time-varying) measures with techniques such as uniformisation [9]. Here we not only show how complete passage time densities can be extracted but also how such passage times are affected by the choice of synchronisation strategy.

A synchronisation strategy defines how the components of a process algebra model interact. In SPAs which just employ Markovian transitions it is a distinguishing feature of the different SPAs [10]. Here we compare two popular strategies in PEPA:

minimum rate strategy from [4], which is easy to implement and understand and is used in most tool implementations. It can occasionally distort global rates of enabled actions. Also implemented in Möbius [11] and PRISM [12].

apparent rate strategy from [1], which has the benefit of making certain equivalences congruences and precisely represents the minimum rate at the global state space level. It is implemented in ipc [13].

Given that these strategies can have an effect on the end performance statistics of a model, it is important to quantify this effect and understand when differences may occur.

The paper is organised as follows: in Section 2 we discuss the background to the choice of synchronisation strategies; in Section 3, we describe the PEPA stochastic process algebra; in Section 4, we demonstrate passage time extraction from a PEPA model and in Section 5 we present a taxonomy of instances of when the synchronisation models differ in behaviour and results.

2 Background

In this section, we discuss the idea of a synchronisation strategy for stochastic process algebras. As described by Milner [14], sequential or serial modelling formalisms contrast with concurrent ones such as process algebras because the former use the *operator/operand* paradigm and the latter use the *cooperator/cooperand* paradigm. That is, concurrently active components are peers who cooperate and share work such that each participates actively in the shared activity. This naturally gives rise to questions such as "when P can perform α at rate r_1 and Q can perform α at rate r_2; and what is the rate at which α occurs when they cooperate to perform it?" The answer to this question is given by the synchronisation strategy of the process algebra.

There are many plausible definitions for a synchronisation strategy for a stochastic process algebra (see [15, 16] for a fuller discussion of this issue) but for a Markovian process algebra (MPA)—restricted to using only exponential distributions for rates—some possibilities are ruled out for the technical reason that an arbitrary combination of two exponential distributions is not necessarily an exponential distribution. For this reason some MPAs (such as TIPP [17]) chose functions for synchronisation strategies primarily because they satisfied the algebraic requirement that the chosen function produces an exponential distribution when applied to two exponentials. One such function is rate product, which was the choice of the designers of the TIPP language.

However, to use rate product as the synchronisation function does not accord well with our intuitions about the physical world and in fact essentially the only reason to choose rate product is because of its algebraic properties. To explain with an example, if we consider a print spooler which can spool PostScript files at a rate of a hundred pages per minute and a printer which can render them at a rate of ten pages per minute then the TIPP prediction is that when these components synchronise then the resulting assembly will be able to print a thousand pages per minute! The PEPA process algebra would instead say that the combination would print ten pages per minute (because the faster component is hindered by the slower one). This is in accord with our expectations and our experience of how the bottleneck device in the system limits the throughput of the whole.

Further, our spooler and printer illustration is not an isolated pathological example. An earlier study [16] showed that the performance results computed from the use of rate product as a synchronisation strategy could be arbitrarily wrong. The same paper went on to show that the strategy used by PEPA produced results which were in good agreement with a range of other reasonable synchronisation disciplines such as first-to-finish and last-to-finish.

The technical definition which provides PEPA with this intuitive behaviour when components synchronise is that of the *apparent rate* defined by Hillston [1] and adopted by other process calculi such as the Stochastic π-calculus [18]. Hillston's apparent rate calculation is perhaps the simplest function which simultaneously satisfies the two key requirements of:

1. building an exponential distribution when applied to two exponentials; and
2. computing numerical results which accord with our intuitions of synchronising timed activities.

However, the computation of the apparent rates of the transitions of a PEPA model must be performed efficiently if the (perhaps large) state-space of the model is to be derived effectively. In order to avoid the cost of this calculation it is tempting to approximate the apparent rate with the minimum rate. The novel contributions of this paper are the discussion of the cost of computing apparent rates and the comparison of the correct calculation of apparent rates with their approximation by minimum rates.

3 PEPA

PEPA [1] is a parsimonious stochastic process algebra that can describe compositional stochastic models. These models consist of components whose actions incorporate random exponential delays.

The syntax of a PEPA component, P, is represented by:

$$\mathbf{P} ::= (\mathbf{a}, \lambda).\mathbf{P} \;\Big|\; \mathbf{P} + \mathbf{P} \;\Big|\; \mathbf{P} \underset{S}{\bowtie} \mathbf{P} \;\Big|\; \mathbf{P}/\mathbf{L} \;\Big|\; \mathbf{A} \tag{1}$$

$(\mathbf{a}, \lambda).\mathbf{P}$ is a prefix operation. It represents a process which does an action, $\mathbf{a}$, and then becomes a new process, $\mathbf{P}$. The time taken to perform $\mathbf{a}$ is described by an exponentially distributed random variable with parameter λ. The rate parameter may also take the value $\top$, which makes the action passive in a cooperation (see below).

