Performance Specification and Evaluation with Unified Stochastic Probes and Fluid Analysis

Richard A. Hayden, Jeremy T. Bradley, and Allan Clark

Abstract—Rapid and accessible performance evaluation of complex software systems requires two critical features: the ability to specify useful performance metrics easily and the capability to analyze massively distributed architectures, without recourse to large compute clusters. We present the *unified stochastic probe*, a performance specification mechanism for process algebra models that combines many existing ideas: state and action-based activation, location-based specification, many-probe specification, and immediate signaling. These features, between them, allow the precise and compositional construction of complex performance measurements. The paper shows how a subset of the stochastic probe language can be used to specify common response-time measures in massive process algebra models. The second contribution of the paper is to show how these response-time measures can be analyzed using so-called fluid techniques to produce rapid results. In doing this, we extend the fluid approach to incorporate immediate activities and a new type of response-time measure. Finally, we calculate various response-time measurements on a complex distributed wireless network of $O(10^{129})$ states in size.

Index Terms—Performance modeling, performance evaluation tools, stochastic process algebra, measurement probes, fluid approximation, passage-time analysis

1 Introduction

Performance modeling of large and complex computer systems has two significant aspects: *specification* of the model and the performance measure to be captured and *evaluation* of the often massive and sometimes intractable computation.

Furthermore, with many performance modeling formalisms, the feature to be measured, timed, counted, or verified has to be explicitly captured in the state space of the underlying model to enable performance analysis to take place. Examples of this include: the size of a buffer in a queue, the number of messages in a protocol exchange, and the number of times an energy cell has to cycle before it needs replacing. Often modelers have to be prescient about the features they wish to measure when constructing the model to allow these features to be made explicit in the model. Equally often, some performance measures only become apparently useful once the model has been constructed, by which time it can be harder to retrofit such features and a new model has to be constructed and verified. In this paper, we show that with the right query mechanism, performance models can have complex queries constructed about them which do not rely on key features being explicitly present in the model.

Just as significantly, except in rare situations where an analytic solution can be found, computing the actual performance measure can be extremely computationally expensive, requiring either massive numerical calculations or huge cluster-based parallel simulations. The ability to specify complex performance queries would be of only limited use without a supporting framework to facilitate their analysis in a timely fashion. We also show how probe-based queries can be analyzed using the latest fluid techniques based around differential equations. This allows us to produce performance results for the combined model-query specification using comparatively small computational resources.

We review how performance modeling of behavioral systems can be enhanced by observer processes, called *unified stochastic probes*, that probe for the start and finish of key performance metrics, in a similar style to that suggested by Wolf and Rosenblum [1].

In this paper, we bring together significant existing stochastic probe mechanisms to create the unified stochastic probe framework in Section 3. Although we base the unified stochastic probe around an enhanced version of the *PEPA* formalism (Section 2), it could be applied to any suitable behavioral performance modeling formalism. Our approach extends the original stochastic probe definition [2], which uses a regular expression specification to define the start and end events of a measure. Specifically, for the first time we bring together the following features under a unified measurement framework.

Immediate signaling permits measurements to be started and terminated precisely with an immediate transition extension to PEPA [3]. It also allows the modeler to have many slave probes signaling to a master probe to permit compound internal features to be measured.

Flexible location-based specification allows unified stochastic probes to be placed precisely within a complex model architecture using a pattern rewriting syntax [4].

- *R.A. Hayden and J.T. Bradley are with the Department of Computing, Imperial College London, Huxley Building, 180 Queen's Gate, London SW7 2BZ, United Kingdom. E-mail: {rh, jb}@doc.ic.ac.uk.*
- *A. Clark is with the School of Informatics, The University of Edinburgh, The Informatics Forum, 11 Crichton Street, Edinburgh EH8 9AB. E-mail: a.d.clark@ed.ac.uk.*

Manuscript received 14 Aug. 2010; revised 24 Oct. 2011; accepted 23 Dec. 2011; published online 11 Jan. 2012.
Recommended for acceptance by M. Kwiatkowska.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSE-2010-08-0252.
Digital Object Identifier no. 10.1109/TSE.2012.1.

Combined action- and state-based enabling allows further flexibility in defining both event-based and state-based specifications for performance measures as well as combinations of the two [5].

Finally, we provide a framework for the rapid analysis of very large distributed systems using unified stochastic probes, based on fast differential equation techniques. To achieve this, we construct a new fluid analysis framework for a stochastic process algebra that incorporates weighted immediate transitions (Section 4). This specifically requires new fluid action-counting and passage-time analysis that can make use of immediate transitions. In Section 4.2, we develop fluid passage-time analysis techniques which are able to cope with the significant additional expressiveness of the unified stochastic probe framework, providing a more general fluid passage time analysis [6]. Furthermore, for the first time we show how fluid techniques can be applied to analyze a new class of *transient individual passage time* (Section 4.2.2). Finally, in Section 5 we demonstrate the unified stochastic probe and fluid analysis techniques on a model of a massive distributed wireless network.

1.1 Related Work

Performance queries are often formed using logical specification languages such as *continuous stochastic logic (CSL)* [7]. CSL, although powerful, expresses only state-based measures based on the global state space of a model. With stochastic probes we seek to express measures that are defined by both state and event-based behavior. An extension, *aCSL* [8], allows constraints based on state predicates and sets of available actions in event-based systems. Later variants incorporate the ability to specify performance measures around both state-based and full automata-governed behavior, in *asCSL* [9] or CSL^{TA} [10]. In all cases, the CSL languages permit steady-state and passage time queries to be formed directly over labeled stochastic state spaces. However, usually the whole global state space of a model has to be explored (at least once) before a performance measure can be extracted. The key benefit of using fluid analysis is that we avoid the explicit state space exploration of the entire model. A more detailed comparison of the measure specification features of the unified stochastic probe framework with those of asCSL and $\mathrm{CSL}^{\mathrm{TA}}$ is made in Section 3.

A close performance-based analog to the stochastic probe lies in Woodside and Shramm [11], who developed the *NICE* system using finite state automata to observe trace data from simulation and experiment. On observing particular sequences of events, measurements would be started or terminated according to defined states in the observing automaton. By contrast, a stochastic probe is applied to a process algebraic model specification. So a probe can be placed at particular locations within the model, replicated many times across the model, and used to construct compound measurements via interprobe communication.

Popular performance tools have means of expressing performance queries. In particular, *DNAmaca* [12], *PRISM* [13], and *Möbius* [14] use a combination of direct state specification, CSL, and probabilistic logic-based specification to express required measures. These approaches all require full global state space exploration, although this can be alleviated by use of symbolic state space representation. With tools such as *PRISM* [13], *CASPA* [15], or *Möbius* [14] (in symbolic mode) which use symbolic *multiterminal binary decision diagrams (MTBDDs)* to store the stochastic transition system, model sizes of up to 10^{11} states can be analyzed (for steady-state measures). However, even this size can depend heavily on the model being studied and on detailed considerations such as the exact variable ordering in the underlying MTBDD and is still potentially overextended by a system of only 20 components with 10 states each.

1.2 Motivating Example

Consider a distributed application such as a wireless network. There are N_c clients which transfer data to each other, where N_c is potentially quite large. Each client is autonomous and subject to failure or power outage. Traditionally, verifying the performance properties of such a system requires construction of an underlying continuous-time Markov chain (CTMC) with an upper bound of d^{N_c} states if each client has d local states. A lumping aggregation [16] can be applied to reduce the size of the state space; however, the number of aggregate states is bounded below (generally very loosely) by $\exp(d)$ for large enough N_c. Even after state space aggregation, it is very easy to construct models many orders of magnitude larger than 10^9 states, which is typically too high to analyze explicitly.

There is clearly a requirement for analysis techniques that can cope with many thousands of interacting components in a complex performance model of, say, a wireless network, PubSub architecture, or Cloud environment. We present a new *fluid* analysis technique for a stochastic process algebra which has been extended to incorporate immediate activities in Section 4. The weighted immediate transition model used is inspired by that of *generalized stochastic Petri nets (GSPNs)* [17], which is sufficient for our needs of permitting instantaneous signaling between probe components. We recognize that immediate extensions to process algebras exist (such as *IMC* [18]) that provide more expressive bespoke scheduling of immediate activities but we do not require that level of functionality for our immediate signaling application.

It is possible to pick out response time measurements between explicit model states using formalisms like CSL or tool languages such as DNAmaca. Extracting derived behavior from performance models is harder, often requiring model modification to capture the behavior being measured.

An example of derived behavior can be seen in the wireless network model above when asking for the duration of four data transfers without a battery failure. The battery failure condition is a path restriction of the sort that is possible in CSL. However, the specification of four data transfers events when only a single data transfer action is explicitly defined in the process model is termed derived behavior and is harder to do in a stochastic logic or simple state specification formalism.

This is the type of more involved query that can be captured using unified stochastic probes. The probe itself is formally specified using a behavioral regular expression and a location formula. The probe is translated into an automaton and then to a process algebra component (in the target process language) which is attached to a part of the model (specified by the location formula) under model composition.

2 Immediate Grouped PEPA (GPEPA)

Grouped PEPA [19] is a simple extension of the stochastic process algebra PEPA [20], which facilitates the application of fluid-analysis techniques for massively parallel models. Immediate GPEPA (iGPEPA) adds immediate action functionality to the grouped PEPA formalism. A GPEPA model consists of a number of labeled cooperating component groups, each of which consists of a large number of components operating together in parallel. We refer to the components within these groups as *f-components*.[1] The f-components are those whose states are tracked explicitly by an approximating system of differential equations.

2.1 F-Components in Immediate GPEPA

F-components are allowed to be any standard PEPA process algebra component with the additional possibility of immediate actions (not available in traditional PEPA). We refer to this extension of PEPA as iPEPA. Immediate actions are necessary to facilitate communication between the measurement processes (defined in Section 3 by unified stochastic probes) which help us to specify passage-time measures of interest. Syntactically, an iPEPA component is specified by the standard PEPA grammar [20], augmented with immediate actions [3]:

$$\begin{aligned} S &::= (\alpha, r).S \mid [\mathrm{a}, w].S \mid S + S \mid \boldsymbol{C}_S \\ P &::= P \underset{L}{\bowtie} P \mid S \mid \boldsymbol{C}_P, \end{aligned} \tag{2.1}$$

where $\alpha \in \mathcal{A}_t$ is a *timed action type*, $\mathrm{a} \in \mathcal{A}_i$ is an *immediate action type*, $L \subseteq \mathcal{A}$, where $\mathcal{A} := \mathcal{A}_t \cup \mathcal{A}_i$ is the set consisting of all action types. Also, $r \in \mathbb{R}_+ \cup \{n\top | n \in \mathbb{Q}, n > 0\}$ is a rate parameter and $w \in \mathbb{R}_+$, $w \neq 0$ is a weight parameter.

In line with (2.1), an iPEPA component can be a purely *sequential component*, S, or a *parallel component*, P, with its own internal parallelism. $\boldsymbol{C}_S$ and $\boldsymbol{C}_P$ represent constants which denote sequential components or parallel components, respectively. The effect of this syntactic separation between constants is to allow cooperation between sequential components only.

We now introduce informally the intended semantics of the iPEPA syntax introduced above.

Prefix. The basic mechanism for describing the behavior of a system is to give a component a designated first action using the prefix combinator, denoted by a full stop. $(\alpha, r).P$ carries out an α-action whose duration is drawn from an exponential distribution with rate parameter r. It subsequently behaves thereafter as P.

Immediate prefix. The component $[\mathrm{a}, w].P$ behaves like the component $(\alpha, r).P$; however, activities enabled by an immediate prefix are *high priority* and are always performed instantaneously and before any timed activities that are currently enabled. The question of how to proceed in states with multiple concurrently enabled immediate actions is resolved probabilistically using the weight parameter. As mentioned, immediate prefix is particularly useful when constructing unified stochastic probes in Section 3. We also allow the syntax $\mathrm{a}.S$ as shorthand for $[\mathrm{a}, 1].S$ in situations for which a weighting is not necessary.

1. In other works on GPEPA, f-components were originally called *fluid components* [21], [22].

Choice. The component $P + Q$ represents a system which may behave either as P or as Q. The activities of both P and Q are enabled. If an activity in P completes first, the system then proceeds by taking on the behavior of the derivative of P following the completed action; and vice versa for Q.

Constant. It is convenient to be able to assign names to patterns of behavior associated with components. Constants are components whose meaning is given by a defining equation. The notation for this is $\boldsymbol{X} \stackrel{def}{=} P$. This allows the recursive definition of components, for example, $\boldsymbol{X} \stackrel{def}{=} (\alpha, r).\boldsymbol{X}$ performs α at rate r forever.

Cooperation. We write $P \underset{L}{\bowtie} Q$ to denote cooperation between P and Q over L. The set which is used as the subscript to the cooperation symbol, the *cooperation set* L, determines those action types on which the components are forced to synchronize. For action types not in L, the components proceed independently and concurrently with their enabled activities. We write $P \| Q$ as an abbreviation for $P \underset{\emptyset}{\bowtie} Q$, where P and Q execute independently in parallel.

Fundamental to PEPA (and thus iPEPA) is the notion of *apparent rate*, $r_\alpha(P)$, which measures the observed rate that an iPEPA component P executes a timed action α. This defines the rate that a cooperating process sees and is therefore integral to the speed of cooperation between processes. It is defined by

$$\begin{aligned} r_\alpha((\beta, \lambda).P) &:= \begin{cases} \lambda & \text{if } \beta = \alpha \\ 0 & \text{if } \beta \neq \alpha \end{cases} \\ r_\alpha([\mathrm{a}, w].P) &:= 0 \\ r_\alpha(P + Q) &:= r_\alpha(P) + r_\alpha(Q) \\ r_\alpha(P \underset{L}{\bowtie} Q) &:= \begin{cases} \min(r_\alpha(P),\ r_\alpha(Q)) & \text{if } \alpha \in L \\ r_\alpha(P) + r_\alpha(Q) & \text{if } \alpha \notin L. \end{cases} \end{aligned} \tag{2.2}$$

If a component enables an activity whose action type is in the cooperation set, it is not able to proceed with that activity until the other component also enables an activity of that type. The two components then proceed together to complete the *shared activity*. Once enabled, the rate of a shared *timed* activity has to be altered to reflect the slower component in a cooperation. Within the cooperation framework, we assume *bounded capacity*, that is, a component cannot be made to perform an activity faster by cooperation, and the rate of a shared timed activity is defined as the minimum of the apparent rates of the activity in the cooperating components.

In some cases, when the rate of a shared timed activity is determined by only one component in the cooperation, then the other component is defined as *passive* with respect to that activity. This means that the rate of the activity is left unspecified (denoted $\top$) and is determined upon cooperation by the rate of the activity in the other component. In defining f-components, we require that all passive actions are synchronized so as not to allow passive cooperation between component groups (as discussed in Section 2.2). Also, an iPEPA component is not allowed to offer the same timed action type both passively and actively (this is a standard restriction in PEPA). More information on the exact behavior of passive cooperation is given in Appendix A.1.

When immediate and timed actions are combined it is important to understand how they may interact. In particular, cooperation between timed and immediate

activities is not a well-defined concept. Therefore, we have disallowed cooperation between them by drawing their types from two disjoint sets, $\mathcal{A}_t$ and $\mathcal{A}_i$, respectively.

The formal structured operational semantics for PEPA without immediate actions can be found in [20, Chapter 3]. We present an explicit structured operational semantics for iPEPA which combines both timed and immediate actions in Appendix A.1. Our approach explicitly considers the removal of *vanishing states* at the level of f-components (Section 2.1.1), which is necessary if ordinary differential equation (ODE) techniques are to be applied. The operational semantics determines a transition relation on iPEPA components, consisting of both immediate and timed transitions. For an iPEPA component P, we define its set of *derivative states* $ds(P)$ to be all components reachable from it by either timed or immediate transitions according to the operational semantics.

2.1.1 Vanishing State Removal

Assuming that an iPEPA component satisfies two natural regularity conditions, its transition system has a straightforward translation to a continuous-time Markov chain, as in the case of PEPA without immediate actions. When an iPEPA component satisfies both of these conditions, we refer to it as *well behaved*. Specifically, the two regularity conditions are:

- *freedom from immediate cycles*, meaning that there are no cycles of immediate transitions in a well-behaved iPEPA component's transition system;
- *deterministic initial behavior*, meaning that there may be at most one path of immediate transitions emanating from a well-behaved iPEPA component's initial state.

We require that all f-components are well behaved. A more formal treatment of these conditions can be found in [22].

If an iPEPA component enables an immediate transition, then any timed transitions also enabled are ignored. If more than one immediate transition is enabled, the one to proceed is selected probabilistically according to the distribution as above. Otherwise, if a component enables only timed transitions, they are raced in the usual manner by sampling from exponential distributions according to the rates of the timed transitions.

This interpretation gives rise to a continuous-time Markov chain, but the state space of a well-behaved iPEPA component P is not $ds(P)$, rather it is the set of *nonvanishing derivative states*, $ds^*(P) \subseteq ds(P)$, the set of derivative states of P which do not enable any immediate actions. The immediate transitions emanating from vanishing derivative states can be removed if we replace paths of immediate transitions with the timed transition they determine between elements of $ds^*(P)$. The resulting transition system on $ds^*(P)$ is referred to as the *derived transition system* of P and is constructed formally in Appendix A.2.1. Note that this transition system *consists only of timed transitions.*

2.2 Immediate Grouped PEPA Models

A *component group* is a parallel cooperation of a normally large number of f-components. Syntactically, a component group, D, is specified by the following grammar:

$$D ::= D \between D \mid P, \tag{2.3}$$

where P is an *f-component*. The combinator $\between$ represents unsynchronized parallelism between f-components.