$\mathbf{P_1} + \mathbf{P_2}$ is a choice operation. A race is entered into between components $\mathbf{P_1}$ and $\mathbf{P_2}$. If $\mathbf{P_1}$ evolves first then any behaviour of $\mathbf{P_2}$ is discarded and vice-versa.

$\mathbf{P_1} \underset{S}{\bowtie} \mathbf{P_2}$ is the cooperation operator. $\mathbf{P_1}$ and $\mathbf{P_2}$ run in parallel and synchronise over the set of actions in the set S. If $\mathbf{P_1}$ is to evolve with an action $\mathbf{a} \in \mathbf{S}$, then it must first wait for $\mathbf{P_2}$ to reach a point where it is also capable of producing an $\mathbf{a}$-action, and vice-versa. In a cooperation, the two components then jointly produce an $\mathbf{a}$-action with a rate that reflects the slower of the two components (R in Figure 1).

$\mathbf{P}/\mathbf{L}$ is a hiding operator where actions in the set L that emanate from the component $\mathbf{P}$ are rewritten as silent τ-actions (with the same appropriate delays). The actions in L can no longer be used in cooperation with other components.

$\mathbf{A}$ is a constant label and allows, amongst other things, recursive definitions to be constructed.

The synchronisation strategy effects the cooperation operator $\mathbf{P_1} \underset{S}{\bowtie} \mathbf{P_2}$. The difference between the *apparent rate strategy* and *minimum rate strategy* occurs when either $\mathbf{P_1}$ or $\mathbf{P_2}$ enable multiple a-transitions, where $a \in S$. In [1], the apparent rate semantics are defined as in Figure 1, where:

$$r_a(P) = \sum_{P \xrightarrow{(a,\lambda_i)}} \lambda_i \tag{2}$$

where $\lambda_i \in \mathbb{R}^+ \cup \{n\top \mid n \in \mathbb{Q}, n > 0\}$, $n\top$ is shorthand for $n \times \top$ and $\top$ represents a passive action rate that will inherit the rate of the coaction from the cooperating component. $\top$ requires the following arithmetic rules:

$$\begin{aligned} m\top < n\top &: \text{for } m < n \text{ and } m, n \in \mathbb{Q} \\ r < n\top &: \text{for all } r \in \mathbb{R}, n \in \mathbb{Q} \\ m\top + n\top = (m+n)\top &: m, n \in \mathbb{Q} \\ \frac{m\top}{n\top} = \frac{m}{n} &: m, n \in \mathbb{Q} \end{aligned}$$

Cooperation (Non-synchronising)

$$\frac{P \xrightarrow{(a,\lambda)} P'}{P \underset{S}{\bowtie} Q \xrightarrow{(a,\lambda)} P' \underset{S}{\bowtie} Q} \; a \notin S$$

$$\frac{Q \xrightarrow{(b,\mu)} Q'}{P \underset{S}{\bowtie} Q \xrightarrow{(b,\mu)} P \underset{S}{\bowtie} Q'} \; b \notin S$$

Cooperation (Synchronising)

$$\frac{P \xrightarrow{(a,\lambda)} P' \quad Q \xrightarrow{(a,\mu)} Q'}{P \underset{S}{\bowtie} Q \xrightarrow{(a,R)} P' \underset{S}{\bowtie} Q'} \; a \in S$$

where $R = \frac{\lambda}{r_a(P)} \frac{\mu}{r_a(Q)} \min(r_a(P), r_a(Q))$

Fig. 1. An excerpt from the operational semantics for PEPA, showing only details of the cooperation operator

Note that $(r + n\top)$ is undefined for all $r \in \mathbb{R}$ in PEPA therefore disallowing components which enable both active and passive actions in the same action type at the same time, e.g. $(\mathbf{a}, \lambda).\mathbf{P} + (\mathbf{a}, \top).\mathbf{P}'$.

The minimum rate strategy is much simpler and is used in many tools [4, 11, 12] which implement PEPA. The semantics of the joint rate, R, in Figure 1 are rewritten as:

$$R = \min(\lambda, \mu) \tag{3}$$

for each instance of a cooperating action pair, where $\lambda, \mu \in \mathbb{R}^+ \cup \{\top\}$ and we only need to know that $r < \top$ for all $r \in \mathbb{R}^+$.