A *grouped PEPA model* is formed by combining multiple labeled component groups together. Syntactically, the grammar for a grouped PEPA model G is

$$G ::= G \underset{L}{\bowtie} G \mid Y\{D\}, \tag{2.4}$$

where Y is a group label, unique to each component group. The term $G \underset{L}{\bowtie} G$ represents synchronization over action types in the set L, where $L \subseteq \mathcal{A}_t$ is a set of timed action types. This component group structure defines a class of models to which fluid analysis is naturally applicable.

We illustrate more clearly how component groups and f-components are used together by constructing a standard PEPA model of a simple massively parallel client/server system with server initialization and failure phases.

$$\begin{array}{cc} \boldsymbol{Client}_0 \stackrel{def}{=} (fetch, r_t).\boldsymbol{Client}_1 & \boldsymbol{Client}_1 \stackrel{def}{=} (reset, r_s).\boldsymbol{Client}_0 \\ \boldsymbol{Serv}_0 \stackrel{def}{=} (initialize, r_i).\boldsymbol{Serv}_1 & \boldsymbol{Serv}_1 \stackrel{def}{=} (fetch, r_t).\boldsymbol{Serv}_0 \\ \boldsymbol{Serv}_2 \stackrel{def}{=} (recover, r_r).\boldsymbol{Serv}_0 & +(fail, r_f).\boldsymbol{Serv}_2 \end{array}$$

$$\boldsymbol{SC}(n,m) \stackrel{def}{=} \underbrace{(\boldsymbol{Client}_0 \| \cdots \| \boldsymbol{Client}_0)}_{n} \underset{\{fetch\}}{\bowtie} \underbrace{(\boldsymbol{Serv}_0 \| \cdots \| \boldsymbol{Serv}_0)}_{m}.$$

This model captures the scenario of n clients cooperating on a $fetch$ action with m servers. Each server has to initialize before synchronizing and a $fetch$ may fail at the server end throwing it into a failure mode, from which it can recover. The natural representation of this situation as an iGPEPA model has the structure

$$\boldsymbol{SC}(n,m) \stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{Client}_0[n]\} \underset{\{fetch\}}{\bowtie} \mathbf{Servers}\{\boldsymbol{Serv}_0[m]\},$$

where

$$P[k] := \underbrace{(P \between \cdots \between P)}_{k}.$$

This grouping unambiguously determines the components to be approximated using fluid analysis, $\boldsymbol{Client}_i$ and $\boldsymbol{Serv}_j$. So in this case, the fluid approximation is given by five coupled differential equations each of which counts the number of each type of component active in the model.

The combinator $\between$ has the same meaning as $\|$. However, the two distinct combinators are necessary to resolve possible ambiguity in the case of component groups which contain f-components with their own internal parallelism. That is, the purpose of the additional level of model structure afforded by iGPEPA models is to define the granularity at which the fluid approximation is performed, as is described in Section 4.1.

We have defined an operational semantics for iPEPA components in terms of an underlying CTMC in the previous section. The semantics for a complete iGPEPA model is a simple extension of this and is given formally in Appendix B.

3 Unified Stochastic Probes

Stochastic probes [2], [5] are a query mechanism for stochastic process algebras. They are used to specify events which determine the start and end of a desired passage-time measure.[2] That is, the stochastic probe specifies a set of *source states* and a set of *target states* in the state space of the underlying stochastic process. The passage time being measured is the delay in reaching any target state from a source state.

Unified stochastic probes are formalism independent and use a regular expression syntax, based on the complete action label set $\mathcal{A}$ of the model, combined with a state-based predicate to define the probe itself (similar to the asCSL [9] or $\mathrm{CSL}^{\mathrm{TA}}$ [10] automaton). Traditionally, a stochastic probe is translated into a fragment of stochastic process algebra that can be composed with a stochastic process algebra model.

Unlike asCSL or $\mathrm{CSL}^{\mathrm{TA}}$, however, a unified stochastic probe also contains a placement specification which locates the probe at a particular point in the compositional structure of the model. This allows the probe to be activated precisely by the exact component that the modeler is interested in measuring. Unified stochastic probes also permit many separate probes to be created and distributed over the model structure; the results of these *local* probes can in turn be observed via immediate signals by a global probe which defines a more sophisticated passage-time measure.

One particular benefit of stochastic probes over a stochastic logic like CSL [7] is that they allow the capture of derived behavior in a measure. Derived behavior is behavior that is not expressed directly in the state space of the original model, but is available in the product probe-model state-space.

As an example, the following is a probe that could be attached to the $\boldsymbol{SC}(n, m)$ model of Section 2.2 that emits, among others, $initialize$, $fetch$, and $fail$ actions:

$$\boldsymbol{Probe} \stackrel{def}{=} initialize : \mathrm{start}, (fetch[3]/fail) : \mathrm{stop}.$$

This specifies a probe that starts a measurement after seeing an $initialize$ action and terminates it on seeing three $fetch$ actions without seeing any $fail$ actions. The start and stop labels are reserved signals that start and stop the measurement.

3.1 Probe Syntax

In this paper, we use stochastic probes inspired by the definitions of [5] augmented with the unique iGPEPA component group labels. These are not available in the standard PEPA syntax, but they provide a convenient mechanism by which to identify specific components and to locate measurement probes. We build on and extend the notions of *local* and *global* probes as discussed in [5] to facilitate the application of fluid-analysis techniques.

For our purposes, *local* probes are probes which can be attached to f-components or their subcomponents and a *global* probe is a probe which observes the entire iGPEPA model. Local probes may communicate with other local probes and, ultimately, with a global probe by means of *signals*. In order to support this, we assume the existence of a set $\mathcal{S}$ of signal labels contained in the set of immediate action types $\mathcal{A}_i$. In Section 3.3, we introduce location specifications for local probes which define exactly which f-component or subcomponent they observe, that is, where they are placed within a model. Every probed model must include exactly one global probe but may utilize zero or many local probes.

3.1.1 Local Probes

A local probe is specified formally in terms of action observations and signal transmissions R_l^s and whether or not it is *repeating*, denoted by a $\hookleftarrow$ superscript:

$$Probe_l ::= R_l^s \mid R_l^{s\hookleftarrow}. \tag{3.1}$$

A repeating local probe simply loops back and repeats its measurement, transmitting a signal for each time it observes the same local pattern of behavior. The grammar R_l^s allows for signals ($signal \in \mathcal{S}$) to be transmitted in sequence following observations made by regular expressions:

$$\begin{array}{rll} R_l^s ::= & R_l^s, R_l^s & \text{sequence} \\ \mid & R_l : signal & \text{transmitted signal.} \end{array} \tag{3.2}$$

Transmitted signals can be observed by another local or a global probe. Indeed, sophisticated measurements can be specified by many probes attached to a system, in which case the signals can be used to start and stop measurements in higher level probes. The grammar R_l allows the full power of an extended regular-expression-based language, suited to passage-time specification:

$$\begin{array}{rl} R_l ::= & R_l, R_l \quad \text{sequence} \\ \mid & R_l \mid R_l \quad \text{choice} \\ \mid & R_l; R_l \quad \text{both} \\ \mid & R_l[n] \quad \text{iterate } n \text{ times} \\ \mid & R_l[m, n] \quad \text{iterate } m \text{ to } n \text{ times} \\ \mid & R_l^? \quad \text{zero or one} \\ \mid & R_l^+ \quad \text{one or more} \\ \mid & R_l^* \quad \text{zero or more} \\ \mid & R_l/R_l \quad \text{reset} \\ \mid & R_l \emptyset R_l \quad \text{fail} \\ \mid & R_l^! \quad \text{not} \\ \mid & . \quad \text{any action or signal} \\ \mid & \underline{action} \quad \text{eventual specific action or signal} \\ \mid & \overline{action} \quad \text{subsequent specific action or signal} \\ \mid & \epsilon \quad \text{empty action or signal sequence,} \end{array} \tag{3.3}$$

where $action \in \mathcal{A}$ is an action (or signal) type.

Since we are interested in capturing passage-time measurements which are started and stopped by signals, we wish to know *as soon* as the observed component has satisfied a regular expression which transmits a signal ($R_l : signal$ in (3.2)). For this reason, we interpret regular expressions preceding signals in a *lazy* or *minimal* manner—for a given trace of the observed component, we are only interested in how long it takes for the first match to occur, at which time the signal is transmitted. We are not interested in any subsequent matches, so, after a signal transmission, we start to match the next regular expression in the sequence $R_l^s,\ R_l^s$ (3.2) independently of the previous one.

2. They can also be used to specify other steady-state and transient measures [2], but we focus on the passage-time measure in this paper, analogous to the *time-bounded until* of a CSL formula.

The atoms of our regular expression grammar are the observed actions. We provide two ways to specify observed actions. A *subsequent specific action or signal*, written $\overline{action}$, matches $action$ only—the usual approach to specifying the atoms of a regular expression. On the other hand, an *eventual specific action or signal*, written $action$, matches any sequence of actions which includes an observation of an action of the given type. This is shorthand for

$$.^*, \overline{action}, .^*$$

where . is shorthand for $(\overline{a}_1 \mid \overline{a}_2 \mid \cdots \mid \overline{a}_{|\mathcal{A}(P)|})$ and $a_1, \ldots, a_{|\mathcal{A}(P)|} \in \mathcal{A}(P)$ is an enumeration of the set of action types $\mathcal{A}(P)$ performed by the component to be observed, say P (the details of how probes are attached to f-components follow in later sections).

We provide the shorthand for eventual observations since this is very often how passage-time queries are specified. The modeler may wish to state the key actions which are required for a passage to complete but is often not concerned further about the details of how this might come about. By way of example, the local probe specified by $\overline{fetch} : \mathrm{end}$ only transmits the end signal if the *first* action it observes is $fetch$. Otherwise, it *never* matches and thus never transmits the end signal. On the other hand, the local probe specified by $fetch : \mathrm{end}$ matches and transmits the end signal as soon as it sees a $fetch$ action, irrespective of the number of intervening actions it may have seen while waiting for it.

We have also introduced a few other nonstandard regular expression shorthands, particularly useful for succinct expression of passage-time measures. The *both* construction $R_l; R_l$ matches if both constituent regular expressions match. The *reset* construction R_l/R_l matches when the left-hand regular expression matches, subject to restarting the matching process every time the right-hand regular expression matches. Finally, the $fail$ construction $R_l \emptyset R_l$ is similar to the *reset* construction, but if the right-hand regular expression matches before the left-hand expression, the matching process fails entirely and the whole expression never matches.

A formal semantics is given for this regular expression language in Section 3.2 by means of the translation from probes to iPEPA components.

3.1.2 Global Probes

A global probe may observe actions and signals like a local probe, but can also directly query the state of the iGPEPA model to which it is attached. Furthermore, a global probe has the role of transmitting the two reserved signals start and stop, which serve to mark the beginning and end of the passage-time measure of interest. It serves no purpose to allow global probes to transmit any other signals so they are not allowed to do so. Formally, a global probe has a similar form to a local probe:

$$Probe_g ::= R_g : \mathrm{start}, R_g : \mathrm{stop} \mid R_g : \mathrm{start}, R_g : \mathrm{stop}^{\hookleftarrow}, \quad (3.4)$$

where R_g has the same grammar as R_l except for the addition of an optional state-based guard $\{pred\}R_g$, defined at the end of this section:

$$R_g ::= \{pred\}R_g \mid R_g, R_g \mid \cdots \mid action \mid \overline{action} \mid \epsilon. \quad (3.5)$$

Note that the grammar of $Probe_g$ gives two possible configurations. As in the case of local probes, a global probe can also be *repeating*, which is again specified by the $\hookleftarrow$ superscript. In the nonrepeating case, for a given trace of the probed model, both the start and stop signals are only ever transmitted at most once. The passage time being measured is then defined to be the difference between the time instants at which these two signals were transmitted.

On the other hand, repeating global probes generally transmit many pairs of start and stop signals in one trace of a probed model. The distribution of the difference between these two time instants is then conditional on the particular time at which the start signal was transmitted. Therefore, it makes sense to interpret repeating global probes as defining a steady-state measurement since the difference between the start and stop signal transmissions is identically distributed in the steady-state regime.

Finally, the state guard predicate, $pred$, is given by

$$\begin{array}{rll} pred ::= & \mathrm{true} \mid \mathrm{false} & \text{boolean} \\ \mid & \neg pred & \text{not} \\ \mid & pred \vee pred & \text{disjunction} \\ \mid & b\_expr & \text{expression} \\ b\_expr ::= & r\_expr \succeq r\_expr & \text{comparison} \\ r\_expr ::= & H : P & \text{component count} \\ \mid & int & \text{number} \\ \mid & r\_expr \oplus r\_expr & \text{arithmetic} \\ \succeq ::= & = \mid \geq \mid \leq & \text{relational ops} \\ \oplus ::= & + \mid - & \text{binary ops,} \end{array} \quad (3.6)$$

where, if G is the iGPEPA model to which the global probe is to be attached, $(H, P) \in \mathcal{B}(G)$.

The formal semantics for the predicate language for an iGPEPA model in state s is given by

$$\begin{array}{ll} s \models \mathrm{true} & \text{for all } s \\ s \models \mathrm{false} & \text{for no } s \\ s \models \neg\psi & \text{iff } s \not\models \psi \\ s \models \psi_1 \vee \psi_2 & \text{iff } s \models \psi_1 \vee s \models \psi_2 \\ s \models b\_expr & eval(b\_expr, s), \end{array}$$

where b_expr is, as above, a Boolean function of $H : P$ expressions, which reference a specific f-component derivative state P in the component group H. The *eval* function evaluates the Boolean function by dereferencing the $H : P$ expressions as the number of P components active in the component group H in the particular state of the model s.

A guarded regular expression $\{pred\}R_g$ matches the pattern of behavior specified by R_g subject to the state-based guard $\{pred\}$ being true when R_g starts to be matched.

3.2 Translating Probes to iPEPA

In this section, we describe how meaning is given to a probe-specified measure by defining a translation from its regular-expression specification to an iPEPA component. The resulting component can then be attached at various points within a performance model to extract the measure of interest. Where traditional CTMC analysis techniques are to be applied, all probes, both local and global, can be translated into iPEPA components and composed with the original iGPEPA model as specified. The resulting composite model

can then be analyzed as a stand-alone model using traditional techniques.

Where fluid analysis techniques are to be used, we translate local probes directly into iPEPA components and compose them with the model at the specified locations. The translation of global probes directly into process algebra components and their subsequent composition with the model of interest is, in fact, technically prohibited by our restriction that iGPEPA models may not cooperate on immediate actions across component groups. Indeed, the underlying reason for both decisions is the same. ODE techniques cannot be applied directly to such models since the synchronization on immediate actions at a global level leads to ill-defined systems of differential equations (see the discussion in [22, Section 3.2.1] for more details on this kind of problem). Instead, global probes are interpreted directly depending on the type of performance measure being computed as is described in Section 4.2.

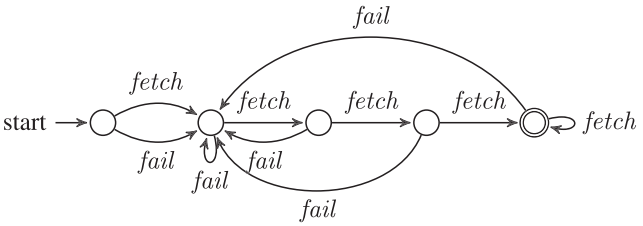
In this section, we discuss the probe-to-component translation and how a probe is attached to an f-component. In the next section, we introduce a mechanism which describes the placement of the local probes in the context of a complete iGPEPA model. We assume that P is the f-component to which the local probe in question is to be attached. Therefore, the set of atomic symbols over which the regular expressions are defined is the set of action types used by P, that is, $\mathcal{A}(P)$.

The translation of local probes to iPEPA components follows the standard approach of first converting the regular expression to a *deterministic finite automaton (DFA)* (e.g., [23]). A DFA is a labeled and directed graph with a single *start state* and a set of *accepting states*, interpreted as a *finite state machine* accepting finite strings of symbols. For each state, there is a directed transition to another state for each atomic symbol. Upon reading a symbol (that is, when a timed or immediate action occurs), the DFA jumps from its current state to another according to the transitions. The start state is where the matching process begins and the accepting states define when the DFA has successfully matched the regular expression. This DFA can then be translated in a straightforward manner to an iPEPA component which can be synchronized with the observed f-component. We begin by showing how a regular expression R_l given by the grammar of (3.3) may be translated to a DFA.