4 Extracting passage time distributions

In this section, we briefly describe how passage time results are extracted from SPA models, so that the reader may better understand the analysis process and the sort of quantitative results that are obtained.

Passage time densities in stochastic models are defined by their source and target states, i.e. the time taken to get from one set of states to another. Process algebras use transitions or actions as their central descriptive philosophy with states only implicitly occurring "between" actions, so we need a technique for moving between these two paradigms: i.e. extracting the source and target states for the passage in a way that can be easily related to the action-model of the process algebra.

It is convenient to define passage time densities in PEPA models by means of fragments of process algebra known as *stochastic probes* [19]. These probes specify the actions that should be seen in order to start and stop the passage time measurement and they can easily be interrogated to identify source and target actions for the passage. It is important that the probe does not affect the time-behaviour of the model it is measuring, so it only presents passive actions for the model to synchronise with and it does not block actions it is not interested in.

In fact, stochastic probes are SPA-independent and can be tailored to any SPA which supports multi-way synchronisation between processes (so that one can probe the key passage activities). PEPA suited our needs here as it has an uncomplicated syntax which lends itself to describing the underlying concepts behind stochastic probes. As we will see, the probe is expressed as a single PEPA component, so that it can then be combined with the model being queried.

$$
\begin{aligned}
\mathbf{A} &\stackrel{\mathit{def}}{=} (\mathbf{run}, \lambda_1).(\mathbf{stop}, \lambda_2).\mathbf{A} \\
\mathbf{B} &\stackrel{\mathit{def}}{=} (\mathbf{run}, \top).(\mathbf{pause}, \lambda_3).\mathbf{B} \\
\mathbf{Sys_0} &\stackrel{\mathit{def}}{=} \mathbf{A} \underset{\{run\}}{\bowtie} \mathbf{B}
\end{aligned}
$$

Fig. 2. PEPA description for model $\mathbf{Sys_0}$—a simple two component system cooperating over the $\mathbf{run}$ action

Figure 2 describes a very simple model with two components $\mathbf{A}$ and $\mathbf{B}$. $\mathbf{A}$ can perform a $\mathbf{run}$ then a $\mathbf{stop}$ before becoming $\mathbf{A}$ again; whereas $\mathbf{B}$ can perform a $\mathbf{run}$ then a $\mathbf{pause}$ before becoming $\mathbf{B}$ again. The two components synchronise over the $\mathbf{run}$ action, with the overall rate of the $\mathbf{run}$ action being dictated by λ_1 from the $\mathbf{A}$ component (since $\mathbf{B}$'s $\mathbf{run}$ action was passive, represented by the $\top$ symbol). We will briefly discuss the derivation of the passage time between successive $\mathbf{stop}$ actions in this model (a detailed discussion of passage times and stochastic probes can be found in [19, 20]).

Figure 3 shows the stochastic probe that is composed with the $\mathbf{Sys_0}$ model. This simple version of a probe just toggles state every time it observes a $\mathbf{stop}$ action. When we examine the PEPA model of Figure 2 with the Imperial PEPA Compiler (ipc) [13], it automatically generates the probe of Figure 3 and inserts the state-based logic to define the passage time between successive $\mathbf{stop}$ actions.

ipc's output is in a form readable by Dingle and Knottenbelt's HYDRA tool [21–23] which can then compute both the probability density function (PDF) and the cumulative distribution function (CDF) shown in Figures 4 and 5.

In this first example, there is only one possible $\mathbf{run}$ synchronisation (derived from exactly one active participant and exactly one passive participant) and the two synchroni-

$$\mathbf{Probe_{idle}} \stackrel{\mathit{def}}{=} (\mathbf{stop}, \top).\mathbf{Probe_{run}}$$
$$\mathbf{Probe_{run}} \stackrel{\mathit{def}}{=} (\mathbf{stop}, \top).\mathbf{Probe_{idle}}$$
$$\mathbf{Sys_1} \stackrel{\mathit{def}}{=} \mathbf{Probe} \underset{\{\mathtt{stop}\}}{\bowtie} \mathbf{Sys_0}$$

Fig. 3. PEPA version of the stochastic probe for model $\mathbf{Sys_0}$: toggles between started and stopped states according to whether it has observed a $\mathbf{stop}$ action or not

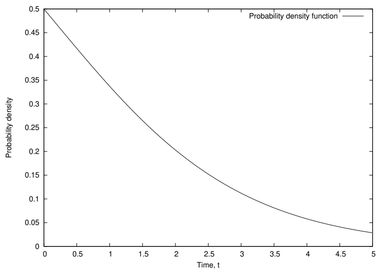

Fig. 4. Probability density function for time taken between successive $\mathbf{stop}$ actions in model $\mathbf{Sys_0}$

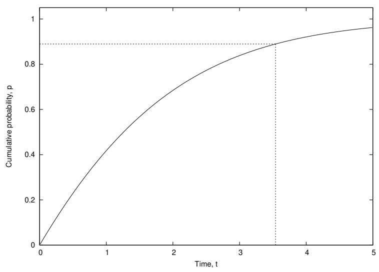

Fig. 5. Cumulative distribution function for time taken between successive $\mathbf{stop}$ actions in model $\mathbf{Sys_0}$. Shows that probability of successive stops occurring in less than 3.54 seconds is 0.8893

sation strategies considered in Section 3 (active and minimum) have identical behaviour in this situation.

In the next section, we will look at small variations of this model which add extra possibilities for synchronisation, in order to create a divergence in behaviour between the minimum rate and active rate strategies. It is this divergence we aim to study in the context of passage time analysis.

5 Results

For the analysis in this section we are going to need to compare passage times from different strategies. So given two random variables representing distinct passages, X_1 and X_2, we construct the quantity:

$$\psi(X_1, X_2, t) = \frac{F_{X_2}(t) - F_{X_1}(t)}{F_{X_1}(t)} \quad (4)$$

to compare their CDFs, $F_{X_1}(t)$ and $F_{X_2}(t)$, at different t-values. For each of the models being considered in this section we will look at both the individual PDFs as well as the

$\psi(\cdot)$ plot over the CDFs. This will give a good idea of the relative difference in CDF and thus passage time quantile result.

We consider three variations on the opening synchronisation example:

Model A multiple passive actions versus a single active action
Model B multiple passive actions versus multiple active actions
Model C multiple active actions versus a single active action

Passage times are then extracted using the same stochastic probe from Figure 3.

5.1 Model A

$$\begin{aligned}
\mathbf{A} &\stackrel{\mathit{def}}{=} (\mathbf{run}, \lambda_1).(\mathbf{stop}, \lambda_2).\mathbf{A} \\
\mathbf{B} &\stackrel{\mathit{def}}{=} (\mathbf{run}, \top).(\mathbf{pause}, \lambda_3).\mathbf{B} \\
\mathbf{Sys_A} &\stackrel{\mathit{def}}{=} \mathbf{A} \underset{\{run\}}{\bowtie} (\mathbf{B} \mid\mid \mathbf{B})
\end{aligned}$$

Fig. 6. PEPA description for model A

Now we consider a small variation on our first model from Figure 6. Here we have two copies of the component $\mathbf{B}$. These act as clients of (the single) $\mathbf{A}$, competing for its $\mathbf{run}$ activity. Thus there are two possible synchronisations: $\mathbf{A}$ with the left-hand $\mathbf{B}$; and $\mathbf{A}$ with the right-hand $\mathbf{B}$. Each of these has one active participant and one passive participant. This synchronisation metaphor has been termed an *implicit choice* because although the PEPA choice operator has not been used in the model definition still there is a choice of different partners for the $\mathbf{run}$ activity.

In this setting, the apparent rate and the minimum rate will produce different results. We plot the two PDFs from these strategies in Figure 7 as we probe the duration between consecutive $\mathbf{stop}$ actions in the model. In Figure 8, we plot the percentage difference between these, as carried through to the CDF function: $100\psi(X_{ars}, X_{mrs}, t)$ for X_{ars} being the passage variable for the apparent rate strategy and X_{mrs} being the passage variable for the minimum rate strategy. We note that it is never more than 9% and that it falls off rapidly as time increases, dropping under 1% within 6 seconds.

5.2 Model B

We consider a third variant on this simple model; Model B in Figure 9. We introduce additional copies of the $\mathbf{A}$ component, which plays the role of the server in the model.