In fact, assuming that DFAs exist for regular expressions R_l^1 and R_l^2, the translation to DFAs of $(R_l^1,\ R_l^2)$, $(R_l^1 \mid R_l^2)$, $R_l^1[n]$, $R_l^1[m,n]$, $R_l^{1\,?}$, R_l^{1+}, and R_l^{1*} is standard and well documented in the literature (e.g., [23]). This is also the case for ϵ, ., $\overline{action}$, and $action$. The remaining cases $R_l^1;\ R_l^2$, R_l^1/R_l^2, $R_l^1 \emptyset R_l^2$, and $R_l^{1!}$ are less standard additional regular-expression constructions which are particularly useful for specifying certain types of passage-time measure. Their translation to DFAs is also relatively straightforward and is documented in Appendix C. By way of example, the local probe regular expression:

$$(fetch \mid fail), (fetch[3]/fail),$$

can be translated to the following DFA, assuming that it is to be attached to an f-component P, where $\mathcal{A}(P) = \{fetch, fail\}$ and the single accepting state is denoted by its double edge:



At this stage, DFA minimization algorithms (e.g., [23]) could be applied to reduce its size if desired. The next stage is the translation of the DFA to an iPEPA component, which is straightforward. Each state of the DFA maps to a derivative state of the iPEPA component. Each arc with a timed action is translated to a passive enabling of that action type in the iPEPA component, and each arc with an immediate action is translated to an immediate enabling of that action type. For the example above, the resulting iPEPA component is

$$\begin{aligned}
\boldsymbol{Pb} &\stackrel{def}{=} (fetch, \top).\boldsymbol{Pb}_1 + (fail, \top).\boldsymbol{Pb}_1\\
\boldsymbol{Pb}_1 &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_1 + (fetch, \top).\boldsymbol{Pb}_2\\
\boldsymbol{Pb}_2 &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_1 + (fetch, \top).\boldsymbol{Pb}_3\\
\boldsymbol{Pb}_3 &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_1 + (fetch, \top).\boldsymbol{Pb}_{\mathrm{Acc}}\\
\boldsymbol{Pb}_{\mathrm{Acc}} &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_1 + (fetch, \top).\boldsymbol{Pb}_{\mathrm{Acc}}.
\end{aligned}$$

A complete local probe may also transmit signals and possibly be repeating according to the grammars of (3.1), (3.2). Indeed, a general instance of the grammar of (3.2) has the form $R_l^1 : \mathrm{s}_1, \ldots, R_l^k : \mathrm{s}_k$. We may construct iPEPA components corresponding to each R_l^i following the methodology outlined above. Then since, as mentioned earlier, we wish for regular expressions directly preceding signals to be matched lazily, for each i all accepting states in the iPEPA component corresponding to R_l^i are merged into a single absorbing accepting state. This is achieved simply by choosing one accepting state (arbitrarily), then redirecting all transitions ending in other accepting states so that they instead end in the chosen one. Then, this single accepting state is made absorbing. Next, in the single accepting derivative state in the iPEPA component corresponding to R_l^i, we enable an immediate action of type s_i, which ends in the start state of the iPEPA component corresponding to R_l^{i+1}. This causes the probe component to emit an s_i signal before beginning to match the next regular expression in the sequence.[3]

If the complete local probe is repeating in the single accepting state in the iPEPA component corresponding to R_l^k, we enable an immediate action of type s_k which ends in the start state of the iPEPA component corresponding to R_l^1. Otherwise, in the nonrepeating case, this immediate action instead ends in a new derivative state which loops on all actions in $\mathcal{A}(P)$ (so as not to block the observed component under synchronization).

Now consider a slight modification of the earlier example to transmit signals at certain points within the matching process:

3. The modeler can avoid concurrently enabled immediate actions derived from signals by not using multiple signals to label the same event.

$$(fetch \mid fail) : \text{begin}, (fetch[3]/fail) : \text{end}.$$

The complete iPEPA component corresponding to this local probe is then:

$$\begin{aligned}
\boldsymbol{Pb} &\stackrel{def}{=} (fetch, \top).\boldsymbol{Pb}_{\text{begin}} + (fail, \top).\boldsymbol{Pb}_{\text{begin}} \\
\boldsymbol{Pb}_{\text{begin}} &\stackrel{def}{=} \text{begin}.\boldsymbol{Pb}_1 \\
\boldsymbol{Pb}_1 &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_1 + (fetch, \top).\boldsymbol{Pb}_2 \\
\boldsymbol{Pb}_2 &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_1 + (fetch, \top).\boldsymbol{Pb}_3 \\
\boldsymbol{Pb}_3 &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_1 + (fetch, \top).\boldsymbol{Pb}_{\text{end}} \\
\boldsymbol{Pb}_{\text{end}} &\stackrel{def}{=} \text{end}.\boldsymbol{Pb}_{\text{Acc}} \\
\boldsymbol{Pb}_{\text{Acc}} &\stackrel{def}{=} (fail, \top).\boldsymbol{Pb}_{\text{Acc}} + (fetch, \top).\boldsymbol{Pb}_{\text{Acc}}.
\end{aligned}$$

After its construction, we compose the local probe with the component to be observed using the standard iPEPA cooperation combinator. For the above example, this results in the f-component $P \underset{\mathcal{A}(P)}{\bowtie} \boldsymbol{Pb}$. If the f-component P happens to enable actions which might not be observed by the probe, say, for the example above that $\mathcal{A}(P) = \{fetch, fail, initialize\}$ instead, then the translation of the probe may involve a large number of self-loop transitions. If an action type is only enabled as a self-loop and this happens in *every* derivative state (apart from those just transmitting signals), we may remove all of the self-loops if we drop the action type in question from the cooperation set since this has exactly the same effect.

In future, we write probe composition as $P \underset{*}{\bowtie} Pb$, where $*$ is shorthand for cooperation over the intersection of $\mathcal{A}(P)$ and the set of observed actions or signals enabled by the probe iPEPA component, after self-loops have been removed where possible. Note that it is important that signals transmitted by the probe are not included in this cooperation set or they will be blocked. Therefore, it is a requirement that a single local probe cannot both observe and transmit the same signal. This clearly does not impact on the kinds of measurements we can specify since we can simply rename signals if necessary.

For certain types of local probe, there are situations where the f-component P enters a state from which it is no longer possible for one of the regular expressions R_l^i to match. For example, consider the local probe $\overline{fetch} : \text{begin}$. If P does not immediately perform a $fetch$ action, the regular expression never matches. If the local probe is repeating, we do not wish for such a situation to block the probe indefinitely, rather, we wish for the local probe to reset and repeat its (attempted) measurement. Thus, for repeating local probes, it is necessary to make a minor adjustment to the transition system of the probed component.

Specifically, consider the probed component $P \underset{*}{\bowtie} Pb$ where Pb is the iPEPA component resulting from the translation of some general repeating local probe $R_l^1 : \text{s}_1, \ldots, R_l^k : \text{s}_k \overset{\hookleftarrow}{}$. For each i, let $\mathcal{R}^i$ be the set of derivative states of Pb corresponding to the DFA states of R_l^i and write Pb^i_{Acc} for the corresponding accepting state of R_l^i. Then, we must consider any derivative state $Q \underset{*}{\bowtie} R \in ds(P \underset{*}{\bowtie} Pb)$ for which $R \in \mathcal{R}^i$ from which it is no longer possible to reach a state where R_l^i has accepted, that is, to reach some $S \underset{*}{\bowtie} Pb^i_{\text{Acc}}$ for $S \in ds(P)$. Then, we modify the transition system so that all transitions into such states $Q \underset{*}{\bowtie} R$ are redirected to the state $Q \underset{*}{\bowtie} Pb$ in order to reset the probe.

We note that if P is an f-component and thus well behaved, then so is $P \underset{*}{\bowtie} Pb$ for any reasonable local probe Pb. More formally, as long as the local probe is not a repeating local probe which makes no observations of actions or signals, $P \underset{*}{\bowtie} Pb$ is free from immediate cycles and has deterministic initial behavior.

Finally, we note that it is straightforward to automate the translation process described above starting from the initial regular expression specification of the local probe. Therefore, the modeler need not interact with the evaluation process further after specifying the regular expression defining the measurement.

3.3 Locating a Local Probe

To locate a local probe, we use a modified version of the model transformation language presented by Clark and Gilmore [4], extended to use the component group labels for the purpose of uniquely locating the probe. Thus, in the $\boldsymbol{SC}(n, m)$ model of Section 2.2, which was defined by

$$\boldsymbol{SC}(n, m) \stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{Client}_0[n]\} \underset{L}{\bowtie} \mathbf{Servers}\{\boldsymbol{Serv}_0[m]\},$$

we might want to position a local probe, $\boldsymbol{Pb}$ around one of the $\boldsymbol{Client}_0$ components in the **Clients** group. To do this we can deploy a model transformation:

$$\begin{aligned}
&\mathbf{Clients}\{\boldsymbol{Client}_0[?n]\} \Longrightarrow \\
&\qquad \mathbf{Clients}\{(\boldsymbol{Client}_0 \underset{*}{\bowtie} \boldsymbol{Pb}) \wr\wr \boldsymbol{Client}_0[?n-1]\}.
\end{aligned}$$

This is used to pattern match against the definition of $\boldsymbol{SC}(n, m)$ and transform the matching part from the form to the left of the $\Longrightarrow$ to the form on the right. In this case the rule matches the $\mathbf{Clients}\{\cdot\}$ group structure in the system and adds the probe in the relevant place. The formal variable $?n$ matches whatever number or symbolic representation of replicated $\boldsymbol{Client}_0$ is specified. It permits subsequent arithmetic operation to be performed on $?n$ in the transformed model.

Formally, we draw the syntax of a location specification term from the transformation language defined by $rule$ below:

$rule$	::=	$group \Longrightarrow group$	replacement rule
$group$	::=	$H\{cpts\}$	component group
	|	$group \underset{actions}{\bowtie} group$	group cooperation
$cpts$	::=	cpt	f-component
	|	$cpt \wr\wr cpt$	f-component parallelism
	|	$cpt[expr]$	f-component array
cpt	::=	P	iPEPA component
	|	$cpt \underset{actions}{\bowtie} cpt$	iPEPA cooperation
	|	$?p$	iPEPA component variable
$actions$	::=	L	action set
	|	$?l$	action set variable
$expr$	::=	int	number
	|	$?n$	numeric variable
	|	$expr \oplus expr$	binary expression,

where $H \in \mathcal{G}(G)$, $P \in \mathcal{B}(G, H')$ for some $H' \in \mathcal{G}(G)$, $L \subseteq \mathcal{A}_t$, $int \in \mathbb{Z}_+$, and n and p are drawn from a set of variable names.

As a final example of probe placement, consider a situation in which $\boldsymbol{Serv}_0$ components operate in pairs, synchronizing on the $fetch$ action, perhaps to deliver the service to the client, represented by the following iGPEPA model:

$$\boldsymbol{SC}'(N, M) \stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{Client}_0[N]\} \underset{L}{\bowtie} \mathbf{Servers}\{(\boldsymbol{Serv}_0 \underset{\{fetch\}}{\bowtie} \boldsymbol{Serv}_0)[M]\}.$$

Perhaps we are interested in monitoring the behavior of one such pair. To do this, we need to employ nested local probes to allow us to distinguish between actions performed by the two partners in a pair. For example, we might use the following model transformation to attach a local probe to each server in a specific pair, and then a master probe to combine their observations:

$$\mathbf{Servers}\{(\boldsymbol{Serv}_0 \underset{\{fetch\}}{\bowtie} \boldsymbol{Serv}_0)[?n]\} \Longrightarrow \mathbf{Servers}\{(((\boldsymbol{Serv}_0 \underset{*}{\bowtie} \boldsymbol{Pb}_1) \underset{\{fetch\}}{\bowtie} (\boldsymbol{Serv}_0 \underset{*}{\bowtie} \boldsymbol{Pb}_2)) \underset{*}{\bowtie} \boldsymbol{Pb}_m) \between (\boldsymbol{Serv}_0 \underset{\{fetch\}}{\bowtie} \boldsymbol{Serv}_0)[?n-1]\}.$$

Nested local probes are supported automatically since an f-component becomes another f-component under probe composition.

It is important to realize that the translation and composition of local probes with an iGPEPA model as described above simply results in another iGPEPA model. In particular, existing techniques for the analysis of iGPEPA models can then be directly applied to the resulting probed model.

Finally, we discuss how a complete measure specification consisting of a model, a set of zero or more local probes, and a single global probe can be represented. Formally, a probed model has the following grammar:

$$\begin{aligned} ProbedModel ::= {} & Probe_g \text{ observes } \{Probe_l\} \\ & \text{where } \{Location\} \\ & \text{in } G \\ & \mid Probe_g \text{ in } G, \end{aligned} \tag{3.7}$$

where $\{Probe_l\}$ is a list of local probes and $\{Location\}$ is a list of location specifications used to place the local probes within the iGPEPA model G.[4] We assume that the location specification transformations are applied in the order of enumeration of this list.

3.4 Example Measure Specifications

In this paper, we focus on the analysis of two classes of passage-time measure that are amenable to fluid analysis and thus allow massive state space models to be analyzed. The first is the *individual passage time*—what is the probability that an individual component in a group has completed some task by time t. The second is the *global passage time*—what is the probability that, globally, the model has completed some passage by time t. In this section, we provide a few such example measurements, specified in the language of unified stochastic probes. Then, in the next section, we proceed to discuss how fluid-analysis techniques may be applied for their analysis.

Recall the simple iGPEPA model from Section 2.2.

$$\begin{aligned} \boldsymbol{Client}_0 &\stackrel{def}{=} (fetch, r_t).\boldsymbol{Client}_1 & \boldsymbol{Client}_1 &\stackrel{def}{=} (reset, r_s).\boldsymbol{Client}_0 \\ \boldsymbol{Serv}_0 &\stackrel{def}{=} (initialize, r_i).\boldsymbol{Serv}_1 & \boldsymbol{Serv}_1 &\stackrel{def}{=} (fetch, r_t).\boldsymbol{Serv}_0 \\ \boldsymbol{Serv}_2 &\stackrel{def}{=} (recover, r_r).\boldsymbol{Serv}_0 & &\quad +(fail, r_f).\boldsymbol{Serv}_2 \end{aligned}$$

$$\boldsymbol{SC}(n, m) \stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{Client}_0[n]\} \underset{\{fetch\}}{\bowtie} \mathbf{Servers}\{\boldsymbol{Serv}_0[m]\}.$$

A global passage-time query using state-based specification on this model might ask *how long before n of the servers are simultaneously in state $\boldsymbol{Serv}_1$*. This is captured by the probed model specification:

$$\begin{aligned} \boldsymbol{PM}_1 \stackrel{def}{=} \epsilon : \text{start}, \{\text{Servers} : \boldsymbol{Serv}_1 \geq n\} : \text{stop} \\ \text{in} \quad \boldsymbol{SC}(n, m). \end{aligned}$$

We might instead be interested in the related query *how long before n $fail$-actions have been performed by servers*, which is captured by the probed model specification:

$$\begin{aligned} \boldsymbol{PM}_2 \stackrel{def}{=} \epsilon : \text{start}, fail[n] : \text{stop} \\ \text{in} \quad \boldsymbol{SC}(n, m). \end{aligned}$$

No local probes or location specifications are required in either of the above cases. However, where derived behavior needs to be captured, activity-based probes and location specifications become key mechanisms. Assume now that we are instead interested in the query *how long before n* different *servers each perform a $fetch$-action*. In order to capture this behavior, we need to employ a local probe which is attached to each $\boldsymbol{Serv}_0$ component:

$$\begin{aligned} \boldsymbol{PM}_3 \stackrel{def}{=} {} & \epsilon : \text{start}, \text{end}[n] : \text{stop} \\ & \text{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} fetch : \text{end} \\ & \text{where} \quad \mathbf{Servers}\{\boldsymbol{Serv}_0[?n]\} \Longrightarrow \\ & \qquad \mathbf{Servers}\{(\boldsymbol{Serv}_0 \underset{*}{\bowtie} \boldsymbol{Probe}_l)[?n]\} \\ & \text{in} \quad \boldsymbol{SC}(n, m). \end{aligned}$$

In this case, the local probe is necessary to extract the derived behavior from each $\boldsymbol{Serv}_0$ component of having *already completed a $fetch$-action*. After observing this it signals to the global probe using the end signal.

Transferring our attention to individual passage-time queries, we might simply be interested in *how long it takes a particular server to complete its first $fetch$-action*. In order to specify this measurement we may again use a local probe similar to that above to capture the derived behavior of having completed a $fetch$-action, but we apply it to only one $\boldsymbol{Serv}_0$ component:

4. Where there is only one local probe or location specification, we drop the $\{\cdot\}$ list notation.

$$\begin{aligned}\boldsymbol{PM}_4 \stackrel{def}{=}\ & \text{begin} : \text{start}, \text{end} : \text{stop}\\ & \text{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \epsilon : \text{begin}, fetch : \text{end}\\ & \text{where} \quad \mathbf{Servers}\{\boldsymbol{Serv}_0[?n]\} \Longrightarrow\\ & \qquad \mathbf{Servers}\{(\boldsymbol{Serv}_0 \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\!\wr \boldsymbol{Serv}_0[?n-1]\}\\ & \text{in} \quad \boldsymbol{SC}(n,m).\end{aligned}$$

Note that it is *not* true that the same measurement could be specified with just the global probe $\epsilon : \text{start}, fetch : \text{stop}$ since this terminates on seeing *any* $fetch$-action from *any* server. Thus, the local probe is necessary. If we were looking to measure a more complicated individual passage-time such as *how long it takes a particular server to complete two* $fetch$*-actions without failing*, we need a more involved local probe, which again ensures that both of the $fetch$-actions and any $fail$-actions are captured only from a single $\boldsymbol{Serv}_0$ component:

$$\begin{aligned}\boldsymbol{PM}_5 \stackrel{def}{=}\ & \text{begin} : \text{start}, \text{end} : \text{stop}\\ & \text{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \epsilon : \text{begin}, (fetch[2])/fail : \text{end}\\ & \text{where} \quad \mathbf{Servers}\{\boldsymbol{Serv}_0[?n]\} \Longrightarrow\\ & \qquad \mathbf{Servers}\{(\boldsymbol{Serv}_0 \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\!\wr \boldsymbol{Serv}_0[?n-1]\}\\ & \text{in} \quad \boldsymbol{SC}(n,m).\end{aligned}$$

We might instead be interested in the same individual passage-time measurement taken in the steady state and starting after a server has recovered, in which case we could use the following specification:

$$\begin{aligned}\boldsymbol{PM}_6 \stackrel{def}{=}\ & \text{begin} : \text{start}, \text{end} : \text{stop}^{\hookleftarrow}\\ & \text{observes}\\ & \quad \boldsymbol{Probe}_l \stackrel{def}{=} recover : \text{begin}, (fetch[2])/fail : \text{end}^{\hookleftarrow}\\ & \text{where} \quad \mathbf{Servers}\{\boldsymbol{Serv}_0[?n]\} \Longrightarrow\\ & \quad \mathbf{Servers}\{(\boldsymbol{Serv}_0 \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\!\wr \boldsymbol{Serv}_0[?n-1]\}\\ & \text{in} \quad \boldsymbol{SC}(n,m).\end{aligned}$$

The global probe $\text{begin} : \text{start}, \text{end} : \text{stop}^{\hookleftarrow}$ used here is repeating and thus specifies a steady-state measurement. The local probe starts measuring when it sees a $recover$-action and then times until the observation of two $fetch$-actions without an intervening $fail$-action, as above.