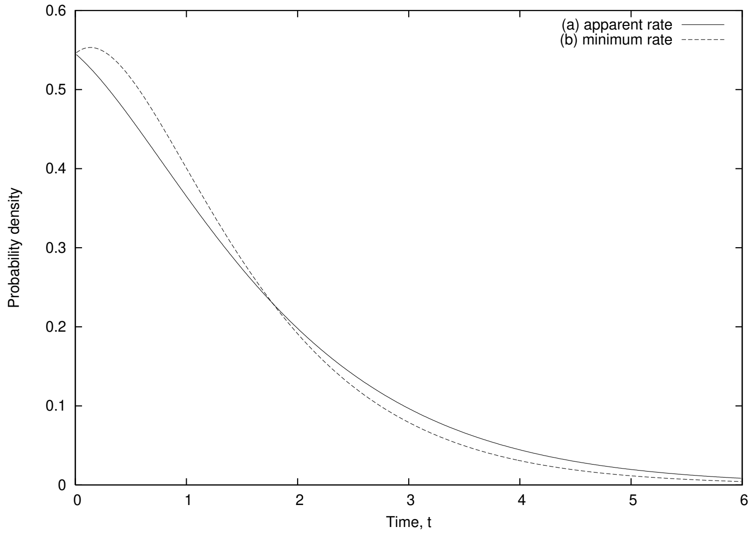

Fig. 7. Probability density functions for time taken between consecutive **stop** actions in model A for (a) the *apparent rate strategy* and (b) the *minimum rate strategy*

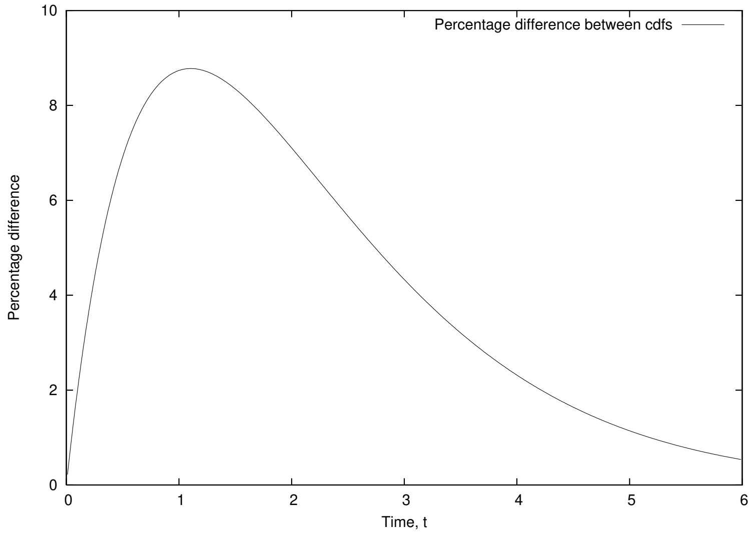

Fig. 8. Model A: Percentage difference from minimum rate strategy as carried through to the CDF

Now there are three servers (three copies of $\mathbf{A}$), which is a change, and two clients (two copies of $\mathbf{B}$), which is as it was in the previous version of the model. All of the servers are independent, as are all of the clients. Now there are six possible types of **run** action, with each of the $\mathbf{A}$s synchronising with each of the $\mathbf{B}$s.

$$\mathbf{A} \stackrel{\text{def}}{=} (\mathbf{run}, \lambda_1).(\mathbf{stop}, \lambda_2).\mathbf{A}$$
$$\mathbf{B} \stackrel{\text{def}}{=} (\mathbf{run}, \top).(\mathbf{pause}, \lambda_3).\mathbf{B}$$
$$\mathbf{Sys_B} \stackrel{\text{def}}{=} (\mathbf{A} \parallel \mathbf{A} \parallel \mathbf{A}) \underset{\{run\}}{\bowtie} (\mathbf{B} \parallel \mathbf{B})$$

Fig. 9. PEPA description for model B

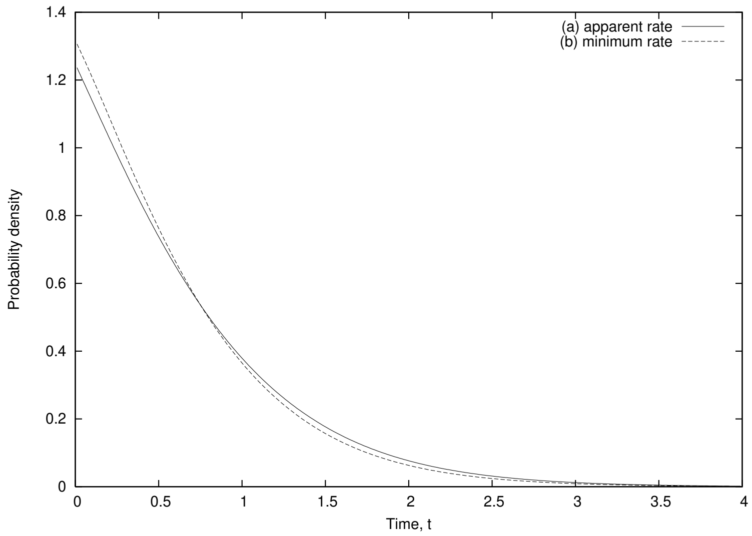

Fig. 10. Probability density functions for time taken between consecutive $\mathbf{stop}$ actions in model B for (a) an apparent rate strategy and (b) the *minimum rate strategy*

Again we plot the PDFs of both the model using the apparent rate computation and the model using the minimum rate (in Figure 10) and the percentage difference between these as carried through to the CDF (in Figure 11) as $100\psi(X_{ars}, X_{mrs})$. Here the agreement is even more encouraging and the results are even better. The difference between the two plots is never more than 6% and the difference reduces to zero even more quickly (within about 4 seconds).