In the next two sections, we investigate the fluid analysis approximation of individual and global passage times specified using unified stochastic probes. We use these techniques to analyze both classes of passage-time measure as applied to a worked example, in Section 5.

4 Fluid Analysis of Immediate GPEPA Models

In this section, we define a fluid analysis for iGPEPA models. This entails generating a set of coupled ordinary differential equations which count the number of f-components in a given state. This is an extension of the derivation of ordinary differential equations from standard GPEPA models [19] to cope with immediate activities. Also new to this paper, we require a set of ODEs that count the number of times that a particular action has occurred during the evolution of the system, so-called *action-counting ODEs*.

TABLE 1
Frequently Used iGPEPA Notation, Where C_i Is Short for $\boldsymbol{Client}_i$ and S_j Is Short for $\boldsymbol{Serv}_j$ from the $\boldsymbol{SC}(n,m)$ Model

$\mathcal{B}(G,H)$	The union of all non-vanishing derivative states of all f-components in the component group of G which has group label H, e.g. $\mathcal{B}(\mathbf{SC}(n,m),\mathbf{C}) = \{\mathbf{C}_0,\mathbf{C}_1\}$.
$\mathcal{B}(G)$	The set of all pairs whose first element is a component group label, say, H and whose second is an element of $\mathcal{B}(G,H)$, e.g. $\mathcal{B}(\mathbf{SC}(n,m)) = \{(\mathbf{C},\mathbf{C}_0),(\mathbf{C},\mathbf{C}_1),(\mathbf{S},\mathbf{S}_0),(\mathbf{S},\mathbf{S}_1),(\mathbf{S},\mathbf{S}_2)\}$.
$\mathcal{S}(G,H)$	The size of the component group with label H. That is, the number of f-components in the group, e.g. $\mathcal{S}(\mathbf{SC}(n,m),\mathbf{C}) = n$.

All of these differential equations form the basis for the fluid approximation of passage times from stochastic probe definitions. In order to do this, we first define some key functions on an iGPEPA model (Table 1).

The quantities which are subject to the fluid approximation are exposed formally through an aggregation of an iGPEPA model's nonvanishing state space. Considering $\boldsymbol{SC}(n,m)$ again, we see there are $n \times m$ different ways the initial shared $fetch$ action can be performed. This is because the $fetch$ action involves exactly one $\boldsymbol{C}_0$ and one $\boldsymbol{S}_0$ component. Each of these transitions occurs at rate $\frac{r_t}{nm}\min(n,m)$. The aggregation collects states together based on the number of f-components in each derivative state in each component group. In the case of $\boldsymbol{SC}(n,m)$, we might represent the initial aggregate state informally as "$n \times \boldsymbol{C}_0$, $0 \times \boldsymbol{C}_1$, $m \times \boldsymbol{S}_0$, $0 \times \boldsymbol{S}_1$, and $0 \times \boldsymbol{S}_2$ components." All of the $n \times m$ $fetch$-transitions then become a single transition from the aggregate state "$n \times \boldsymbol{C}_0$, $0 \times \boldsymbol{C}_1$, $m \times \boldsymbol{S}_0$, $0 \times \boldsymbol{S}_1$, and $0 \times \boldsymbol{S}_2$ components" to the aggregate state "$(n-1) \times \boldsymbol{C}_0$, $1 \times \boldsymbol{C}_1$, $(m-1) \times \boldsymbol{S}_0$, $1 \times \boldsymbol{S}_1$, and $0 \times \boldsymbol{S}_2$ components" at an aggregate rate of $r_t \min(n,m)$. The general extension of this aggregation process constructs an *underlying aggregated CTMC* from a given iGPEPA model (as originally constructed for PEPA [16] and GPEPA [19]).

In the case of GPEPA (i.e., without immediate actions), it has been shown [19, Theorem 2.12] that the underlying CTMC of a GPEPA model can always be aggregated in this way. Each state of the underlying aggregated CTMC of a GPEPA model, G, can be uniquely determined by the model's initial state and a function $E \in \mathcal{B}(G) \to \mathbb{Z}_+$. This function counts the number of f-components in each derivative state currently active in a given component group.

When f-components are allowed to perform immediate actions, this aggregation result extends directly. This is because cooperation on immediate actions between component groups is not allowed, so no vanishing state of an f-component can become nonvanishing under composition as part of a grouped model and vice versa. Therefore, when we later refer to transitions between f-component derivative states, written as $P \xrightarrow{(\alpha,\lambda),\cdot}$, we are referring to the transition system obtained after vanishing states have been removed (the *derived transition system*).

4.1 Differential Equations Associated with an iGPEPA Model

We require differential equations which describe the evolution of large process models in order to evaluate particular classes of unified stochastic probe query (as described in Section 3.4). In particular, we need ODEs which describe both the number of a particular component at a time t and the number of actions of a particular type that have occurred by a time t.

An iGPEPA model G has an integer-valued stochastic process $N_{H,P}(t)$ which counts the number of f-components in the nonvanishing derivative state P, active at a given time $t \geq 0$ within the component group H. We intend to define, by means of a system of ODEs, real-valued deterministic functions $v_{H,P}(t)$ as approximations to the stochastic processes $N_{H,P}(t)$.

Definition 4.1 (Component-counting ODEs for an iGPEPA model). *We define the evolution of the $v_{H,P}(t)$ over time for $(H,P) \in \mathcal{B}(G)$ for an iGPEPA model, G, by the system of first-order coupled ODEs:*

$$\dot{v}_{H,P}(t) = \underbrace{\sum_{\alpha \in \mathcal{A}_t} \left(\sum_{Q \in \mathcal{B}(G,H)} p_\alpha(Q,P)\, \mathcal{R}_\alpha(G, V_t, H, Q) \right)}_{(1)} - \underbrace{\sum_{\alpha \in \mathcal{A}_t} \mathcal{R}_\alpha(G, V_t, H, P)}_{(2)}, \tag{4.1}$$

for all $(H,P) \in \mathcal{B}(G)$ and where for $t \in \mathbb{R}_+$, $V_t \in \mathcal{B}(G) \to \mathbb{R}_+$ is given by $V_t(H,P) := v_{H,P}(t)$. The initial conditions, $V_0 \in \mathcal{B}(G) \to \mathbb{R}_+$ are those naturally defined by the initial state of G.

Part (2) of (4.1) specifies the total rate at which P-components evolve in the group H. This is computed by summing over all timed action types α, the overall rate at which P components perform an α-action in H in the model G using the other component counting differential equations provided by V_t. Formally, this quantity is given by the *component rate* function $\mathcal{R}_\alpha(G, V_t, H, P)$, whose definition follows.

Definition 4.2 (Component rate). *Let G be a iGPEPA model. For $(H,P) \in \mathcal{B}(G)$, timed action type $\alpha \in \mathcal{A}_t$ and $V_t \in \mathcal{B}(G) \to \mathbb{R}$, the component rate function is defined as follows:*

$$\mathcal{R}_\alpha(M_1 \underset{L}{\bowtie} M_2, V_t, H, P) := \begin{cases} \dfrac{\mathcal{R}_\alpha(M_i, V_t, H, P)}{r_\alpha(M_i, V_t)} \min(r_\alpha(M_1, V_t), r_\alpha(M_2, V_t)) \\ \quad if\ \alpha \in L \text{ and } H \in \mathcal{G}(M_i),\ for\ i = 1 \text{ or } 2 \\ \mathcal{R}_\alpha(M_i, V_t, H, P) \\ \quad if\ \alpha \notin L \text{ and } H \in \mathcal{G}(M_i),\ for\ i = 1 \text{ or } 2 \end{cases}$$

$$\mathcal{R}_\alpha(Y\{D\}, N, H, P) := \begin{cases} V_t(H,P)\, r_\alpha(P) & if\ H = Y \text{ and } P \in \mathcal{B}(G,H) \\ 0 & otherwise. \end{cases}$$

Terms with zero-valued denominators are defined to be zero.

The component rate function requires an iGPEPA version of the PEPA apparent rate function (2.2) defined in terms of component counts.

Definition 4.3 (Count-oriented apparent rate). *Let G be an iGPEPA model. Let $\alpha \in \mathcal{A}_t$ be a timed action type and $V_t \in \mathcal{B}(G) \to \mathbb{R}_+$. Then, the apparent rate is defined as follows:*

$$r_\alpha(M_1 \underset{L}{\bowtie} M_2, V_t) := \begin{cases} \min(r_\alpha(M_1, V_t), r_\alpha(M_2, V_t)) & if\ \alpha \in L \\ r_\alpha(M_1, V_t) + r_\alpha(M_2, V_t) & otherwise \end{cases}$$

$$r_\alpha(Y\{D\}, V_t) := \sum_{P \in \mathcal{B}(Y\{D\}, Y)} V_t(Y, P)\, r_\alpha(P).$$

Part (1) of (4.1) specifies the total rate at which components evolve into P-components in H. In order to compute this, in addition to the component rate function we require the *derivative weighting function* $p_\alpha(P,Q)$. This computes the probability that on performing an α-action P evolves into the component Q. The formal definition follows.

Definition 4.4 (Derivative weighting function). *Let G be an iGPEPA model and $H \in \mathcal{G}(G)$ a component group label. Let $P, Q \in \mathcal{B}(G,H)$ be f-component derivative states and let $\alpha \in \mathcal{A}_t$ be a timed action type. Then, $p_\alpha(P,Q) := \frac{1}{r_\alpha(P)} \sum_{P \xrightarrow{(\alpha,\lambda)} Q} \lambda$. This is defined to be zero when $r_\alpha(P) = 0$.*

We need action-counting ODEs in order to evaluate global probes which count the number of actions that have occurred by time t. For some action type (timed or immediate) $a \in \mathcal{A}$, let $N_a(t)$ count the number of a-transitions which have occurred up to time t. The following ODEs define deterministic approximations, $v_a(t)$, to the counting processes, $N_a(t)$.

Definition 4.5 (Action-counting ODEs for an iGPEPA model). *We define the evolution of $v_a(t)$ for the iGPEPA model G over time for $a \in \mathcal{A}$ by the first-order coupled ODE:*

$$\dot{v}_a(t) = r_a(G, V_t).$$

The initial conditions are $v_\alpha(0) := 0$ for all $\alpha \in \mathcal{A}$. If initial states of f-components in G are themselves vanishing, then we must set $v_{\mathrm{a}}(0)$ to take any immediate transitions which occur instantaneously into account.

The intuition behind Definition 4.5 is that the ODE approximation of $N_a(t)$ should increase at the apparent rate of a actions observed in the model G at time t.

Where $a \in \mathcal{A}_t$ is a timed action, the count-oriented apparent rate for an iGPEPA model was given above in Definition 4.3. However, where $\mathrm{a} \in \mathcal{A}_i$ is an immediate action, we have to extend the definition of count-oriented apparent rate to cover immediate actions, $\mathrm{a} \in \mathcal{A}_i$:

$$r_{\mathrm{a}}(G, V_t) := \sum_{\alpha \in \mathcal{A}_t} \sum_{k \in \mathbb{Z}_+} k \times r_{\alpha,(k,\mathrm{a})}(G, V_t),$$

where $r_{\alpha,(k,\mathrm{a})}(G, V_t)$ is the apparent rate at which k immediate actions of type a instantaneously follow a timed α-action.

Definition 4.6 (Count-oriented immediate apparent rate). *We define $r_{\alpha,(k,\mathrm{a})}(G, V_t)$ on an iGPEPA model G where $\alpha \in \mathcal{A}_t$ is a timed action type, $k \in \mathbb{Z}_+$, $\mathrm{a} \in \mathcal{A}_i$ is an immediate action type, and $V_t \in \mathcal{B}(G) \to \mathbb{R}_+$.*

$$r_{\alpha,(k,\mathrm{a})}(M_1 \underset{L}{\bowtie} M_2, V_t) := \begin{cases} \displaystyle\sum_{i+j=k} \frac{r_{\alpha,(i,\mathrm{a})}(M_1, V_t)}{r_\alpha(M_1, V_t)} \frac{r_{\alpha,(j,\mathrm{a})}(M_2, V_t)}{r_\alpha(M_2, V_t)} & \\ \min(r_\alpha(M_1, V_t), r_\alpha(M_2, V_t)) & \text{if } \alpha \in L \\ r_{\alpha,(k,\mathrm{a})}(M_1, V_t) + r_{\alpha,(k,\mathrm{a})}(M_2, V_t) & \text{otherwise} \end{cases}$$

$$r_{\alpha,(k,\mathrm{a})}(Y\{D\}, V_t) := \sum_{P \in \mathcal{B}(Y\{D\}, Y)} V_t(Y, P) \sum_{P \xrightarrow[\#(\mathrm{a} \in \mathcal{I}) = k.]{(\alpha,\lambda),\mathcal{I}}} \lambda.$$

4.2 Fluid Analysis of Unified Stochastic Probe Queries

In this section, we show how a selection of important passage-time measures specified using unified stochastic probes can be analyzed using the fluid-analysis techniques developed for iGPEPA.

4.2.1 Steady-State Individual Passage Times

A steady-state individual passage-time measurement on the model G is encoded by a probe specification of the form:

$$\begin{aligned} &\mathrm{begin} : \mathrm{start}, \mathrm{end} : \mathrm{stop}^{\hookleftarrow} \\ &\text{observes} \quad Pb \stackrel{def}{=} R_l : \mathrm{begin}, R_l : \mathrm{end}^{\hookleftarrow} \\ &\text{where} \quad H\{P[?n]\} \Longrightarrow H\{(P \underset{*}{\bowtie} Pb) \wr P[?n-1]\} \\ &\text{in} \quad G. \end{aligned} \tag{4.2}$$

The location specification applies the local probe Pb to the first f-component in a group H of identical components since we are interested in monitoring a single individual. The global probe delegates control to the local probe by means of the begin and end signals.

In order to compute the CDF associated with this passage time, we need to consider two different probed versions of G. $\tilde{G}$ represents the model G in composition with Pb after application of the location specification in (4.2). Analysis of $\tilde{G}$ captures the steady-state distribution which is used to initialize the passage-time calculation. Therefore, in order for this passage-time measure to be meaningful, we assume further that the underlying CTMC of the model $\tilde{G}$ has a unique stationary distribution.[5] $\tilde{G}'$ is constructed in the same way as $\tilde{G}$ but the composition is with Pb', an absorbing version of the local probe, thus $Pb' \stackrel{def}{=} R_l : \mathrm{begin}, R_l : \mathrm{end}$. The model $\tilde{G}'$ is used to capture the evolution of the passage time itself.

The computation of the passage time by fluid techniques follows the mathematical development of [6] (see also Hayden's thesis [22]); however, the required structural modifications (previously very tricky to specify even for simple query specifications) as expressed more powerfully using probes are completely new.

5. Note that this could be implied by irreducibility of the CTMC underlying $\tilde{G}$ or, alternatively, if it is reducible but has a single communicating class.

Performing fluid analysis on these models produces the quantities $\tilde{v}_{Y,Q}(t)$ for $\tilde{G}$ and $\tilde{v}'_{Y,Q}(t)$ for $\tilde{G}'$ for each component Q in each group Y. The steady-state quantities $\tilde{v}_{Y,Q}$ are given by the long-time limits $\lim_{t\to\infty} \tilde{v}_{Y,Q}(t)$, which are usually found by computing a fixed point.[6] Note that since we are interested in individual passage times, the probed models $\tilde{G}$ and $\tilde{G}'$ have only one copy of their respective probed component. Therefore, the ODE solutions $\tilde{v}_{H,Q}(t)$ and $\tilde{v}'_{H,Q}(t)$ for probed component derivative states $Q \in ds^*(P \underset{*}{\bowtie} Pb)$ or $Q \in ds^*(P \underset{*}{\bowtie} Pb')$, respectively, capture the *probability* that the observed individual is in state Q at time t. At first glance this seems to contradict the requirement for accurate fluid analysis that all components be present in large numbers. However, as detailed in Hayden's thesis [22], fluid convergence still holds since the probed component exists within a large population of identically behaving (but unprobed) P components.

We are now in a position to write down the fluid computation of the CDF for the steady-state individual passage time of (4.2), obtained simply by summing over the approximated probabilities of the absorbing probe being in its accepting state, say Pb_{Acc}:

$$F_{\mathrm{si}}(t) := \sum_{C \in - \underset{*}{\bowtie} Pb_{\mathrm{Acc}}} \tilde{v}'_{H,C}(t). \tag{4.3}$$

The initial conditions for the set of ODEs defining $\tilde{v}'_{Y,Q}(t)$ are computed using the steady-state quantities given by $\tilde{v}_{Y,Q}$. Specifically, in order to capture the passage-time quantity of (4.2) correctly, we need to compute the steady-state distribution of the probed f-component *immediately after the* begin *signal has fired*. The probability that the probed component is in state $Q \in ds^*(P \underset{*}{\bowtie} Pb)$ immediately after a begin signal fires, is the expected rate of transitions where a begin signal fires which also result in the probed component entering the state Q, divided by the total expected rate of all transitions firing begin signals (e.g., [24, Definition 96 and Proposition 97]). The fluid analysis approximation of this quantity is therefore:

$$\frac{\sum_{\alpha \in \mathcal{A}_t, C \in ds^*(P \underset{*}{\bowtie} Pb), C \xrightarrow{(\alpha,\lambda),(\ldots,\mathrm{begin},\ldots)} Q} \left(\frac{\mathcal{R}_\alpha(\tilde{G}, \tilde{V}, H, C)\lambda}{r_\alpha(C)} \right)}{\sum_{\alpha \in \mathcal{A}_t, C \in ds^*(P \underset{*}{\bowtie} Pb), C \xrightarrow{(\alpha,\lambda),(\ldots,\mathrm{begin},\ldots)} } \left(\frac{\mathcal{R}_\alpha(\tilde{G}, \tilde{V}, H, C)\lambda}{r_\alpha(C)} \right)}, \tag{4.4}$$

where $\tilde{V} \in \mathcal{B}(\tilde{G}) \to \mathbb{Z}_+$ is defined by $\tilde{V}(Y, Q) := \tilde{v}_{Y,Q}$ for all $(Y, Q) \in \mathcal{B}(\tilde{G})$.