5.3 Model C

There is a final PEPA synchronisation idiom which we have not considered, which is when both of the partners in a synchronisation contribute actively to the result. Such a model is Model C, in Figure 12. Now $\mathbf{B}$ is an active participant in the $\mathbf{run}$ action and is no longer a client of $\mathbf{A}$. Again the differences between the apparent rate calculation

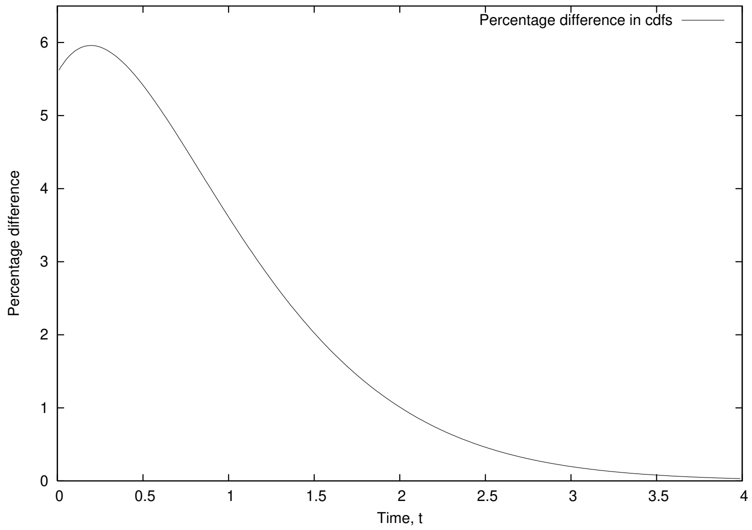

Fig. 11. Model B: Percentage difference between the *minimum rate strategy* and the *apparent rate strategy* as carried through to the CDF function

and the minimum rate calculation are computed (and plotted in Figures 13 and 14). The percentage difference between these is plotted in Figure 15. This time the calculations are repeated for five different values of the rate μ_1 while holding the value of $\lambda_1 = 1.0$ constant.

$$\mathbf{A} \stackrel{def}{=} (\mathbf{run}, \lambda_1).(\mathbf{stop}, \lambda_2).\mathbf{A}$$
$$\mathbf{B} \stackrel{def}{=} (\mathbf{run}, \mu_1).(\mathbf{pause}, \lambda_3).\mathbf{B}$$
$$\mathbf{Sys_C} \stackrel{def}{=} \mathbf{A} \underset{\{run\}}{\bowtie} (\mathbf{B} \mid\mid \mathbf{B})$$

Fig. 12. PEPA description for Model C

The difference between the two computations is greatest for slower rates (low values of μ_1) and least for faster rates (high values of μ_1). When μ_1 has the value 0.8 the difference between the two values is not more than 3% and again reduces to zero very quickly (within about 4 seconds) showing that even in this case the approximation provided by the minimum rate is acceptable. Not displayed is the passage difference, $\psi(\cdot)$, for $\mu_1 = 1.0$, which is 0 throughout, i.e. the strategies produce the same results when the synchronising rates are the same.

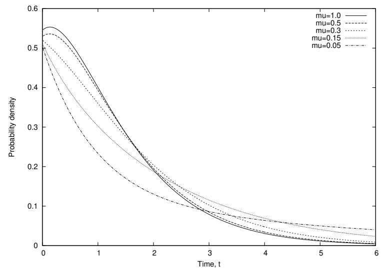

Fig. 13. Probability density functions for time taken between consecutive $\mathbf{stop}$ actions in Model C for $1.0 \geq \mu \geq 0.05$, using the *apparent rate strategy*

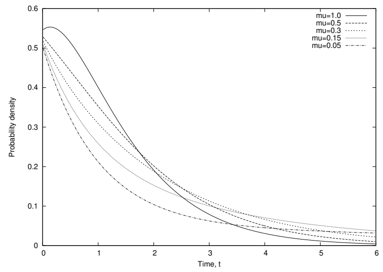

Fig. 14. Probability density functions for time taken between consecutive $\mathbf{stop}$ actions in Model C for $1.0 \geq \mu \geq 0.05$, using the *minimum rate strategy*

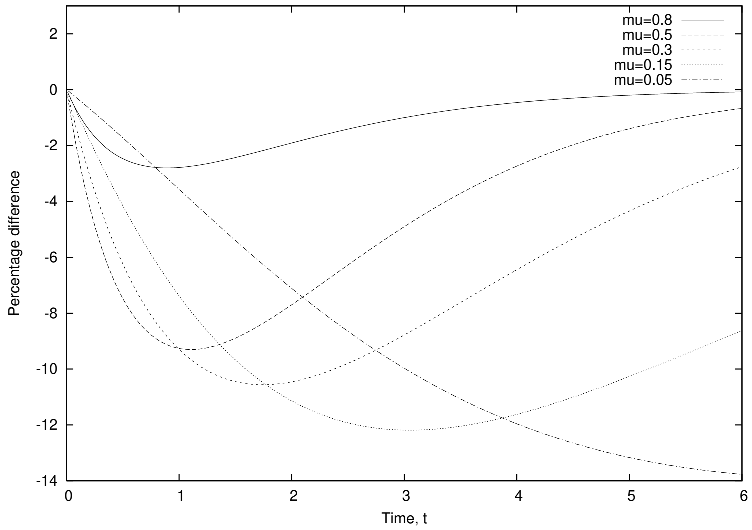

Fig. 15. Model C: Percentage difference between the *minimum rate strategy* and the *apparent rate strategy* as carried through to the CDF functions for 5 different values of μ_1

6 Conclusions

ipc [13] is a compiler for PEPA which carefully supports aspects of the PEPA language definition which have been approximated by other tools, in particular the crucial definition of apparent rate. ipc offers in addition the capability to specify and, with the aid of HYDRA [20], calculate passage time quantiles from PEPA models.

Using ipc, we have been able to compare different synchronisation strategies in PEPA. We have shown that when passive actions are involved, the minimum rate strategy tends to overestimate the global rate of action evolution. This translates into a slightly increased probability of early completion in passage time measures, as can be seen by the positive difference measure in Figures 8 and 11. Conversely, in active synchronisations, we have shown that the minimum strategy results in a slightly decreased probability of early completion.

It would appear that, except in cases where the Markov chain is stiff (has rates of orders of magnitude difference), the differences in strategy are neither too large nor too long-lasting. It should be borne in mind that we have deliberately taken worst case scenarios of synchronisations that reoccur very frequently and that in larger models where the synchronisations are more separated, the differences in performance metrics are likely to be very small. As future work, we intend to look at how synchronisation affects larger models of system behaviour.

User experience tells us that the computation of apparent rates is a noticeable overhead on state-space generation time and so a reasonable methodological approach might be to work with the approximation to the apparent rate computation for swift initial investigation of the problem before progressing to the more accurate (but more expensive) calculation later. In this way, reasonable performance results can be obtained quickly while the PEPA model is being developed and debugged and the performance modeller can progress to a more careful calculation of performance metrics after this initial development phase has ended.

Acknowledgements: Stephen Gilmore is supported by the DEGAS (Design Environments for Global ApplicationS) project IST-2001-32072 funded by the FET Proactive Initiative on Global Computing and by the EPSRC grant GR/S21717/01 (Enhancing the Performance Predictability of Grid Applications with Patterns and Process Algebras). Jeremy Bradley is supported in part by the Nuffield Foundation under grant reference NAL/00805/G.

References

1. Hillston, J.: A Compositional Approach to Performance Modelling. Volume 12 of Distinguished Dissertations in Computer Science. Cambridge University Press (1996)
2. Bernardo, M., Gorrieri, R.: Extended Markovian Process Algebra. In Montanari, U., Sassone, V., eds.: CONCUR'96, Proc. of the 7th Int. Conference on Concurrency Theory. Volume 1119 of LNCS., Springer-Verlag (1996) 315–330
3. Hermanns, H.: Interactive Markov Chains. PhD thesis, Univ. Erlangen-Nürnberg (1998)
4. Gilmore, S., Hillston, J.: The PEPA workbench: A tool to support a process algebra-based approach to performance modelling. In Haring, G., Kotsis, G., eds.: Proc. of the 7th Int. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation. Volume 794 of LNCS., Springer-Verlag (1994) 353–368
5. Clark, G., Gilmore, S., Hillston, J., Thomas, N.: Experiences with the PEPA performance modelling tools. In: UKPEW'98, Proceedings of the 14th UK Performance Engineering Workshop. (1998)
6. de Alfaro, L.: How to specify and verify the long-run average behaviour of probabilistic systems. In: Proc. of the 13th IEEE Symp. on Logic in Computer Science, IEEE (1998)