So the initial condition $\tilde{v}'_{H,Q}(0)$ for a probed component derivative state $Q \in ds^*(P \underset{*}{\bowtie} Pb')$ is the conditioned quantity computed above in (4.4) (and is zero if $Q \notin ds^*(P \underset{*}{\bowtie} Pb)$). For all other $(Y, R) \in \mathcal{B}(\tilde{G}')$, the initial value $\tilde{v}'_{Y,R}(0)$ is set to $\tilde{v}_{Y,R}$.

Another interpretation of steady-state individual passage times. In this section, we briefly discuss an alternative interpretation of a measure specification of the form of (4.2). In particular, instead of attaching the repeating local probe

6. This system of ODEs usually has infinitely many fixed points, but normally only one that is meaningful in the context of the original model, that is, for example, where the total component population at the fixed point is correct. This is discussed in more detail in Hayden [22, Chapters 4 and 5].

$R_l : \text{begin}, R_l : \text{end}^{\hookleftarrow}$ to the initial state of the model and considering the stationary regime of the composed model $\tilde{G}$, we can instead attach the (nonrepeating) local probe $R_l : \text{begin}, R_l : \text{end}$ at some random time in the stationary regime of the uncomposed model G and then compute the passage time from this point. This is closer to how steady-state passage times specified in CSL^{TA} are interpreted [25].

To see that there can be a difference, consider the simple f-component $\boldsymbol{P}_0$ given by

$$\boldsymbol{P}_0 \stackrel{def}{=} (a, r).\boldsymbol{P}_1 \qquad \boldsymbol{P}_1 \stackrel{def}{=} (a, r).\boldsymbol{P}_2$$
$$\boldsymbol{P}_2 \stackrel{def}{=} (b, r).\boldsymbol{P}_0.$$

Attaching the local probe $a : \text{begin}, b : \text{end}^{\hookleftarrow}$ to $\boldsymbol{P}_0$ and performing the analysis as in the last section means that the passage we time is the evolution $\boldsymbol{P}_1 \stackrel{a}{\to} \boldsymbol{P}_2 \stackrel{b}{\to} \boldsymbol{P}_0$. If, however, we allow $\boldsymbol{P}_0$ to evolve in its stationary regime and then attach the local probe $a : \text{begin}, b : \text{end}$ at some random time, then depending on the state of the observed component at this time, either the passage $\boldsymbol{P}_1 \stackrel{a}{\to} \boldsymbol{P}_2 \stackrel{b}{\to} \boldsymbol{P}_0$ or $\boldsymbol{P}_2 \stackrel{b}{\to} \boldsymbol{P}_0$ is timed, meaning that the resulting passage-time distribution is a weighted combination of the two. The reason for this difference is that in the interpretation discussed in the previous section, the local probe evolves concurrently with the observed component from the start and the passage that is measured is thus potentially dependent on the model's initial state.

It turns out that we may adapt the fluid analysis method of the last section straightforwardly to compute a steady-state individual passage time under this alternative interpretation. In this case, we are interested in two iGPEPA models: the unprobed model G and the model with the absorbing local probe attached, say $\tilde{G}$. To begin, we compute the long-time ODE limits of the stationary component counts $(Y, Q) \in \mathcal{B}(G)$ of the unprobed model G, say $\tilde{v}_{Y,Q}$. We can then initialize the ODEs for the probed model, say $\tilde{G}$, as follows:

$$\tilde{v}_{H,R \underset{*}{\bowtie} Pb'}(0) := \frac{v_{H,R}}{\mathcal{S}(G, H)}$$

for $R \in ds^*(P)$,[7] $\tilde{v}_{Y,Q}(0) := v_{Y,Q}$ for $(Y, Q) \in \mathcal{B}(G)$, and all other ODE components are set to zero. This captures the state of the probed model at the instant that the local probe is attached in the stationary regime.

Similarly to the last section, we then use the solutions to these ODEs $\tilde{v}_{Y,Q}(t)$ to approximate the steady-state distribution of the probed component immediately after the begin signal has fired. Write $\mathcal{P}$ for the set of all probed component derivative states $Q \in ds^*(R \underset{*}{\bowtie} Pb')$ where $R \in ds^*(P)$. Then, for $Q \in \mathcal{P}$, this distribution can be computed by conditioning on the time at which the begin signal fires:

$$\int_0^\infty S^s(Q) \times \frac{\mathrm{d}K(s)}{\mathrm{d}s}\, \mathrm{d}s, \tag{4.5}$$

where $S^s(Q)$ is the conditional distribution given that the begin signal fired at time $s > 0$,[8] defined similarly to (4.4) in the previous section:

$$S^s(Q) := \frac{\sum_{\alpha \in \mathcal{A}_t, C \in \mathcal{P}, C \xrightarrow{(\alpha,\lambda),(\ldots,\text{begin},\ldots)} Q} \left(\frac{\mathcal{R}_\alpha(\tilde{G}, \tilde{V}_s, H, C)\lambda}{r_\alpha(C)} \right)}{\sum_{\alpha \in \mathcal{A}_t, C \in \mathcal{P}, C \xrightarrow{(\alpha,\lambda),(\ldots,\text{begin},\ldots)}} \left(\frac{\mathcal{R}_\alpha(\tilde{G}, \tilde{V}_s, H, C)\lambda}{r_\alpha(C)} \right)},$$

where $\tilde{V}_s$ encodes the $\tilde{v}_{Y,Q}(s)$ similarly to the previous section. The function $K(s) := \sum_{Q \in \mathcal{Q}} \tilde{v}_{H,Q}(s)$ approximates the probability that the probed component has begun the passage by time s, where $\mathcal{Q} \subseteq \mathcal{P}$ is the set of all states which the probed component can be in *after the* begin signal has been transmitted.

Finally, the passage-time CDF is given by the same formula as in the previous section:

$$F_{\text{si}}(t) := \sum_{C \in - \underset{*}{\bowtie} Pb_{\text{Acc}}} \tilde{v}'_{H,C}(t),$$

where the ODE solutions $\tilde{v}'_{Y,Q}(t)$ are again obtained from the ODEs associated with the probed model $\tilde{G}$ but this time with the initial condition for $Q \in \mathcal{P}$, $\tilde{v}'_{H,Q}(0)$ given by the conditioned quantity of (4.5). For all other $(Y, R) \in \mathcal{B}(G)$, the initial value $\tilde{v}'_{Y,R}(0)$ is set to $v_{Y,R}$.

We consider just the interpretation of the previous section in the remainder of this paper since it is the usual one for stochastic probes [3].

4.2.2 Transient Individual Passage Times

A transient individual passage-time measurement on the model G is encoded by a probe specification of the form:

$$\begin{aligned} &\text{begin} : \text{start}, \text{end} : \text{stop} \\ &\text{observes} \quad Pb \stackrel{def}{=} R_l : \text{begin}, R_l : \text{end} \\ &\text{where} \quad H\{P[?n]\} \Longrightarrow H\{(P \underset{*}{\bowtie} Pb) \wr P[?n-1]\} \\ &\text{in} \quad G. \end{aligned} \tag{4.6}$$

This is identical to the steady-state individual passage-time specification of the last section except that the global probe is nonrepeating and thus specifies a transient measure.

In the evaluation of this kind of passage time, we need only consider one probed version of G since all calculations here are transient. Like before, $\tilde{G}$ represents the model G in composition with Pb. $\tilde{G}$ is used in two ways. First, analysis of $\tilde{G}$ captures the transient distribution at each time $s \geq 0$, which is used to initialize the passage time calculation, conditional on it starting at time s. Second, $\tilde{G}$ is also used to capture the evolution of the conditional passage time itself. Finally, in (4.8), we average over all such conditional passage-time computations to obtain the actual passage time of interest.

We are interested in a number of different fluid-analysis solutions to the ODEs constructed from $\tilde{G}$. The quantities $\tilde{v}_{Y,Q}(t)$ are the solutions started from the initial conditions of the model. On the other hand, for each $s \geq 0$, the quantities $\tilde{v}^s_{Y,Q}(t)$ are the solutions obtained by using appropriate

7. For local probes of the form $\epsilon : \text{begin}, R_l : \text{end}$, $R \underset{*}{\bowtie} Pb'$ is vanishing, so in such cases we set instead $\tilde{v}_{H,R \underset{*}{\bowtie} S}(0) := \frac{v_{H,R}}{\mathcal{S}(G,H)}$, where S is the nonvanishing state of Pb' following the transmission of the begin signal.

8. The quantity $S^0(Q)$ only has an effect on the value of the integral for local probes of the form $\epsilon : \text{begin}, R_l : \text{end}$. In such cases, it is given simply by $S^0(Q) := \tilde{v}_{H,Q}(0)$.

initial conditions derived from the values $\bar{v}_{Y,Q}(s)$, as is discussed shortly.

We can now write down the fluid computation of the CDF *conditional* on the passage having started at time s:

$$F^s_{\mathrm{ti}}(t) := \sum_{C \in - \underset{*}{\bowtie} Pb_{\mathrm{Acc}}} \tilde{v}^s_{H,C}(t). \tag{4.7}$$

Then, the unconditional CDF is recovered by normalizing over all times $s \geq 0$:

$$F_{\mathrm{ti}}(t) := \int_0^\infty F^s_{\mathrm{ti}}(t) \times \frac{\mathrm{d}K(s)}{\mathrm{d}s}\, \mathrm{d}s. \tag{4.8}$$

Like in the last section, the function $K(s) := \sum_{Q \in \mathcal{Q}} \tilde{v}_{H,Q}(s)$ approximates the probability that the observed individual has begun the passage by time s. Again, $\mathcal{Q}$ is the set of all states $P \underset{*}{\bowtie} Pb$ can be in *after the* begin signal has been transmitted.

The initial conditions for the set of ODEs defining $\tilde{v}^s_{Y,Q}(t)$ for each $s \geq 0$ are computed from the solutions $\tilde{v}_{Y,Q}(s)$ in a similar manner to the previous sections. Specifically, the fluid approximation of the probability that the probed component is in state $Q \in ds^*(P \underset{*}{\bowtie} Pb)$ immediately after a begin signal fires at time $s > 0$ is given by:[9]

$$\frac{\sum_{\alpha \in \mathcal{A}_t, C \in ds^*(P \underset{*}{\bowtie} Pb), C \xrightarrow{(\alpha,\lambda),(\ldots,\mathrm{begin},\ldots)} Q} \left(\frac{\mathcal{R}_\alpha(\tilde{G}, \tilde{V}_s, H, C)\lambda}{r_\alpha(C)} \right)}{\sum_{\alpha \in \mathcal{A}_t, C \in ds^*(P \underset{*}{\bowtie} Pb), C \xrightarrow{(\alpha,\lambda),(\ldots,\mathrm{begin},\ldots)} } \left(\frac{\mathcal{R}_\alpha(\tilde{G}, \tilde{V}_s, H, C)\lambda}{r_\alpha(C)} \right)}, \tag{4.9}$$

where $\tilde{V}_s$ encodes the $\tilde{v}_{Y,Q}(s)$ similarly to the previous sections.

So the initial condition $\tilde{v}^s_{H,Q}(0)$ for a probed component derivative state $Q \in ds^*(P \underset{*}{\bowtie} Pb)$ is the conditioned quantity computed above in (4.9). For all other $(Y, R) \in \mathcal{B}(\tilde{G})$, the initial value $\tilde{v}^s_{Y,R}(0)$ is set to $\tilde{v}_{Y,R}(s)$.

4.2.3 Global Passage Times

Global passage-time measurements consider the joint behavior of an entire population of f-components. Local behavior is picked out by first attaching local probes to f-components and the evolution of many f-components in the model is then considered by specifying a suitable global probe. A transient global passage-time measurement on the model G is encoded by a probe specification of the form:

$$\begin{array}{l} \epsilon : \mathrm{start}, R_g : \mathrm{stop} \\ \mathrm{observes} \quad \{Probe_l\} \\ \mathrm{where} \quad \{Location\} \\ \mathrm{in} \quad G, \end{array} \tag{4.10}$$

where $\{Probe_l\}$ is a list of local probes and $\{Location\}$ is a list of location specifications. There is no restriction on the form of the local probes. However, while it is possible to analyze the full expressiveness of individual passage-time measures using fluid analysis, we are not in quite such a fortunate position with global probes. Fluid analysis is currently capable of analysing global probes R_g specifying global passage-time measures of the form:

$$\begin{array}{rl} R_g ::= & \{pred\}R_g \mid R_g, R_g \mid R_g \mid R_g \\ & \mid R_g; R_g \mid R^a_g[n], \end{array} \tag{4.11}$$

where

$$R^a_g ::= R^a_g \mid R^a_g \mid action,$$

and $action \in \mathcal{A}$ is an action (or signal) type.

It should be noted that global passage times that fall outside of this pattern can still be analyzed as long as the state space is small enough. In such cases, a general global probe (as defined by the complete grammar of (3.5)) can be translated to an iGPEPA component following Section 3.2 and then attached to the iGPEPA model to be observed. Traditional Markov chain analysis or simulation techniques can then be applied directly to the resulting composite model.

The first step in the fluid analysis of a global passage time is to apply the local probes in accordance with the location specifications for the measurement. We assume this has been done and that we are left with a model that contains the integrated local probes $\tilde{G}$. In order to apply fluid techniques to compute the global passage-time measurement, we proceed by translating the global probe to an equivalent symbolic mathematical expression in terms of the fluid approximations to the action and component counts, $v_a(t)$ and $v_{H,P}(t)$, respectively, over the integrated model $\tilde{G}$. We deploy a function $\mathcal{U}(R, e)$ which calculates the point-mass approximation to the distribution of a global passage time specified by the probe R starting from time e. So computing $\mathcal{U}(R, 0)$ in terms of the fluid approximations gives the desired passage-time approximation.

Given R as a global regular expression of the type (4.11). $\mathcal{U}(R, 0)$ is the equivalent mathematical expression we seek, where $\mathcal{U}(R, e)$ is defined by

$$\begin{aligned} \mathcal{U}((R_1, R_2), e) &:= \mathcal{U}(R_2, \mathcal{U}(R_1, e)) \\ \mathcal{U}((R_1 \mid R_2), e) &:= \min(\mathcal{U}(R_1, e), \mathcal{U}(R_2, e)) \\ \mathcal{U}((R_1; R_2), e) &:= \max(\mathcal{U}(R_1, e), \mathcal{U}(R_2, e)) \\ \mathcal{U}(\{pred\}R, e) &:= \mathcal{U}(R, \inf\{t \geq e : pred(t)\}), \end{aligned}$$

where $pred(t)$ is simply the state guard predicate where each component count expression, say $H : P$ for $(H, P) \in \mathcal{B}(G)$, is replaced with the corresponding sum of potentially probed component counts at time t, $\sum_{Q \in P \underset{*}{\bowtie} \_} v_{H,Q}(t)$. Finally, we have

$$\mathcal{U}(R^a[n], e) := \inf\{t \geq e : \mathcal{U}'(R^a, e, t) \geq n\},$$

for $\mathcal{U}'(a_1 \mid \cdots \mid a_k, e, t) := \sum_{i=1}^k v_{a_i}(t) - \sum_{i=1}^k v_{a_i}(e)$.

4.3 Example Fluid Passage-Time Calculation

Before proceeding to investigate a more complicated worked example in the next section, we illustrate here an example probe specification and its associated fluid analysis for a simple passage-time measure on the client-service model of Section 2.2.

Specifically, we consider the example measurement specification $\boldsymbol{PM}_6$ given earlier in Section 3.4:

9. Like in the last section, the quantity of (4.9) evaluated at $s = 0$ only has an effect on the value of the integral for local probes of the form $\epsilon : \mathrm{begin}, R_l : \mathrm{end}$. In such cases, it is given simply by $\tilde{v}_{H,Q}(0)$.

$$\begin{aligned}\boldsymbol{PM}_6 \stackrel{def}{=} & \text{begin} : \text{start}, \text{end} : \text{stop}^{\hookleftarrow} \\ & \text{observes} \\ & \quad \boldsymbol{Probe}_l \stackrel{def}{=} recover : \text{begin}, (fetch[2])/fail : \text{end}^{\hookleftarrow} \\ & \text{where} \quad \mathbf{Servers}\{\boldsymbol{Serv}_0[?n]\} \Longrightarrow \\ & \quad \mathbf{Servers}\{(\boldsymbol{Serv}_0 \underset{*}{\bowtie} \boldsymbol{Probe}_l) \between \boldsymbol{Serv}_0[?n-1]\} \\ & \text{in} \quad \boldsymbol{SC}(n,m).\end{aligned} \tag{4.12}$$

This is the steady-state individual passage-time measurement of *how long, after recovery, it takes a particular server to complete two fetch-actions without failing*.