7. Bowman, H., Bryans, J.W., Derrick, J.: Analysis of a multimedia stream using stochastic process algebras. The Computer Journal **44** (2001) 230–245
8. El-Rayes, A., Kwiatkowska, M., Norman, G.: Solving infinite stochastic process algebra models through matrix-geometric methods. [24] 41–62
9. Wan, F.: Interface engineering and transient analysis for the PEPA Workbench. Master's thesis, School of Computer Science, The University of Edinburgh (2000)
10. Hermanns, H., Herzog, U., Hillston, J.: Stochastic process algebras—A formal approach to performance modelling. Tutorial, Dept. of Computer Science, Univ. of Edinburgh (1996)
11. Clark, G., Sanders, W.: Implementing a stochastic process algebra within the Möbius modeling framework. In de Alfaro, L., Gilmore, S., eds.: Proc. of the 1st joint PAPM-PROBMIV Workshop. Volume 2165 of LNCS., Aachen, Germany, Springer-Verlag (2001) 200–215
12. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic symbolic model checking with PRISM: A hybrid approach. In: TACAS'02, Proceedings of Tools and Algorithms for Construction and Analysis of Systems. Volume 2280 of Lecture Notes in Computer Science., Grenoble, Springer-Verlag (2002) 52–66
13. Bradley, J.T., Dingle, N.J., Gilmore, S.T., Knottenbelt, W.J.: Derivation of passage-time densities in PEPA models using ipc: the Imperial PEPA Compiler. In Kotsis, G., ed.: MASCOTS'03, Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, University of Central Florida, IEEE Computer Society Press (2003) 344–351
14. Milner, R.: Communication and Concurrency. Prentice Hall (1989)
15. Hillston, J.: The nature of synchronisation. In Herzog, U., Rettelbach, M., eds.: Proc. of the 2nd Int. Workshop on Process Algebras and Performance Modelling, Erlangen (1994) 51–70
16. Bradley, J.T., Davies, N.J.: Reliable performance modelling with approximate synchronisations. [24] 99–118
17. Götz, N., Herzog, U., Rettelbach, M.: TIPP—Introduction and application to protocol performance analysis. In König, H., ed.: Formale Beschreibungstechniken für verteilte Systeme. FOKUS, Saur-Verlag (1993)
18. Priami, C.: A stochastic π-calculus. In Gilmore, S., Hillston, J., eds.: Process Algebra and Performance Modelling Workshop. Volume 38(7) of Special Issue: The Computer Journal., CEPIS (1995) 578–589
19. Argent-Katwala, A., Bradley, J.T., Dingle, N.J.: Expressing performance requirements using regular expressions to specify stochastic probes over process algebra models. In Almeida, V., Lea, D., eds.: WOSP'04, Proceedings of the 4th International Workshop on Software and Performance, Redwood City, California, ACM (2004) 49–58
20. Bradley, J.T., Dingle, N.J., Gilmore, S.T., Knottenbelt, W.J.: Extracting passage times from PEPA models with the HYDRA tool: a case study. In Jarvis, S.A., ed.: UKPEW'03, Proceedings of 19th Annual UK Performance Engineering Workshop. (2003) 79–90
21. Harrison, P.G., Knottenbelt, W.J.: Passage-time distributions in large Markov chains. In Martonosi, M., e Silva, E.d.S., eds.: Proc. of ACM SIGMETRICS 2002. (2002) 77–85
22. Dingle, N.J., Harrison, P.G., Knottenbelt, W.J.: Response time densities in Generalised Stochastic Petri Net models. In: Proceedings of the 3rd International Workshop on Software and Performance (WOSP'2002), Rome (2002) 46–54
23. Dingle, N.J., Knottenbelt, W.J., Harrison, P.G.: HYDRA: HYpergraph-based Distributed Response-time Analyser. In Arabnia, H.R., Man, Y., eds.: PDPTA'03, Proceedings of the 2003 International Conference on Parallel and Distributed Processing Techniques and Applications. Volume 1., Las Vegas, NV (2003) 215–219
24. Hillston, J., Silva, M., eds.: PAPM'99, Proceedings of the 7th International Workshop on Process Algebra and Performance Modelling. In Hillston, J., Silva, M., eds.: Process Algebra and Performance Modelling Workshop, Centro Politécnico Superior de la Universidad de Zaragoza, Prensas Universitarias de Zaragoza (1999)