Following Section 4.2.1, the first step in the analysis is to translate the local probe $\boldsymbol{Probe}_l$ to an iPEPA component, say $\boldsymbol{Pb}_S$, following Section 3.2:

$$\begin{aligned}\boldsymbol{Pb}_S &\stackrel{def}{=} (fetch, \top).\boldsymbol{Pb}_S + (fail, \top).\boldsymbol{Pb}_S + (recover, \top).\boldsymbol{Pb}_{Sr} \\ \boldsymbol{Pb}_{Sr} &\stackrel{def}{=} \text{begin}.\boldsymbol{Pb}_F \\ \boldsymbol{Pb}_F &\stackrel{def}{=} (fetch, \top).\boldsymbol{Pb}_{Ff} + (fail, \top).\boldsymbol{Pb}_F + (recover, \top).\boldsymbol{Pb}_F \\ \boldsymbol{Pb}_{Ff} &\stackrel{def}{=} (fetch, \top).\boldsymbol{Pb}_{Fff} + (fail, \top).\boldsymbol{Pb}_F + (recover, \top).\boldsymbol{Pb}_{Ff} \\ \boldsymbol{Pb}_{Fff} &\stackrel{def}{=} \text{end}.\boldsymbol{Pb}_S.\end{aligned}$$

In order to perform the fluid approximation, we follow Section 4.2.1 and construct the probed iGPEPA models $\widetilde{\boldsymbol{SC}}(n,m)$ and $\widetilde{\boldsymbol{SC}'}(n,m)$ as follows:

$$\begin{aligned}\widetilde{\boldsymbol{SC}}(n,m) \stackrel{def}{=} & \mathbf{Clients}\{\boldsymbol{Client}_0[n]\} \\ & \underset{\{fetch\}}{\bowtie} \mathbf{Servers}\{(\boldsymbol{Serv}_0 \underset{K}{\bowtie} \boldsymbol{Pb}_S) \between \boldsymbol{Serv}_0[m-1]\},\end{aligned} \tag{4.13}$$

and

$$\begin{aligned}\widetilde{\boldsymbol{SC}'}(n,m) \stackrel{def}{=} & \mathbf{Clients}\{\boldsymbol{Client}_0[n]\} \\ & \underset{\{fetch\}}{\bowtie} \mathbf{Servers}\{(\boldsymbol{Serv}_0 \underset{K}{\bowtie} \boldsymbol{Pb}'_S) \between \boldsymbol{Serv}_0[m-1]\},\end{aligned} \tag{4.14}$$

where $K = \{fetch, fail, recover\}$ and $\boldsymbol{Pb}'_S$ is the translation of the absorbing version of $\boldsymbol{Probe}_l$. We then proceed by performing vanishing state removal as in Section 2.1.1 to obtain the derived transition systems of the probed f-components $\boldsymbol{Serv}_0 \underset{K}{\bowtie} \boldsymbol{Pb}_S$ and $\boldsymbol{Serv}_0 \underset{K}{\bowtie} \boldsymbol{Pb}'_S$. We note that the probes identify the beginning of the passage by the derived transition:

$$\boldsymbol{Serv}_2 \underset{K}{\bowtie} \boldsymbol{Pb}_S \xrightarrow{(recover,\cdot),(\text{begin})} \boldsymbol{Serv}_0 \underset{K}{\bowtie} \boldsymbol{Pb}_F,$$

and the passage is complete when the observed component is in any of the states $\boldsymbol{Serv}_0 \underset{K}{\bowtie} \boldsymbol{Pb}_{\text{Acc}}$, $\boldsymbol{Serv}_1 \underset{K}{\bowtie} \boldsymbol{Pb}_{\text{Acc}}$ or $\boldsymbol{Serv}_2 \underset{K}{\bowtie} \boldsymbol{Pb}_{\text{Acc}}$, that is, where the absorbing probe $\boldsymbol{Pb}'_S$ is in its accepting state $\boldsymbol{Pb}_{\text{Acc}}$.

As described in Section 4.2.1, fluid analysis of the passage time requires us to derive the system of 14 coupled ODEs corresponding to the probed iGPEPA model of (4.13). These are then solved to obtain an approximation to the stationary component count expectations, say $\tilde{v}_{Y,Q}$ for $(Y,Q) \in \mathcal{B}(\widetilde{\boldsymbol{SC}}(n,m))$. This can be performed either by numerical integration for a sufficiently long period of time, or, often much more efficiently, by algebraic means to find their meaningful fixed point.

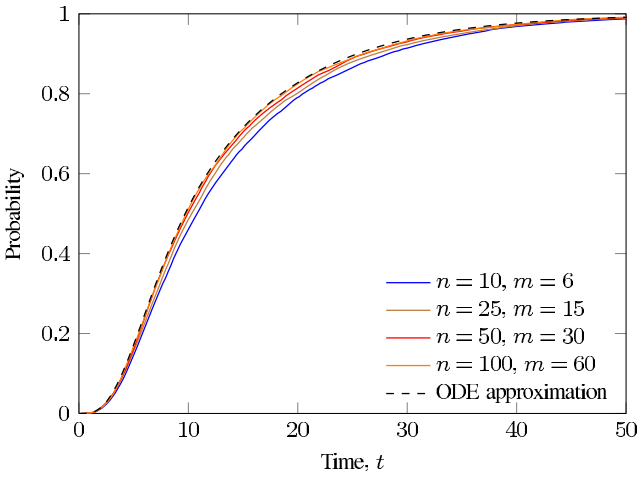


Fig. 1. Passage-time CDFs computed by simulation for the steady-state individual passage time specified by (4.12) compared with the ODE-derived approximation. Rates: $r_t = 0.4$, $r_s = 0.15$, $r_i = 0.6$, $r_r = 0.35$, and $r_f = 0.1$.

The final step, the computation of the CDF itself, requires us to derive the system of 17 coupled ODEs corresponding to the iGPEPA model of (4.14). The ODE approximation to the CDF is obtained in terms of the solutions to these ODEs, say $\tilde{v}'_{Y,Q}(t)$, by evaluating (4.3). For this measure, (4.3) is

$$\tilde{v}'_{\boldsymbol{Serv}_0 \underset{K}{\bowtie} \boldsymbol{Pb}_{\text{Acc}}}(t) + \tilde{v}'_{\boldsymbol{Serv}_1 \underset{K}{\bowtie} \boldsymbol{Pb}_{\text{Acc}}(t)} + \tilde{v}'_{\boldsymbol{Serv}_2 \underset{K}{\bowtie} \boldsymbol{Pb}_{\text{Acc}}}(t),$$

where the ODEs are initialized suitably with the quantities $\tilde{v}_{Y,Q}$ computed above as specified by the formula of (4.4).[10] For this measure, the initialization is fairly straightforward:

$$\begin{aligned} \tilde{v}'_{\boldsymbol{Client}_0}(0) &:= \tilde{v}_{\boldsymbol{Client}_0} \quad & \tilde{v}'_{\boldsymbol{Client}_1}(0) &:= \tilde{v}_{\boldsymbol{Client}_1} \\ \tilde{v}'_{\boldsymbol{Serv}_0}(0) &:= \tilde{v}_{\boldsymbol{Serv}_0} & \tilde{v}'_{\boldsymbol{Serv}_1}(0) &:= \tilde{v}_{\boldsymbol{Serv}_1} \\ \tilde{v}'_{\boldsymbol{Serv}_2}(0) &:= \tilde{v}_{\boldsymbol{Serv}_2} & \tilde{v}'_{\boldsymbol{Serv}_0 \underset{K}{\bowtie} \boldsymbol{Pb}_F}(0) &:= 1, \end{aligned}$$

with all other initial values set to zero.

Integrating these ODEs for the approximate CDF gives the results depicted in Fig. 1. The actual CDFs are also given for increasing component population size (n and m), and, as expected, we see that in this limit the accuracy of the approximation increases.[11] The simulated CDFs are computed to a maximum error of 0.01 with at least 95 percent confidence (as explained in Section 5.1).

5 Worked Example: Distributed Wireless Network

A distributed wireless network is composed of clients operating over a fixed bandwidth network. The clients are autonomous in so far as they have their own battery which can be recharged from their surroundings once the battery has discharged. The client is out of action until the battery is recharged. The wireless clients send data and control

10. For brevity, we drop the group names in the ODE quantities in this section.

11. It is a straightforward property of the approximating ODE systems that their solution is sensitive only to the initial relative component proportions, and not to their actual magnitude. This is why there is only one ODE-derived CDF depicted in Fig. 1 and in later figures.

messages to each other over the network, drawing on their batteries each time they do so.

The aim of this model is to investigate the high-level dynamics of intermittently available wireless clients as driven by a power-supply model. We show that despite the mass parallelism of the model and the many component states in the client state space, we can model and analyze this system to observe interesting features, as picked out by stochastic probes.

A wireless client has three states: hibernate, standby, and radio-in-use. Each state has a power usage profile. To send a data transfer, a client initializes the wireless radio with $radio\_init$ and then issues a $data\_tfr$ message. To send a control message, the client issues a $cont\_tfr$ message. The following f-component is used to model the wireless client:

$$\begin{aligned}
\boldsymbol{ClientHibernate} &\stackrel{def}{=} (clt_on, r_{on}).\boldsymbol{ClientStandby} \\
&\quad + (clt_shutdown, \top).\boldsymbol{ClientHibernate} \\
\boldsymbol{ClientStandby} &\stackrel{def}{=} (clt_off, r_{off}).\boldsymbol{ClientHibernate} \\
&\quad + (radio_init, r_{init}).\boldsymbol{ClientRadioUse} \\
&\quad + (cont_tfr, r_{cont}).\boldsymbol{ClientStandby} \\
&\quad + (clt_shutdown, \top).\boldsymbol{ClientHibernate} \\
\boldsymbol{ClientRadioUse} &\stackrel{def}{=} (data_tfr, r_{radio}).\boldsymbol{ClientStandby} \\
&\quad + (clt_shutdown, \top).\boldsymbol{ClientHibernate}.
\end{aligned}$$

The battery has N_b charge levels and discharges probabilistically from level i to $i-1$. With probability $\frac{\omega_\alpha}{1+\omega_\alpha}$, the battery drops a charge level with the ω_α determined by the exact action that is causing it to discharge. On reaching the zero level, the battery forces the wireless client into a hibernation mode by sending a $clt\_shutdown$ signal. Alternatively, the battery may be recharged a level at any point. The following f-component is used to model the battery:

$$\begin{aligned}
\boldsymbol{BatteryEmpty} &\stackrel{def}{=} (clt_charge, r_{charge}).\boldsymbol{Battery}_1 \\
\boldsymbol{Battery}_0 &\stackrel{def}{=} (clt_shutdown, r_{shutdown}).\boldsymbol{BatteryEmpty} \\
\boldsymbol{Battery}_i &\stackrel{def}{=} (data_tfr, \omega_{tfr} \times \top).\boldsymbol{Battery}_{i-1} \\
&\quad + (data_tfr, \top).\boldsymbol{Battery}_i \\
&\quad + (cont_tfr, \omega_{tfr} \times \top).\boldsymbol{Battery}_{i-1} \\
&\quad + (cont_tfr, \top).\boldsymbol{Battery}_i \\
&\quad + (radio_init, \omega_{init} \times \top).\boldsymbol{Battery}_{i-1} \\
&\quad + (radio_init, \top).\boldsymbol{Battery}_i \\
&\quad + (clt_off, \top).\boldsymbol{Battery}_i \\
&\quad + (clt_on, \top).\boldsymbol{Battery}_i \\
&\quad + (clt_charge, r_{charge}).\boldsymbol{Battery}_{\min(N_b, i+1)} \\
&\qquad : \text{for } 1 \leq i \leq N_b.
\end{aligned}$$

We then compose the $\boldsymbol{ClientHibernate}$ and $\boldsymbol{Battery}_{N_b}$ components together to form an f-component which models the combined client-battery interaction:

$$\boldsymbol{CB} \stackrel{def}{=} \boldsymbol{ClientHibernate} \underset{L_1 \cup L_2}{\bowtie} \boldsymbol{Battery}_{N_b}.$$

The client cannot switch on again until the battery is recharged via the clt_charge-action. Once this has happened the clt_on-action is available and the client can come into operation for another charge cycle. The action set $L_1 = \{clt\_on, clt\_off, clt\_shutdown\}$ represents the control actions between the battery and the wireless client and $L_2 = \{radio\_init, data\_tfr, cont\_tfr\}$ is the set of actions that discharge the battery.

The network consists of N_h channels which facilitate the transfer of data and control signals between wireless clients. Each channel is modeled using the following f-component:

$$\begin{aligned}
\boldsymbol{Chan} &\stackrel{def}{=} (data_tfr, r_{radio}).\boldsymbol{ChanBusy}_1 \\
&\quad + (cont_tfr, r_{cont}).\boldsymbol{ChanBusy}_2 \\
\boldsymbol{ChanBusy}_1 &\stackrel{def}{=} (data_tfr, r_{radio}).\boldsymbol{Chan} + (tmt, r_{tmt}).\boldsymbol{Chan} \\
\boldsymbol{ChanBusy}_2 &\stackrel{def}{=} (cont_tfr, r_{cont}).\boldsymbol{Chan} + (tmt, r_{tmt}).\boldsymbol{Chan}.
\end{aligned}$$

The final system is composed of the N_h channel network and N_c wireless clients which communicate using the actions in $M = \{data\_tfr, cont\_tfr\}$ to form the final iGPEPA model:

$$\boldsymbol{DWN}(N_c, N_h) \stackrel{def}{=} \mathbf{Clients}\{\boldsymbol{CB}[N_c]\} \underset{M}{\bowtie} \mathbf{Net}\{\boldsymbol{Chan}[N_h]\}.$$

In the next section, we present some example probe-specified passage-time measurements on this model alongside their approximate computations using the techniques detailed in this paper.

5.1 Example Measurements and Results

In this section, we provide example probe-specified passage-time measurements on the distributed wireless network model described above. For all of our examples, we let $N_b = 3$. We give examples for each of the types of passage time introduced above alongside comparisons of the fluid approximation and the actual corresponding cumulative distribution functions (CDFs) as computed by stochastic simulation. For all figures, the maximum error at each point of a simulated CDF is bounded above by 0.01 with at least 95 percent confidence. Confidence intervals were computed exactly using *binomial proportion confidence intervals* [26]. All numerical results were generated using the *Grouped PEPA Analyser (GPA)* tool [27], [28].

5.1.1 Steady-State Individual Passage Time

In order to provide an example of this kind of individual passage time, assume that we are interested in the time for a client's battery to discharge measured from immediately after it has performed at least one data transmission and one control interaction. A measure specification expressing this is

$$\begin{aligned}
\boldsymbol{PM} &\stackrel{def}{=} \text{begin} : \text{start}, \text{end} : \text{stop}^{\hookleftarrow} \\
&\quad \text{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \\
&\qquad (data_tfr; cont_tfr)/clt_shutdown : \text{begin}, \\
&\qquad clt_shutdown : \text{end}^{\hookleftarrow} \qquad (5.1) \\
&\quad \text{where} \quad \mathbf{Clients}\{\boldsymbol{CB}[?n]\} \Longrightarrow \\
&\qquad \mathbf{Clients}\{(\boldsymbol{CB} \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr\wr \boldsymbol{CB}[?n-1]\} \\
&\quad \text{in} \quad \boldsymbol{DWN}(N_c, N_h).
\end{aligned}$$

In order to perform the fluid approximation, we follow Section 4.2.1 and construct the probed iGPEPA models $\widetilde{\boldsymbol{DWN}}(N_c, N_h)$ and $\widetilde{\boldsymbol{DWN}}'(N_c, N_h)$ as follows:

$$\widetilde{\boldsymbol{DWN}}(N_c, N_h) \stackrel{def}{=} \mathbf{Clients}\{(\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}_S) \wr \boldsymbol{CB}[N_c - 1]\} \underset{M}{\bowtie} \mathbf{Net}\{\boldsymbol{Chan}[N_h]\}, \tag{5.2}$$

and

$$\widetilde{\boldsymbol{DWN}}'(N_c, N_h) \stackrel{def}{=} \mathbf{Clients}\{(\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}'_S) \wr \boldsymbol{CB}[N_c - 1]\} \underset{M}{\bowtie} \mathrm{Net}\{\boldsymbol{Chan}[N_h]\}, \tag{5.3}$$

where the local probe $\boldsymbol{Probe}_l$ has been translated to the iPEPA component $\boldsymbol{Pb}_S$ according to Section 3.2 as follows:

$$\begin{aligned}
\boldsymbol{Pb}_S &\stackrel{def}{=} (clt_shutdown, \top).\boldsymbol{Pb}_S + (data_tfr, \top).\boldsymbol{Pb}_{Sd} \\
&\quad + (cont_tfr, \top).\boldsymbol{Pb}_{Sc} \\
\boldsymbol{Pb}_{Sd} &\stackrel{def}{=} (clt_shutdown, \top).\boldsymbol{Pb}_S + (data_tfr, \top).\boldsymbol{Pb}_{Sd} \\
&\quad + (cont_tfr, \top).\boldsymbol{Pb}_{Scd} \\
\boldsymbol{Pb}_{Sc} &\stackrel{def}{=} (clt_shutdown, \top).\boldsymbol{Pb}_S + (data_tfr, \top).\boldsymbol{Pb}_{Scd} \\
&\quad + (cont_tfr, \top).\boldsymbol{Pb}_{Sc} \\
\boldsymbol{Pb}_{Scd} &\stackrel{def}{=} \mathrm{begin}.\boldsymbol{Pb}_F \\
\boldsymbol{Pb}_F &\stackrel{def}{=} (clt_shutdown, \top).\boldsymbol{Pb}_{Fs} + (data_tfr, \top).\boldsymbol{Pb}_F \\
&\quad + (cont_tfr, \top).\boldsymbol{Pb}_F \\
\boldsymbol{Pb}_{Fs} &\stackrel{def}{=} \mathrm{end}.\boldsymbol{Pb}_S,
\end{aligned}$$

and $K = \{clt\_shutdown, data\_tfr, cont\_tfr\}$. Note that $\boldsymbol{Pb}'_S$ is the translation of the absorbing version of $\boldsymbol{Probe}_l$. We then proceed by performing vanishing state removal as in Section 2.1.1 to obtain the derived transition systems of the probed f-components $\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}_S$ and $\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}'_S$.

The next stage of the analysis as described in Section 4.2.1 requires us to derive the system of 56 coupled ODEs corresponding to the probed iGPEPA model of (5.2). These are then solved to obtain an approximation to the stationary component count expectations, say $\tilde{v}_{Y,Q}$ for $(Y, Q) \in \mathcal{B}(\widetilde{\boldsymbol{DWN}}(N_c, N_h))$. As before, this can be performed either by numerical integration or algebraically to find their meaningful fixed point.

The final stage of the analysis requires us to derive the system of 68 coupled ODEs corresponding to the iGPEPA model of (5.3). The ODE approximation to the CDF is obtained in terms of the solutions to these ODEs, say $\tilde{v}'_{Y,Q}(t)$, as the quantity $\sum_{C \in \underset{*}{\bowtie} \boldsymbol{Pb}'_{\mathrm{Acc}}} \tilde{v}'_{H,C}(t)$, where $\boldsymbol{Pb}'_{\mathrm{Acc}}$ is the accepting state of the local probe $\boldsymbol{Pb}'_S$, and the ODEs are initialized suitably with the quantities $\tilde{v}_{Y,Q}$ computed above as specified by the formula of (4.4).

Fig. 2 compares the actual CDF for increasing numbers of f-components with the ODE approximation. As expected, we see that the accuracy of the approximation increases as the component populations (N_c and N_h) are scaled up.

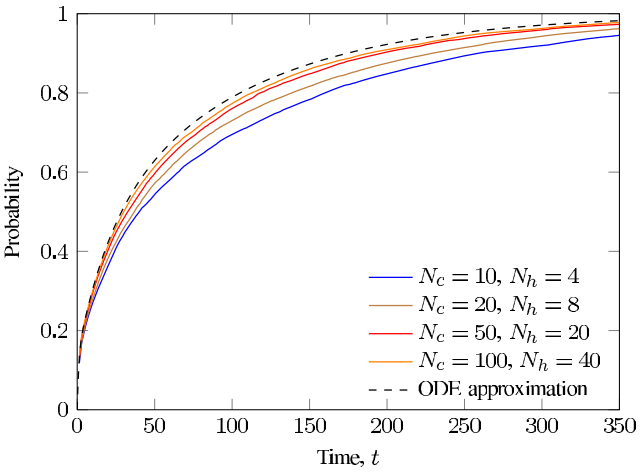


Fig. 2. Passage-time CDFs for the steady-state individual passage time specified by (5.1) compared with the ODE-derived approximation. Rates: $r_{on} = 0.3$, $r_{off} = 0.6$, $r_{init} = 0.5$, $r_{cont} = 0.85$, $r_{radio} = 0.16$, $r_{charge} = 0.02$, $r_{shutdown} = 1.5$, and $r_{tmt} = 0.2$; and weights: $\omega_{init} = 0.07$ and $\omega_{tfr} = 0.15$.

5.1.2 Transient Individual Passage Time

Consider a scenario where a data session begins after a $cont\_tfr$-action and ends after four $data\_tfr$-actions. We might be interested in the duration of such a data session. A measure specification expressing this is:

$$\begin{aligned}
\boldsymbol{PM} \stackrel{def}{=}\ & \mathrm{begin} : \mathrm{start}, \mathrm{end} : \mathrm{stop} \\
& \text{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} \\
& \qquad cont_tfr : \mathrm{begin}, data_tfr[4] : \mathrm{end} \\
& \text{where} \quad \mathbf{Clients}\{\boldsymbol{CB}[?n]\} \Longrightarrow \\
& \qquad \mathbf{Clients}\{(\boldsymbol{CB} \underset{*}{\bowtie} \boldsymbol{Probe}_l) \wr \boldsymbol{CB}[?n - 1]\} \\
& \text{in} \quad \boldsymbol{DWN}(N_c, N_h).
\end{aligned} \tag{5.4}$$

Note that due to the global probe being nonrepeating, this passage-time measurement is made from the model's deterministic initial state $\boldsymbol{DWN}(N_c, N_h)$, rather than in the steady-state regime. It thus measures the duration of the first data session.

In order to perform the ODE approximation, we follow Section 4.2.2 and construct the probed iGPEPA model $\widetilde{\boldsymbol{DWN}}(N_c, N_h)$ as follows:

$$\widetilde{\boldsymbol{DWN}}(N_c, N_h) \stackrel{def}{=} \mathbf{Clients}\{(\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}_S) \wr \boldsymbol{CB}[N_c - 1]\} \underset{M}{\bowtie} \mathbf{Net}\{\boldsymbol{Chan}[N_h]\}. \tag{5.5}$$

In this case, the local probe is translated to the iPEPA component $\boldsymbol{Pb}_S$:

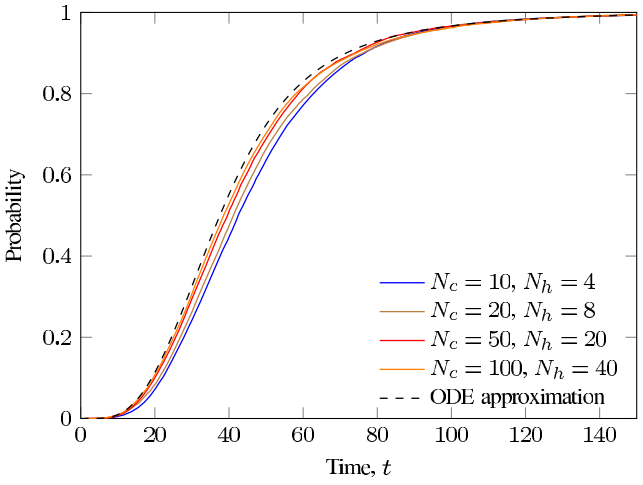


Fig. 3. Passage-time CDFs for the transient individual passage time specified by (5.4) compared with the ODE-derived approximation. Rates: $r_{on} = 0.3$, $r_{off} = 0.6$, $r_{init} = 0.5$, $r_{cont} = 0.85$, $r_{radio} = 0.35$, $r_{charge} = 0.05$, $r_{shutdown} = 1.5$, and $r_{tmt} = 0.2$; and weights: $\omega_{init} = 0.07$ and $\omega_{tfr} = 0.15$.

$$
\begin{aligned}
\boldsymbol{Pb}_S &\stackrel{def}{=} (cont\_tfr, \top).\boldsymbol{Pb}_{Sc} + (data\_tfr, \top).\boldsymbol{Pb}_S \\
\boldsymbol{Pb}_{Sc} &\stackrel{def}{=} \text{begin}.\boldsymbol{Pb}_F \\
\boldsymbol{Pb}_F &\stackrel{def}{=} (data\_tfr, \top).\boldsymbol{Pb}_{F1d} + (cont\_tfr, \top).\boldsymbol{Pb}_F \\
\boldsymbol{Pb}_{F1d} &\stackrel{def}{=} (data\_tfr, \top).\boldsymbol{Pb}_{F2d} + (cont\_tfr, \top).\boldsymbol{Pb}_{F1d} \\
\boldsymbol{Pb}_{F2d} &\stackrel{def}{=} (data\_tfr, \top).\boldsymbol{Pb}_{F3d} + (cont\_tfr, \top).\boldsymbol{Pb}_{F2d} \\
\boldsymbol{Pb}_{F3d} &\stackrel{def}{=} (data\_tfr, \top).\boldsymbol{Pb}_{F4d} + (cont\_tfr, \top).\boldsymbol{Pb}_{F3d} \\
\boldsymbol{Pb}_{F4d} &\stackrel{def}{=} \text{end}.\boldsymbol{Pb}_{\text{Acc}} \\
\boldsymbol{Pb}_{\text{Acc}} &\stackrel{def}{=} (data\_tfr, \top).\boldsymbol{Pb}_{\text{Acc}} + (cont\_tfr, \top).\boldsymbol{Pb}_{\text{Acc}},
\end{aligned}
$$

and $K = \{data\_tfr, cont\_tfr\}$. Like in the last section, we proceed by performing vanishing state removal to obtain the derived transition system of the probed f-component $\boldsymbol{CB} \underset{K}{\bowtie} \boldsymbol{Pb}_S$. Next, we derive the system of 83 coupled ODEs corresponding to the probed iGPEPA model of (5.5). The CDF fluid approximation is then given by (4.8) expressed in terms of the particular ODE solutions $\tilde{v}_{Y,Q}(t)$ and $\tilde{v}^s_{Y,Q}(t)$ for $(Y, Q) \in \mathcal{B}(\widetilde{\boldsymbol{DWN}}(N_c, N_h))$ computed as described in Section 4.2.2.

Fig. 3 compares the actual CDF for increasing numbers of f-components with the ODE approximation. As expected, we see that the accuracy of the approximation increases as the component populations (N_c and N_h) are scaled up.

5.1.3 Global Passage Time

In order to provide an example of a global passage-time measurement, we consider the time for half of the clients to have exhausted their battery at least once, starting from the model's initial state $\boldsymbol{DWN}(N_c, N_h)$. Initially, it might seem that no local probes are necessary and that an appropriate global probe specifying this is

$$\epsilon : \text{start}, clt\_shutdown[N_c/2] : \text{stop}. \tag{5.6}$$

However, this does not specify the correct quantity since if a single individual performs $clt\_shutdown$ more than once, the global probe should only count it the first time. The

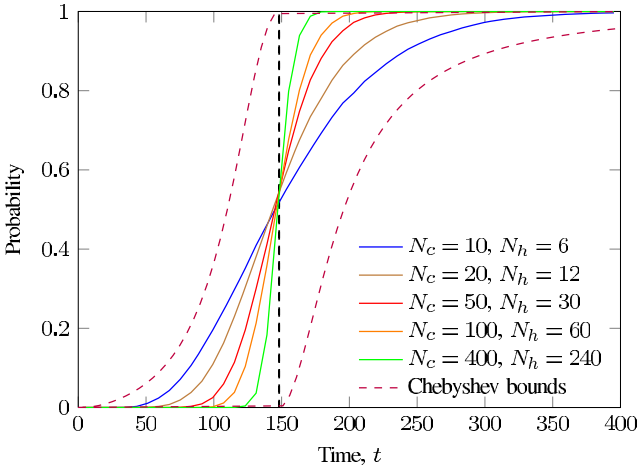


Fig. 4. Passage-time CDFs for the global passage time specified by (5.7) compared with the ODE-derived point-mass approximation (the dashed line). Also shown are ODE-derived approximate CDF bounds for the case $N_c = 20$, $N_h = 12$. Rates: $r_{on} = 0.3$, $r_{off} = 0.6$, $r_{init} = 0.5$, $r_{cont} = 0.85$, $r_{radio} = 0.4$, $r_{charge} = 0.1$, $r_{shutdown} = 1.5$, and $r_{tmt} = 0.2$; and weights: $\omega_{init} = 0.07$ and $\omega_{tfr} = 0.15$.

situation is easily resolved however by attaching an appropriate local probe. A complete measure specification for this passage-time quantity is

$$
\begin{aligned}
\boldsymbol{PM} \stackrel{def}{=}\ & \epsilon : \text{start}, \text{end}[N_c/2] : \text{stop} \\
& \text{observes} \quad \boldsymbol{Probe}_l \stackrel{def}{=} clt\_shutdown : \text{end} \\
& \text{where} \quad \mathbf{Clients}\{\boldsymbol{CB}[?n]\} \Longrightarrow \\
& \qquad \mathbf{Clients}\{(\boldsymbol{CB} \underset{*}{\bowtie} \boldsymbol{Probe}_l)[?n]\} \\
& \text{in} \quad \boldsymbol{DWN}(N_c, N_h).
\end{aligned} \tag{5.7}
$$

This is now correct because the local probe is nonrepeating and thus the end signal is only transmitted at most once for each $\boldsymbol{CB}$ component.

Like for previous examples, we proceed by constructing the probed iGPEPA model's derived transition system and the associated system of 26 ODEs—one each whose solution counts the number of f-components in each of their nonvanishing derivative states, and one whose solution, $v_{\text{end}}(t)$, corresponds to the end-counting process, $N_{\text{end}}(t)$. In line with Section 4.2.3, the deterministic approximation to the global passage time of interest is then given by

$$\inf\{t \geq 0 : v_{\text{end}}(t) \geq N_c/2\}.$$

Although outside the explicit scope of this paper, it is possible to derive similar systems of ODEs approximating higher order moments as detailed in [19] for GPEPA and [22] for iGPEPA. These can then be combined with well-known inequalities such as Chebyshev's inequality or other general techniques for bounding distributions based on moment information (e.g., [29]) to obtain accurate bounds on the global passage-time CDF. These are particularly useful for smaller populations where the deterministic approximation is far too coarse. See [22], [6] for more details and discussion on this approach.

Fig. 4 compares the actual CDF for increasing numbers of f-components with the deterministic ODE approximation and with an example of approximate CDF bounds derived

using the second-order Chebyshev's inequality. As expected, we see that the accuracy of the deterministic approximation increases as the component populations (N_c and N_h) are scaled up.

6 Conclusion

In this paper, we have introduced the unified stochastic probe mechanism for efficiently specifying complex passage-time performance measures. We have shown that, in its most general form, many stochastic probes can be placed on a process model in order to capture precise behavior from distinct components of the model and that these can in turn signal to a master measurement probe which can start and stop key passage-time measures. Other probe mechanisms are also supported, such as state-based activation to provide a flexible mechanism for the modeler and for easy comparison with other techniques and tools. In order to support the unified stochastic probe mechanism, we have formally extended the stochastic process algebra GPEPA with immediate transitions and presented a fluid translation to a set of differential equations for this formalism. Additionally, we have selected a subset of the unified stochastic probe formalism and shown how popular passage-time measures can be specified, so-called *global* and *individual* passage times, useful for SLAs. We have shown how these passage-time queries can be implemented using fluid ODE analysis, which permits very rapid analysis of massively parallel models. Where fluid techniques cannot be used to tackle a particular unified probed query, then traditional CTMC analysis techniques and tools can be deployed [5], [13], [14]. Finally, we have demonstrated the combined techniques on an $O(10^{129})$ state stochastic model of a distributed wireless network.

Appendix A
PEPA with Immediate Actions

A.1 Structured Operational Semantics

Before we present the operational semantics for iPEPA, we discuss in more detail the exact behavior of passive cooperation. Recall that the rate of a timed action can be any element of $\mathbb{R}_+ \cup \{n\top | n \in \mathbb{Q}, n > 0\}$, where $n\top$ is shorthand for $n \times \top$ and $\top$ represents the passive action rate that inherits the rate of the coaction from the cooperating component. $\top$ requires the following arithmetic rules:

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

Note that rates of the form $(r + n\top)$ are disallowed for all $r \in \mathbb{R}, r \neq 0, n \in \mathbb{Q}, n \neq 0$ in iPEPA. This is in line with the requirement that iPEPA components are not allowed to enable both active and passive actions in the same action type at the same time, e.g., $(\alpha, \lambda).P + (\alpha, \top).P'$ is not allowed.

Fig. 5 gives the operational semantics for iPEPA. It requires the definition of the *total weight* function $w(P)$, which computes the sum of the weights of concurrently enabled immediate activities for an iPEPA component P: $w((\beta, \lambda).P) := 0$, $w([\mathsf{a}, w].P) := w$, $w(P + Q) := w(P) + w(Q)$ and $w(P \underset{L}{\bowtie} Q) := w(P) + w(Q)$. We also require the *total probability function*, which computes the sum of the probabilities of concurrently-enabled immediate activities of a given immediate action type, for a given iPEPA component: $Z_{\mathsf{a}}(P) := \sum_{P \xrightarrow{(\mathsf{a},\cdot,z)}} z$.

We may now give a formal definition of the set of derivative states of an iPEPA component: $ds(P)$ is the smallest set of iPEPA components such that $P \in ds(P)$ and if, for any $P_1 \in ds(P)$, $P_1 \xrightarrow{(\cdot,\cdot)} P_2$ or $P_1 \xrightarrow{(\cdot,\cdot,\cdot)} P_2$, then $P_2 \in ds(P)$. Immediate transitions have three parameters.

Prefix

$$\frac{}{(\alpha,\lambda).P \xrightarrow{(\alpha,\lambda)} P}\ \alpha \in \mathcal{A}_t \qquad \frac{}{[\mathsf{a},w].P \xrightarrow{(\mathsf{a},w,1)} P}\ \mathsf{a} \in \mathcal{A}_i$$

Competitive Choice

$$\frac{P \xrightarrow{(\alpha,\lambda)} P'}{P + Q \xrightarrow{(\alpha,\lambda)} P'}\ \alpha \in \mathcal{A}_t \qquad \frac{Q \xrightarrow{(\beta,\mu)} Q'}{P + Q \xrightarrow{(\beta,\mu)} Q'}\ \beta \in \mathcal{A}_t$$

$$\frac{P \xrightarrow{(\mathsf{a},w,z)} P'}{P + Q \xrightarrow{(\mathsf{a},w,Z)} P'}\ \mathsf{a} \in \mathcal{A}_i \qquad \frac{Q \xrightarrow{(\mathsf{b},w,z)} Q'}{P + Q \xrightarrow{(\mathsf{b},w,Z)} Q'}\ \mathsf{b} \in \mathcal{A}_i$$

where $Z := \frac{w}{w(P+Q)}$

Cooperation

$$\frac{P \xrightarrow{(\alpha,\lambda)} P'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,\lambda)} P' \underset{L}{\bowtie} Q}\ \alpha \in \mathcal{A}_t, \alpha \notin L$$

$$\frac{Q \xrightarrow{(\beta,\mu)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\beta,\mu)} P \underset{L}{\bowtie} Q'}\ \beta \in \mathcal{A}_t, \beta \notin L$$

$$\frac{P \xrightarrow{(\mathsf{a},\cdot,z)} P'}{P \underset{L}{\bowtie} Q \xrightarrow{(\mathsf{a},\cdot,Z_1)} P' \underset{L}{\bowtie} Q}\ \mathsf{a} \in \mathcal{A}_i, \mathsf{a} \notin L$$

$$\frac{Q \xrightarrow{(\mathsf{b},\cdot,z)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\mathsf{b},\cdot,Z_2)} P \underset{L}{\bowtie} Q'}\ \mathsf{b} \in \mathcal{A}_i, \mathsf{b} \notin L$$

$$\frac{P \xrightarrow{(\alpha,\lambda)} P' \quad Q \xrightarrow{(\alpha,\mu)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,R)} P' \underset{L}{\bowtie} Q'}\ \alpha \in \mathcal{A}_t, \alpha \in L$$

$$\frac{P \xrightarrow{(\mathsf{a},\cdot,z_1)} P' \quad Q \xrightarrow{(\mathsf{a},\cdot,z_2)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\mathsf{a},\cdot,Z)} P' \underset{L}{\bowtie} Q'}\ \mathsf{a} \in \mathcal{A}_i, \mathsf{a} \in L$$

where $Z_1 := \frac{w(P)}{w(P \underset{L}{\bowtie} Q)} z$, $Z_2 := \frac{w(Q)}{w(P \underset{L}{\bowtie} Q)} z$,
$R := \frac{\lambda}{r_\alpha(P)} \frac{\mu}{r_\alpha(Q)} \min(r_\alpha(P), r_\alpha(Q))$ and
$Z := \frac{w(P)}{w(P \underset{L}{\bowtie} Q)} z_1 \frac{z_2}{z_{\mathsf{a}}(Q)} + \frac{w(Q)}{w(P \underset{L}{\bowtie} Q)} z_2 \frac{z_1}{z_{\mathsf{a}}(P)}$

Constant

$$\frac{P \xrightarrow{(\mathsf{a},w,z)} P'}{\boldsymbol{C} \xrightarrow{(\mathsf{a},w,z)} P'}\ \mathsf{a} \in \mathcal{A}_i, \boldsymbol{C} \stackrel{def}{=} P \qquad \frac{P \xrightarrow{(\alpha,\lambda)} P'}{\boldsymbol{C} \xrightarrow{(\alpha,\lambda)} P'}\ \alpha \in \mathcal{A}_t, \boldsymbol{C} \stackrel{def}{=} P$$

Fig. 5. Operational semantics for iPEPA.

The first is the corresponding immediate action type. The second is the weight of the transition, which is only used for sequential components and is thus suppressed in Fig. 5 for model components. Finally, the third parameter is a probability value determining a probability distribution over all concurrently enabled immediate transitions. This distribution is used to decide which of these immediate transitions actually completes. This probability is recomputed at each structural level by the operational semantics in a similar fashion to the computation of the rate of a timed action under cooperation.

The evolution of a composed model enabling immediate transitions depends on the ratio of the sums of enabled weights between the constituent sequential components. This affords the modeler more flexibility. Specifically, it allows the modeler to have control both over the local probabilistic choice arising due to multiply enabled immediate actions in sequential components, and, if desired, over the choice arising due to multiply enabled immediate actions in model components.

A.2 Vanishing State Removal

A.2.1 The Derived Transition System of an iPEPA Component

When viewed as a CTMC, the state space of a well-behaved iPEPA component P is the set of *nonvanishing derivative states*, $ds^*(P) \subseteq ds(P)$. That is, the set of derivative states of P which do not enable any immediate actions:

$$ds^*(P) := \{P' \in ds(P) : P' \not\xrightarrow{} \}.$$

The immediate transitions emanating from vanishing derivative states can be removed if we replace paths of immediate transitions with the timed transition they determine between elements of $ds^*(P)$. Specifically, for each $P' \in ds^*(P)$, we consider every distinct path from P' to other nonvanishing derivative states $P'' \in ds^*(P)$ where the first transition is timed and the remaining transitions are immediate:

$$P' \xrightarrow{(\alpha,r)} P_{(1)} \xrightarrow{(\mathrm{a}_1,\cdot,z_1)} \cdots \xrightarrow{(\mathrm{a}_{K-1},\cdot,z_{K-1})} P_{(K)} \xrightarrow{(\mathrm{a}_K,\cdot,z_K)} P'',$$

where $K \geq 1$. This path can be replaced by a single new timed α-transition of rate R, $P' \xrightarrow{(\alpha,R),\mathcal{I}} P''$, where $R := r \times \prod_{i=1}^{K} z_i$ and $\mathcal{I} := (\mathrm{a}_1, \ldots, \mathrm{a}_K)$. The extra sequence parameter $\mathcal{I}$ extends the transition system so that it is still possible to determine when immediate actions are performed and in which order.[12]

In the case that the initial derivative state P of a well-behaved iPEPA component is itself vanishing, the well-behaved condition guarantees that there is a unique nonvanishing derivative state reachable by following immediate transitions from P. This nonvanishing derivative state is taken to be the initial state of the iPEPA component in the derived transition system.

12. In the derived transition system, where a timed transition does not originate from the removal of immediate transitions we set $\mathcal{I} := ()$, the empty sequence.

Appendix B

Immediate Grouped PEPA Operational Semantics

We define the operational semantics of an iGPEPA model to be identical to that of the equivalent iPEPA model which is obtained syntactically by removing the group labels and replacing the $\wr\wr$ combinators with $\|$. We call this operation *flattening* and a formal *flattening function* is given in Definition B.1. We may thus define the equivalent operational semantics on iGPEPA models by composing the flattening function with the operational semantics of iPEPA, given in Fig. 5 of Appendix A.1.

Formally, for any two iGPEPA models, G_1 and G_2, we say $G_1 \xrightarrow{(\alpha,r)} G_2$ or $G_1 \xrightarrow{(\mathrm{a},w,z)} G_2$ if and only if $\mathcal{F}(G_1) \xrightarrow{(\alpha,r)} \mathcal{F}(G_2)$ or $\mathcal{F}(G_1) \xrightarrow{(\mathrm{a},w,z)} \mathcal{F}(G_2)$, respectively, also counting the multiplicity of such transitions. We may thus extend the notion of (nonvanishing) derivative states to iGPEPA models in the analogous way. Removing the vanishing states of an iGPEPA model exactly as in the case of iPEPA components gives rise to the *underlying CTMC* of an iGPEPA model, which is trivially isomorphic to that of the corresponding equivalent iPEPA model obtained through flattening.

It is important to note at this stage that we have explicitly disallowed cooperation on immediate actions over component groups by restricting the cooperation set in (2.4) to elements of $\mathcal{A}_t$. Moreover, cooperation on timed activities enabled passively is also not allowed to occur across component groups. This requirement has already been stated implicitly since we required that all passive actions enabled in f-components were synchronized. Both of these requirements are necessary to ensure a class of model which is amenable to fluid analysis.

Definition B.1 (Flattening function). *For any iGPEPA model G, the corresponding iPEPA model, $\mathcal{F}(G)$, can be recovered from the grouped model. $\mathcal{F}(\cdot)$ is defined by $\mathcal{F}(M_1 \underset{L}{\bowtie} M_2) := \mathcal{F}(M_1) \underset{L}{\bowtie} \mathcal{F}(M_2)$ and $\mathcal{F}(Y\{D\}) := \mathcal{F}'(D)$, where for component groups $\mathcal{F}'(D_1 \wr\wr D_2) := \mathcal{F}'(D_1) \| \mathcal{F}'(D_2)$ and $\mathcal{F}'(P) := P$.*

Appendix C

Translation of Nonstandard Regular Expressions to DFAs

Let R_l^1 and R_l^2 be two regular expressions according to the grammar of (3.3). Let $M^1 = (Q^1, \Sigma, \delta^1, q_0^1, F^1)$ and $M^2 = (Q^2, \Sigma, \delta^2, q_0^2, F^2)$ be their respective DFAs, where:

- Q^1 and Q^2 are the *finite sets of states*;
- Σ is the *input alphabet* (for our purposes, this is, $\mathcal{A}(P)$ where P is the f-component to which the local probe is to be applied);
- $\delta^1 : Q^1 \times \Sigma \to Q^1$ and $\delta^2 : Q^2 \times \Sigma \to Q^2$ are the *transition functions*;
- $q^1 \in Q^1$ and $q^2 \in Q^2$ are the *start states*;
- $F^1 \subseteq Q^1$ and $F^2 \subseteq Q^2$ are the *sets of accepting states*.

We now show how to construct DFAs for the regular expressions $R_l^1;\ R_l^2$, R_l^1/R_l^2, and $R_l^1 \emptyset R_l^2$.

C.1 Both ($R_l^1;\ R_l^2$)

The DFA corresponding to $R_l^1;\ R_l^2$ is $M = (Q, \Sigma, \delta, q_0, F)$, where:

- $Q := Q^1 \times Q^2$;
- For all $x \in Q^1$, $y \in Q^2$, and $a \in \Sigma$, $\delta((x,y),a) := (\delta^1(x,a), \delta^2(y,a))$;
- $q_0 := (q_0^1, q_0^2)$;
- $F := F^1 \times F^2$.

C.2 Reset (R_l^1/R_l^2)

The DFA corresponding to R_l^1/R_l^2 is $M = (Q, \Sigma, \delta, q_0, F)$, where:

- $Q := Q^1 \times Q^2$;
- For all $x \in Q^1$, $y \in Q^2$, and $a \in \Sigma$, $\delta((x,y),a) := (\delta^1(x,a), \delta^2(y,a))$ if $\delta^2(y,a) \notin F^2$, or $\delta((x,y),a) := q_0$ otherwise.
- $q_0 := (q_0^1, q_0^2)$;
- $F := F^1 \times (Q^2 \setminus F^2)$.

C.3 Fail ($R_l^1 \emptyset R_l^2$)

The DFA corresponding to $R_l^1 \emptyset R_l^2$ is $M = (Q, \Sigma, \delta, q_0, F)$, where:

- $Q := Q^1 \times Q^2$;
- For all $x \in Q^1$, $y \in Q^2 \setminus F^2$, and $a \in \Sigma$, $\delta((x,y),a) := (\delta^1(x,a), \delta^2(y,a))$, and for all $x \in Q^1$, $y \in F^2$, and $a \in \Sigma$, $\delta((x,y),a) := (x,y)$;
- $q_0 := (q_0^1, q_0^2)$;
- $F := F^1 \times (Q^2 \setminus F^2)$.

REFERENCES

[1] A.L. Wolf and D.S. Rosenblum, "A Study in Software Process Data Capture and Analysis," *Proc. Second Int'l Conf. Software Process,* pp. 115-124, Feb./Mar. 1993.

[2] A. Argent-Katwala, J.T. Bradley, and N.J. Dingle, "Expressing Performance Requirements Using Regular Expressions to Specify Stochastic Probes over Process Algebra Models," *Proc. Fourth Int'l Workshop Software and Performance,* pp. 49-58, 2004.

[3] A. Argent-Katwala, J. Bradley, A. Clark, and S. Gilmore, "Location-Aware Quality of Service Measurements for Service-Level Agreements," *Proc. Third Int'l Conf. Trustworthy Global Computing,* pp. 222-239, 2008.

[4] A. Clark and S.T. Gilmore, "Transformations in PEPA Models and Stochastic Probe Placement," *Proc. 25th UK Performance Eng. Workshop,* K. Djemame, ed., pp. 1-16, 2009.

[5] A. Clark and S.T. Gilmore, "State-Aware Performance Analysis with eXtended Stochastic Probes," *Proc. Fifth European Performance Eng. Workshop,* N. Thomas and C. Juiz, eds., pp. 125-140, 2008.

[6] R.A. Hayden, A. Stefanek, and J.T. Bradley, "Fluid Computation of Passage Time Distributions in Large Markov Models," *Theoretical Computer Science,* vol. 413, pp. 106-141, Jan. 2012.

[7] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton, "Verifying Continuous-Time Markov Chains," *Proc. Eighth Int'l Conf. Computer-Aided Verification,* pp. 269-276, 1996.

[8] H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, and M. Siegle, "Towards Model Checking Stochastic Process Algebra," *Proc. Second Int'l Conf. Integrated Formal Methods,* pp. 420-439, Nov. 2000.

[9] C. Baier, L. Cloth, B.R. Haverkort, M. Kuntz, and M. Siegle, "Model Checking Markov Chains with Actions and State Labels," *IEEE Trans. Software Eng.,* vol. 33, no. 4, pp. 209-224, Apr. 2007.

[10] S. Donatelli, S. Haddad, and J. Sproston, "Model Checking Timed and Stochastic Properties with $\mathrm{CSL}^{\mathrm{TA}}$," *IEEE Trans. Software Eng.,* vol. 35, no. 2, pp. 224-240, Mar. 2009.

[11] C.M. Woodside and C. Shramm, "Complex Performance Measurements with NICE (Notation for Interval Combinations and Events)," *Software—Practice and Experience,* vol. 24, pp. 1121-1144, Dec. 1994.

[12] W.J. Knottenbelt and P.G. Harrison, "Distributed Disk-Based Solution Techniques for Large Markov Models," *Proc. Third Int'l Conf. the Numerical Solution of Markov Chains,* pp. 58-75, Sept. 1999.

[13] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: Probabilistic Symbolic Model Checker," *Proc. 12th Int'l Conf. Modelling Techniques and Tools for Computer Performance Evaluation,* pp. 200-204, 2002.

[14] D.D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J.M. Doyle, W.H. Sanders, and P.G. Webster, "The Möbius Framework and Its Implementation," *IEEE Trans. Software Eng.,* vol. 28, no. 10, pp. 956-969, Oct. 2002.

[15] M. Kuntz and M. Siegle, "Symbolic Model Checking of Stochastic Systems: Theory and Implementation," *Proc. 13th Int'l SPIN Workshop Model Checking Software,* pp. 89-107, 2006.

[16] S. Gilmore, J. Hillston, and M. Ribaudo, "An Efficient Algorithm for Aggregating PEPA Models," *IEEE Trans. Software Eng.,* vol. 27, no. 5, pp. 449-464, May 2001.

[17] M.A. Marsan, G. Conte, and G. Balbo, "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems," *ACM Trans. Computer Systems,* vol. 2, pp. 93-122, May 1984.

[18] H. Hermanns, *Interacting Markov Chains and the Quest for Quantified Quality.* Springer, 2002.

[19] R.A. Hayden and J.T. Bradley, "A Fluid Analysis Framework for a Markovian Process Algebra," *Theoretical Computer Science,* vol. 411, pp. 2260-2297, May 2010.

[20] J. Hillston, *A Compositional Approach to Performance Modelling.* Cambridge Univ. Press, 1996.

[21] R.A. Hayden and J.T. Bradley, "A Fluid Analysis Framework for a Markovian Process Algebra," *Theoretical Computer Science,* vol. 411, pp. 2260-2297, May 2010, doi: //10.1016/j.tcs.2010.02.001.

[22] R.A. Hayden, "Scalable Performance Analysis of Massively Parallel Stochastic Systems," PhD thesis, Imperial College London, http://pubs.doc.ic.ac.uk/hayden-thesis/, 2011.

[23] M. Sipser, *Introduction to the Theory of Computation.* PWS, 1997.

[24] R. Serfozo, *Basics of Applied Stochastic Processes.* Springer, 2009.

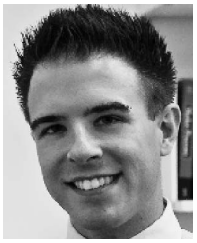
[25] E.G. Amparore, M. Beccuti, S. Donatelli, and G. Franceschinis, "Probe Automata for Passage Time Specification," *Proc. Eighth Int'l Conf. Quantitative Evaluation of Systems,* pp. 101-110, 2011.

[26] H.R. Neave, *Statistics Tables for Mathematicians, Engineers, Economists and the Behavioural and Management Sciences.* George Allen & Unwin, 1978.

[27] A. Stefanek, R.A. Hayden, and J.T. Bradley, "A New Tool for the Performance Analysis of Massively Parallel Computer Systems," *Proc. Eighth Workshop Quantitative Aspects of Programming Languages,* pp. 159-181, 2010.

[28] A. Stefanek, R.A. Hayden, and J.T. Bradley, "GPA—A Tool for Fluid Scalability Analysis of Massively Parallel Systems," *Proc. Eighth Int'l Conf. Quantitative Evaluation of Systems,* pp. 147-148, http://pubs.doc.ic.ac.uk/gpanalyser/, 2011.

[29] A. Tari, M. Telek, and P. Buchholz, "A Unified Approach to the Moments Based Distribution Estimation—Unbounded Support," *Proc. Int'l Conf. European Performance Eng., and Web Services and Formal Methods, Int'l Conf. Formal Techniques for Computer Systems and Business Processes,* pp. 79-93, 2005.



Richard A. Hayden received the PhD degree in performance modeling from the Department of Computing at Imperial College London in 2011. He was awarded the Best Computational Science Student of the Year at the 2007 Science, Engineering & Technology Student of the Year Awards, a national award sponsored by Microsoft Research. He is now a research associate at Imperial College and his research interests lie in tackling the state-space explosion problem for massive performance models using fluid analysis and related techniques.

Jeremy T. Bradley received the PhD degree in computer science from the University of Bristol in 2000. He is a reader in scalable performance analysis in the Department of Computing at Imperial College London. After research posts at Bristol and Durham, he was appointed a lecturer at Imperial College London in 2003. His research has produced parallel algorithms for transient and response-time analysis of large semi-Markov processes. Lately, he has focused on fluid analysis of performance and energy models, as derived from stochastic Petri nets and stochastic process algebras augmented with rewards.

Allan Clark received both the undergraduate and doctorate degrees from the University of Edinburgh School of Informatics. His PhD thesis concerned functional programming but in his postdoctorate research he has worked on performance modeling and, in particular, on the related process algebras PEPA and Bio-PEPA. He is now a researcher at the University of Edinburgh splitting his time between the School of Informatics and the Centre for Systems Biology at Edinburgh, where he combines performance modeling and formal methods to enable reliable biological modeling of models containing fewer human errors.